PROD. TEC. ITV DS - N025/2021
DOI 10.29223/PROD.TEC.ITV.DS.2021.25.GirolamoNeto

RELATÓRIO TÉCNICO ITV DS

COMPARAÇÃO DE MODELOS PARA PREDIÇÃO DO DESMATAMENTO NA AMAZÔNIA BRASILEIRA

PROJETO DEFINIÇÃO DE ÁREAS PRIORITÁRIAS PARA RECUPERAÇÃO FLORESTAL

Cesare di Girolamo Neto
Rodolfo Jaffé
Rosane Barbosa Lopes Cavalcante
Sâmia Nunes

Belém / PA
Agosto / 2021

INSTITUTO TECNOLÓGICO VALE

Título: Modelagem do desmatamento na Amazônia Brasileira	
PROD. TEC. ITV DS N025/2021	**Revisão**
Classificação: ( ) Confidencial ( ) Restrita ( ) Uso Interno (x) Pública	**00**

.

Informações Confidenciais - Informações estratégicas para o Instituto e sua Mantenedora. Seu manuseio é restrito a usuários previamente autorizados pelo Gestor da Informação.

Informações Restritas - Informação cujo conhecimento, manuseio e controle de acesso devem estar limitados a um grupo restrito de empregados que necessitam utilizá-la para exercer suas atividades profissionais.

Informações de Uso Interno - São informações destinadas à utilização interna por empregados e prestadores de serviço

Informações Públicas - Informações que podem ser distribuídas ao público externo, o que, usualmente, é feito através dos canais corporativos apropriados

Citar como
GIROLAMO NETO, Cesare di; *et al.* **Comparação de modelos para predição do desmatamento na Amazônia brasileira.** Belém: ITV, 2021. (Relatório Técnico N025/2021) DOI 10.29223/PROD.TEC.ITV.DS.2021.25.GirolamoNeto

Dados Internacionais de Catalogação na Publicação (CIP)

G526 Girolamo Neto, Cesare di.
Comparação de modelos para predição do desmatamento na Amazônia brasileira. / Cesare di Girolamo Neto, Rodolfo Jaffé, Rosane Barbosa Lopes Cavalcante, Sâmia Nunes - Belém: ITV, 2021.

76 p. : il.

Relatório Técnico (Instituto Tecnológico Vale) – 2021
PROD.TEC.ITV.DS – N025/2021
DOI 10.29223/PROD.TEC.ITV.DS.2021.25.GirolamoNeto

1. Amazônia Brasileira – desmatamento. 2. Desmatamento – Modelagem – Amazônia Brasileira. I. Jaffé, Rodolfo. II. Cavalcante, Rosane Barbosa Lopes. III. Nunes, Sâmia. IV. Título

CDD 23. ed.363.70098115

Bibliotecária responsável: Nisa Gonçalves / CRB 2 – 525

RESUMO EXECUTIVO

O presente relatório contém resultados parciais do projeto “Definição de áreas prioritárias para recuperação florestal”, referentes a atividade “Uso e comparação da acurácia de diferentes modelos preditivos de desmatamento na Amazônia”. Estudos recentes do ITV indicaram uma área total de 327.500 ha que precisa ser recuperada apenas na bacia hidrográfica do rio Itacaiúnas, sudeste paraense, sendo 59% em RL e 41% em APP, além de identificar áreas prioritárias para recuperação florestal na bacia, com uma abordagem multicritério. A recuperação e conservação florestal estão alinhadas com políticas e iniciativas nacionais e internacionais de limitar o aquecimento global e conservar a biodiversidade. Nesta etapa do projeto, foarm analisados o desempenho de diferentes modelos para a predição do desmatamento na Amazônia brasileira, visando a identificação de áreas prioritárias para ações de combate ao desmatamento. Uma ampla base de dados geográficos foi gerada por meio da integração de dados de diversas instituições brasileiras e está apresentada no Apêndice, assim como os scripts desenvolvidos. Os principais drivers de desmatamento identificados estão relacionados à fragmentação florestal e à expansão de áreas de pastagem na Amazônia, corroborando com outros trabalhos encontrados em literatura. A modelagem obteve melhores resultados com o uso dos modelos Random Forest e Spatial Random Forest, prevendo um desmatamento de 31 mil km^2 para 2020, concentrados no nordeste e extremo sul do bioma.

RESUMO

O presente relatório contém resultados parciais do projeto “Definição de áreas prioritárias para recuperação florestal”, referentes a atividade “Uso e comparação da acurácia de diferentes modelos preditivos de desmatamento na Amazônia”. O objetivo deste estudo foi a implementação de modelos preditivos de desmatamento na Amazônia brasileira com base nas técnicas de Random Forest (RF), Spatial Random Forest (SpRF) e Integrated Nested Laplace Approximations (INLA) e comparação dos erros obtidos com cada modelo. Uma base de dados geográficos foi gerada por meio da integração de dados de diversas instituições brasileiras, como IBGE, MMA e INPE, utilizando células de 25 x 25 km e uma janela temporal de um ano. Os principais drivers de desmatamento identificados estão relacionados à fragmentação florestal e à expansão de áreas de pastagem na Amazônia, corroborando com outros trabalhos encontrados em literatura. A modelagem obteve melhores resultados com o uso dos modelos RF e SpRF em relação aos modelos do tipo INLA, com menores valores de erro médio quadrático obtido em conjuntos de dados de treinamento e validação dos algoritmos. A previsão de desmatamento para o ano de 2020 foi de 31 mil km^2, dados que apresentam uma superestimava devido ao método utilizado para o cálculo do desmatamento. Entre as ações identificadas que podem ser adotadas em trabalhos futuros para melhorar a previsão do desmatamento, cita-se o uso da abordagem CLUE e a melhoria de algumas bases de dados utilizada, a exemplo da malha viária.

Palavras-chave: modelagem de desmatamento; Random Forest; INLA.

ABSTRACT

This report contains partial results of the project Definition of priority areas for forest recovery. This study aimed to implement predictive models of deforestation for the Brazilian Amazon based on the algorithms Random Forest (RF), Spatial Random Forest (SpRF), and Integrated Nested Laplace Approximations (INLA) and to compare the errors obtained with each technique. We generated a geodatabase integrating data from several Brazilian institutions, using 25 x 25 km cells and a one-year time window. The most important deforestation drivers found in this project are forest fragmentation and the expansion of pasture areas in the Amazon, corroborating other works found in the literature. Better results were obtained using the RF and the SpRF models than the INLA-type models, with lower mean square error values in the training and validation datasets. The deforestation forecast for the year 2020 was about 31,000 km. This value is overestimated due to the methods used to calculate deforestation. Among the identified actions that future studies can adopt to improve deforestation forecasting, we highlight the use of a CLUE approach and the improvement of some databases used.

Keywords: deforestation model; Random Forest; INLA.

LISTA DE FIGURAS

Figura 1 - Esquema de construção de uma Random Forest. 13
Figura 2 - Exemplo de Kernel Fixo (esquerda) e Kernel Adaptativo (direita) – Adaptado de Taylor et al., 2019).....15
Figura 3: Esquema geral da base de dados 18
Figura 4: Exemplo da criação de espaço celular com tamanho 25 x 25 Km 18
Figura 5: Limites da Amazônia Legal e Amazônia Bioma utilizados por Jaffe et al. (2021), bem como exemplos de células que tiveram seus atributos calculados incorretamente pela confusão dos limites. 19
Figura 6: Exemplo do cálculo do desmatamento entre os anos de 2015 e 2016. 21
Figura 8: Exemplo da abordagem CLUE-S utilizada por Aguiar et al. (2016) para modelagem futura do desmatamento. 25
Figura 9: Calibração do parâmetro mtry para valores de 1-30 (valor ótimo = 7).....26
Figura 10: Calibração da profundidade de cada árvore da floresta (valor ótimo = 300) 26
Figura 11: Valor do número de árvores na floresta em relação ao RMSE (valor ótimo = 1200) 27
Figura 11: Valor do número de árvores na floresta em relação ao RMSE.....27
Figura 12: Calibração do parâmetro de vizinhos mais próximos para o Spatial Random Forest (variação do kernel de 1-200) 28
Figura 13: Calibração do parâmetro de vizinhos mais próximos para o Spatial Random Forest (variação do kernel de 1-200) 29
Figura 14: Avaliação dos melhores atributos para a modelagem em Random Forest com base na métrica de impureza dos nos (GINI index) 31
Figura 16: Avaliação dos melhores atributos para a modelagem em Spatial Random Forest com base na métrica de impureza dos nos (GINI index) 31
Figura 16: Previsão de desmatamento para o ano de 2020 com dados do INLA.....33
Figura 17: Previsão de desmatamento para o ano de 2020 com dados do Random Forest. 34
Figura 18: Previsão de desmatamento para o ano de 2020 com dados do Spatial Random Forest. 35
Figura 19: Dados das Terras Indígenas (FUNAI) 42

Figura 20: Unidades de Conservação obtidas do MMA.....43
Figura 21: Shapefile de Hidrovias processado pela iniciativa do MapBiomas.....45
Figura 22: Dados de Corpos d'agua do IBGE. 46
Figura 23: IDH E PIB calculados por município dentro do bioma Amazonian. 50
Figura 24: Índice de aridez gerado pelo Google Earth Engine de acordo com a metodologia da FAO.....52
Figura 25: Dado de Altitude gerado no Google Earth Engine utilizando o SRTM com 90m de resolução espacial.....53
Figura 26: Uso e cobertura da iniciativa do Mapbiomas com classes agrupadas. ...55
Figura 27: Exemplo de cálculo para a variável Forest Edge Density: Polígono simplificado em marrom e a borda do raster de Floresta em verde. 58
Figura 28: Distancia de polos madeireiros para as células de 25 x 25 Km.....60
Figura 29: Distancia de Assentamentos rurais para as células de 25 x 25 Km 61

LISTA DE TABELAS

Tabela 1 - Atributos da base de dados (TV – TerraView).....20
Tabela 2 - Conversões consideradas como desmatamento.....21
Tabela 3 - Valores de RMSE para os conjuntos de treinamento, teste e validação para os 4 modelos desenvolvidos. 33

SUMÁRIO

1	**INTRODUÇÃO**	**08**
2	**REVISÃO BIBLIOGRÁFICA**	**11**
2.1	RANDOM FOREST	11
2.2	SPATIAL RANDOM FOREST	13
2.3	INTEGRATED NESTED LAPLACE APPROXIMATIONS	15
3	**MATERIAIS E MÉTODOS**	**15**
3.1	BASE DE DADOS	16
3.1.1	**Esquema geral da base de dados**	**16**
3.1.2	**Projeção e limites**	**18**
3.1.3	**Atributos da base de dados**	**19**
3.2	MODELAGEM DO DESMATAMENTO	21
3.2.1	**Remoção de atributos correlacionados**	**21**
3.2.2	**Treinamento e validação dos modelos Random Forest e Spatial Random Forest**	**22**
3.2.3	**Modelagem do tipo INLA**	**23**
4	**MODELAGEM PILOTO NA AMAZÔNIA**	**23**
4.1	REMOÇÃO DE ATRIBUTOS CORRELACIONADOS	23
4.2	CALIBRAÇÃO DOS MODELOS	24
4.2.1	Calibração dos modelos Random Forest	24
4.2.2	Calibração dos modelos Spatial Random Forest com kernel adaptativo	26
4.2.3	Calibração dos modelos Spatial Random Forest com kernel fixo	27
4.3	AVALIAÇÃO DOS ATRIBUTOS MAIS IMPORTANTES	28
4.4	RESULTADOS DA PREDIÇÃO E COMPARAÇÃO DE MODELOS DE DESMATAMENTO	31
5	**CONCLUSÕES E TRABALHOS FUTUROS**	**34**
	REFERÊNCIAS	**36**
	APÊNDICES	**39**

1 INTRODUÇÃO

O presente relatório contém resultados parciais do projeto “Definição de áreas prioritárias para recuperação florestal”, referentes a atividade “Uso e comparação da acurácia de diferentes modelos preditivos de desmatamento na Amazônia”.

O bioma Amazônia ocupa cerca de 6,7 milhões de km^2 na América do Sul, sendo 4,2 milhões de Km^2 do território nacional, e abrange a maior floresta tropical contínua do planeta (IBGE, 2019). A importância mundial deste bioma pode ser retratada em diversos pontos, como, por exemplo, armazenar mais de 123 Petagramas de carbono na sua vegetação e no solo, o que é mais do que o total de emissões de CO_2 pela queima de combustíveis fósseis em uma década (OBORN et al., 2011; MALHI et al., 2006), além de sua vegetação atuar como um sumidouro de carbono atmosférico, contribuindo para a regulação deste ciclo em todo o planeta (GATTI et al., 2021; PHILIPS et al., 2017; BRIENEN et al., 2015).

Outro ponto importante é o fato da bacia amazônica conter cerca de 20% da água doce do planeta (MMA, 2017), sendo que um terço de sua precipitação anual tem origem na própria bacia (STAAL et al., 2018; ARAGÃO, 2012). A Amazônia exerce um papel fundamental no ciclo hidrológico da América do Sul, influenciando a produtividade primária e agrícola de diversos ecossistemas, bem como a regulação de grande parte da energia hidroelétrica (MAURANO, 2018; SPRACKLEN et al., 2012). Sua importância biológica e sociocultural também é inestimável, contendo cerca de 10% de todas as espécies conhecidas mundialmente, abrigando cerca de 30 milhões de pessoas em mais de 350 grupos étnicos (WWF, 2016).

Mesmo com todas essas considerações, o governo brasileiro não cumpriu a promessa de redução no desmatamento na Amazônia na última década. Considerando a legislação vigente e atualizada, o país se comprometeu a reduzir em 80% os índices anuais de desmatamento na Amazônia Legal em relação à média verificada entre os anos de 1996 a 2005 (MMA, 2020a). Todavia, mesmo com a implementação dos Planos de Ação para Prevenção e Controle do Desmatamento (I, II e III), a redução foi estimada em apenas 44% (Silva Junior et al., 2020).

Para agravar ainda mais esta situação, os principais sistemas de monitoramento do desmatamento no país, como o Projeto de Monitoramento do Desmatamento na Amazônia por Satélites (PRODES) e o Sistema de Alerta de Desmatamento da Amazônia (SAD), desenvolvidos, respectivamente, pelo Instituto Nacional de Pesquisas Espaciais (INPE) e pelo Instituto do Homem e Meio Ambiente

da Amazônia (Imazon), apontam um aumento nas taxas de desmatamento anuais da Amazônia a partir de 2015 (INPE, 2021; IMAZON, 2021). O alerta de desmatamentos no Brasil, criado recentemente pela iniciativa do MapBiomas (MapBiomas Alerta), corrobora essa condição para os últimos 3 anos (MAPBIOMAS, 2021).

Neste sentido, a adoção de políticas públicas por empresas não governamentais que visam a conservação de remanescentes florestais e a recuperação de áreas degradadas são fundamentais para o Plano Nacional para Controle do Desmatamento Ilegal e Recuperação da Vegetação Nativa, divulgado em novembro de 2020 (MMA, 2020b). A Vale S.A. assumiu a meta voluntária de conservar e recuperar mais de 500.000 hectares de floresta até 2030.

O Instituto Tecnológico Vale (ITV) está contribuindo com esta iniciativa identificando áreas prioritárias para conservação ou restauração. Entre as iniciativas, estão sendo implementados diferentes modelos preditivos de desmatamento por meio de técnicas de análise espacial, visando predizer quais os locais mais críticos em termos de desmatamentos futuros.

A construção destes modelos requer, inicialmente, uma avaliação dos fatores que levam ao desmatamento (também chamados de *deforestation drivers*). Ometto et al. (2011) apontam que os principais fatores de desmatamento na Amazônia são a expansão de pastagens e de lavouras de soja, a intensificação agrícola, a baixa presença de tecnologia no setor madeireiro, a conectividade com mercados exportadores, a expansão da infraestrutura local e a presença de estradas (pavimentadas ou não).

Todavia, estudos recentes mostram que novos fatores podem estar associados a intensificação do desmatamento, como a concessão de crédito rural (JUSYS, 2016), a privatização e associação de pequenas comunidades rurais a grandes produtores (RAVIKUMAR et al., 2017; BENNET et al., 2018) e a intensificação do uso do fogo e mudança do uso da terra em áreas indígenas (OLIVEIRA et al., 2020). Todos os estudos mencionados anteriormente, contam com um processo extenso de compatibilização espaço-temporal de diversas bases de dados geográficos.

Com uma base de dados compatibilizada e consistente, podem ser utilizadas técnicas de análise espacial para avaliação de drivers de desmatamento e também predição dos locais de novos desmatamentos. Neste sentido, destacam-se os modelos baseados em regressões logísticas que consistem em adicionar ou remover iterativamente dados preditores (os fatores de desmatamento), a fim de encontrar o

melhor subconjunto de variáveis para a predição de uma variável alvo (áreas de desmatamento). Uma das abordagens mais comuns de construção de modelos de regressão é chamada de *Stepwise*, a qual consiste na construção de um modelo com os preditores de entrada mais importantes e posteriormente remover preditores com base em análises estatísticas e de desempenho do modelo (JAMES et al., 2014; BRUCE et al., 2020).

Também podem ser utilizados modelos do tipo GAM (*Generalized Additive Models*), os quais contém uma função espaço-temporal para ajustar os preditores. Estas funções podem ser relações lineares ou não lineares entre variáveis, bem como uma relação com seus vizinhos mais próximos (WOOD et al., 2017). Considerando a inclusão de variáveis temporais, principalmente informações sobre o passado, os modelos de GAM apresentaram um desempenho superior a regressão linear (TOH et al., 2020).

Por sua vez, os modelos baseados em técnicas de *machine learning*, como árvores de decisão, por exemplo, visam a descoberta de regras de classificação para uma variável alvo por meio da subdivisão de um conjunto de dados que está sendo analisado (WITTEN et al., 2011). Já as redes neurais artificiais são compostas por uma camada de entrada (preditores) e uma camada de saída (variável alvo) conectadas por diversas camadas intermediárias compostas por uma função de ativação. As redes neurais são treinadas para descobrir a variável alvo, realizando um processo de retropropagação de erro e ajustando valores matemáticos das camadas intermediárias a cada iteração (HAYKIN et al., 2009).

Desta maneira, o presente estudo tem como objetivo comparar os resultados de predição do desmatamento e drivers de desmatamento utilizando diferentes modelos. Para tanto, foram definidos os seguintes objetivos específicos:

- Gerar uma base de dados integrada e compatibilizada contendo os dados geográficos de diversos fatores que podem influenciar no desmatamento.

- Desenvolvimento de Scripts na Linguagem de programação R para a análise espacial e parametrização de modelos do tipo Random Forest (RF), Spatial Random Forest (SpRF) e Integrated Nested Laplace Approximations (INLA)

- Comparar os resultados de predição do desmatamento e drivers de desmatamento para um espaço celular de 25x25 km no Bioma Amazônia para os anos de 2016 até 2020.

- Avaliação dos diferentes impactos do uso de dados de desmatamento provenientes de fontes com diferentes metodologias de cálculo para a região piloto de Altamira/PA.

2 REVISÃO BIBLIOGRÁFICA

Este capítulo contém uma revisão bibliográfica sobre os métodos de modelagem utilizados neste trabalho.

2.1 RANDOM FOREST

O algoritmo de Random Forest é um algoritmo de classificação ou regressão baseado na construção de Árvores de Decisão ou Regressão, respectivamente. Uma árvore de decisão é uma técnica de mineração de dados utilizada para descobrir regras de classificação para um atributo a partir da subdivisão dos dados em um conjunto que está sendo analisado. Árvores de Decisão são simples representações de conhecimento e classificam exemplos em um número finito de classes (APTE; WEISS, 1997). Elas podem ser representadas graficamente por nós e ramos, semelhante a uma árvore no sentido invertido (WITTEN et al., 2011). Sua representação visual torna mais fácil para o usuário analisar e compreender os resultados (FAYYAD et al., 1996).

O nó raiz é o primeiro nó da árvore, localizado no topo da estrutura. Os nós internos, incluindo o nó raiz, são nós de decisão. Cada um destes contém um teste sobre um atributo e seus resultados formam os ramos da árvore. Cada regra tem início no nó raiz da árvore e caminha até uma de suas folhas (REZENDE, 2002). Os algoritmos que constroem Árvores de Decisão buscam encontrar aqueles atributos e valores que provêm máxima segregação dos registros de dados, com respeito ao atributo que se quer classificar, a cada nível da árvore.

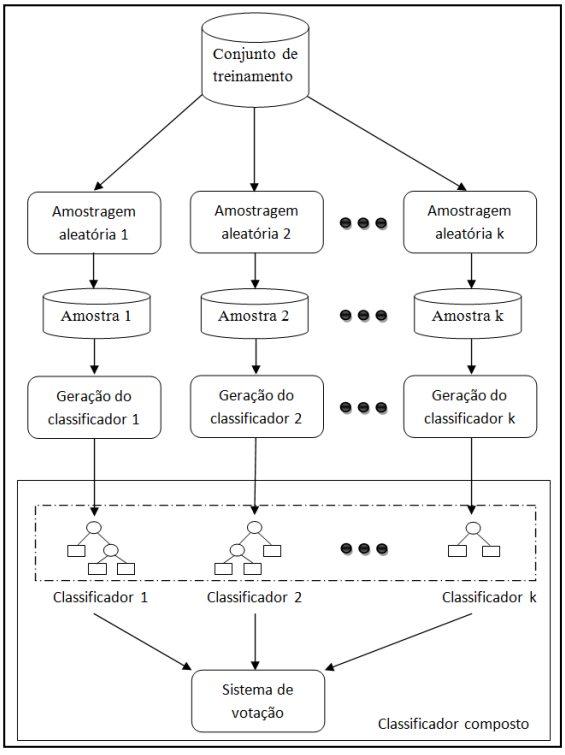
Random Forest é definido por Breiman (2001) como um classificador composto por uma coleção de Árvores de Decisão $\{h_k(x)\}$, $k=1,2,...,L$, onde T_k são amostras aleatórias independentes e identicamente distribuídas e cada árvore vota na classe mais popular para uma entrada X. Cada árvore de decisão é gerada a partir de um novo conjunto de atributos selecionado aleatoriamente por uma técnica chamada Bootstrap.

Bootstrap é uma técnica de amostragem com reposição. Para um conjunto de treinamento inicial T, são selecionados aleatoriamente exemplos para um novo

subconjunto de treinamento T_k. Cada subconjunto gerado possui o mesmo tamanho do conjunto original e é utilizado para gerar uma Árvore de Decisão $\{h_k(x)\}$. Normalmente, o subconjunto T_k contém, em média, 63,2% dos exemplos[1] do conjunto original T. Os exemplos que não foram utilizados no subconjunto T_k formam o conjunto out-of-bag, que são exemplos que não foram utilizados na construção do classificador h_k

Cada árvore de decisão é construída usando esse novo subconjunto. A cada nó da árvore, um número aleatório m de atributos é selecionado. O melhor atributo é escolhido, normalmente, em função do ganho de informação, para dividir o nó. Este procedimento é repetido para os demais nós da árvore, que cresce sem poda. Um número k de árvores é gerado, formando a floresta (Figura 1).

Figura 1 - Esquema de construção de uma Random Forest.



Fonte: Adaptado de Oshiro (2013).

O erro de classificação do Random Forest depende da força individual de cada árvore na floresta, ou seja, uma medida de desempenho para cada árvore, onde

[1] A probabilidade de uma amostra do conjunto T ser selecionada para o conjunto T_k é dada por $1 - \left(1 - \frac{1}{n}\right)^n$. Para um valor de n muito grande essa equação é, aproximadamente, $1 - \frac{1}{e} \approx 63{,}2\%$ (DIETTERICH, 2000).

árvores com taxa de acerto maior tem uma força individual maior, reduzindo o erro de classificação. A correlação entre estas árvores construídas também está relacionada com o erro de classificação, uma vez que a construção de diversas árvores e a seleção aleatória de atributos para cada nó são responsáveis por gerar árvores que sejam diferentes, diminuindo a correlação entre elas e a baixa correlação tende a diminuir o erro de classificação.

O algoritmo de Random Forest pode lidar com conjuntos com um grande número de atributos. O uso da amostragem bootstrap torna o algoritmo mais poderoso do que uma simples árvore, apresentando boa taxa de acerto quando testado em diferentes conjuntos de dados (BELLE, 2008). A combinação de diversos modelos em Árvores de Decisão tende a uma melhor taxa de acerto, visto que o erro em uma única classificação é sobreposto pela combinação de múltiplas classificações (SESNIE et al., 2010; COSTA, 2014). As Random Forests são computacionalmente eficientes, além de evitarem o sobreajuste (overfitting) e serem menos sensíveis a ruídos (BREIMAN, 2001).

O Random Forest utiliza três parâmetros que devem ser calibrados: a profundidade das árvores, indicada pelo número de níveis da árvore, o número de atributos utilizados em cada nó da árvore e o número de árvores de decisão da floresta.

2.2 SPATIAL RANDOM FOREST

O algoritmo chamado de Spatial Random Forest (SpRF) ou Geographically Weighted Random Forest (GWRF) consiste em uma extensão espacial do algoritmo Random Forest (GEORGANOS et al., 2018). O algoritmo foi adaptado com conceitos da GWR (Geographically Weighted Regression), a qual consiste em um método estatístico que considera relações de dependência espacial entre as variáveis do problema. A equação 1 descreve um problema genérico da GWR:

$$Y_i = \beta_{i0} + \sum_{k=1}^{m} \beta_{ik} * X_{ik} + \varepsilon_i \quad (1)$$

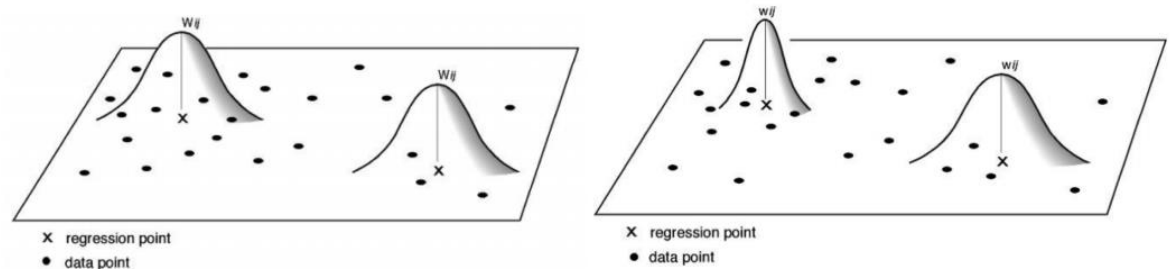
Onde, Y_i é a variável dependente na localização i, β_{i0} é o valor do intercepto na localização i, X_{ik} é o vetor correspondente a {1, ... ,m} variáveis independentes na localização i e ε_i e o erro na localização i.

A localização i, consiste em uma matriz quadrática com n x n observações e contendo m variáveis independentes (GOLLIN et al., 2003). A partir da matriz i podem ser extraídas relações de uma observação com os seus vizinhos, por meio de uma função de distância chamada de função W, a qual normalmente considera a distância euclidiana como relação principal (GAO; LI, 2011).

Desta maneira torna-se possível adaptar o algoritmo do Random Forest com a construção de um conjunto de árvores de decisão $\{h_{k,i}(x)\}$ para cada localização i incluindo n vizinhos, de acordo com a função W derivada da GWR. Desta maneira, cada observação do conjunto de dados tem seu próprio conjunto de árvores gerados, com seu respectivo poder de predição (GEORGANOS et al., 2018).

A implementação do Spatial Random Forest no Software R foi realizada por Kalagirou e Georganos (2018) por meio do pacote chamado de SpatialML. Neste pacote a função W é chamada de Kernel e pode receber um valor fixo ou adaptado. O valor adaptado consiste em incluir um número fixo de vizinhos, independente da distância que estes estão da localização i, o que pode ser útil quando há a descontinuidade dos dados geográficos, como a presença de ilhas ou a má distribuição amostral no espaço. Já o valor fixo trabalha dentro um raio regular passado pelo usuário e considera todos os vizinhos que estão inclusos neste raio. A Figura 2 mostra um exemplo de como a função W se comporta para essas duas situações.

Figura 2 - Exemplo de Kernel Fixo (esquerda) e Kernel Adaptativo (direita).



Fonte: Adaptado de Taylor et al. (2019)

As principais vantagens da extensão espacial do algoritmo Random Forest é um maior poder de predição do que o algoritmo tradicional, justamente por considerar a influência da vizinhança, permitir trabalhar com a heterogeneidade espacial e também por avaliar a importância local de cada variável, justamente por criar um conjunto de árvores de decisão para cada ponto amostral (GEORGANOS et al., 2018; SANTOS et al., 2019)

2.3 INTEGRATED NESTED LAPLACE APPROXIMATIONS

Os modelos do tipo “Integrated Nested Laplace Approximations” (INLA) se baseiam na técnica de inferência bayesiana para trabalhar com os Modelos Gaussianos Latentes – LGM (RUE et al., 2008; RUE et al., 2017). Utilizando um esquema complexo de integração numérica e combinações analíticas, os modelos do tipo INLA podem gerar aproximações determinísticas altamente precisas para obter resultados dos modelos LGM.

Os modelos LGM podem ser resolvidos utilizando-se cadeias de Markov com a repetição de Monte Carlo, todavia a inferência bayesiana dos modelos INLA pode ser considerada computacionalmente mais eficiente (MARTINO; RIEBLER, 2019). O equacionamento destes modelos pode ser consultado nas seguintes referências (de acordo com o tema em questão):

- Cadeias de Markov (RUE; HELD, 2005);
- Inferência Bayesiana do INLA (GOMES-RUBIO, 2020);
- Aplicações espaciais com modelos INLA (KRAINSKY et al., 2019; MORAGA et al., 2020; BLANGIARDO; CAMELETTI, 2015).

Os modelos do tipo INLA estão implementados no software estatístico R (https://www.r-project.org/) por meio do pacote R-INLA (https://www.r-inla.org/). Este pacote representa uma versão amigável e versátil para executar a modelagem com o INLA. Após a implementação, o pacote retorna todos os parâmetros do modelo, bem como as informações marginais e um resumo correspondente. Uma série de tutoriais de como utilizar o R-INLA podem ser encontrados nas referências sobre aplicações espaciais com modelos INLA.

3 MATERIAIS E MÉTODOS

Essa seção apresenta um detalhamento da construção da base de dados, apresentando atualizações à base de dados utilizada por Jaffe et al. (2021). Os scripts

desenvolvidos para realizar a modelagem do desmatamento também são compartilhados e comentados nesta seção.

3.1 BASE DE DADOS

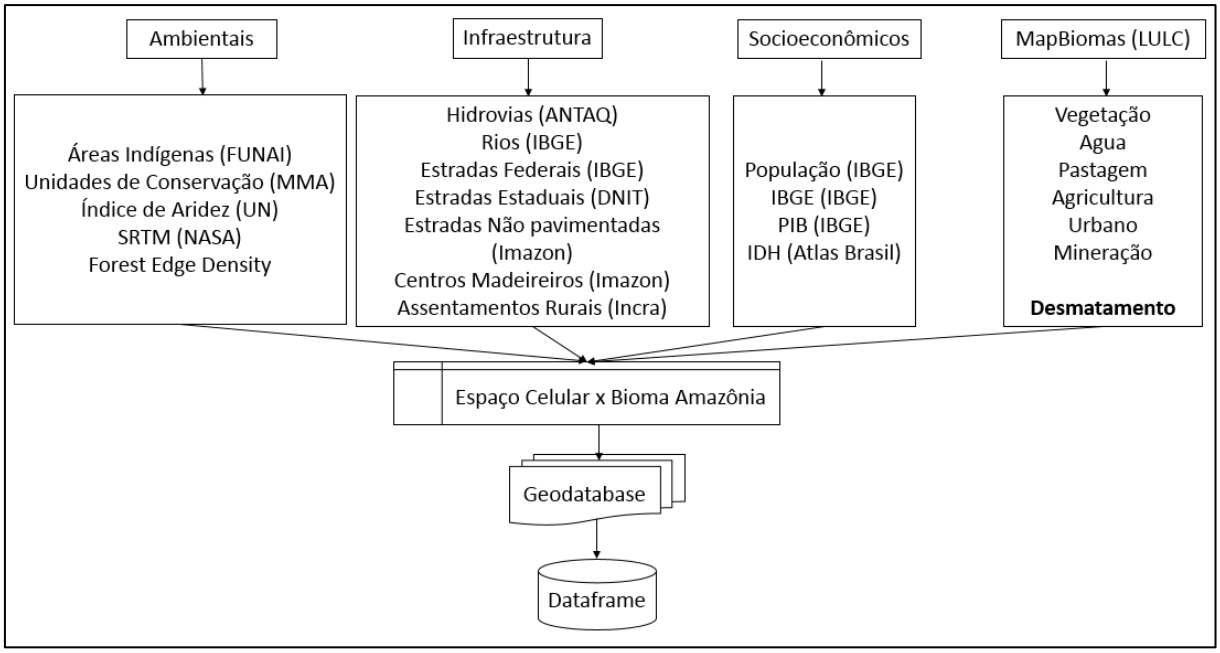
A base de dados inicial foi construída de acordo com Jaffe et al. (2021), e). Os códigos e dados estão disponíveis nos links disponibilizados pelos autores. Esta base foi atualizada visando a melhor consistência dos dados para uso na predição do desmatamento.

3.1.1 Esquema geral da base de dados

O fluxograma utilizado para montar a base de dados encontra-se na Figura 3. Foram utilizados dados ambientais, de infraestrutura, socioeconômicos e de uso e cobertura da terraterra. Estes dados foram obtidos de diversas instituições como a Fundação Nacional do Índio (FUNAI), o Ministério do Meio Ambiente (MMA), O Instituto Brasileiro de Geografia e Estatistica (IBGE), dentre outros. Os dados de uso e cobertura da terraterra foram obtidos por meio dos mapas do MapBiomas (https://mapbiomas.org/en).

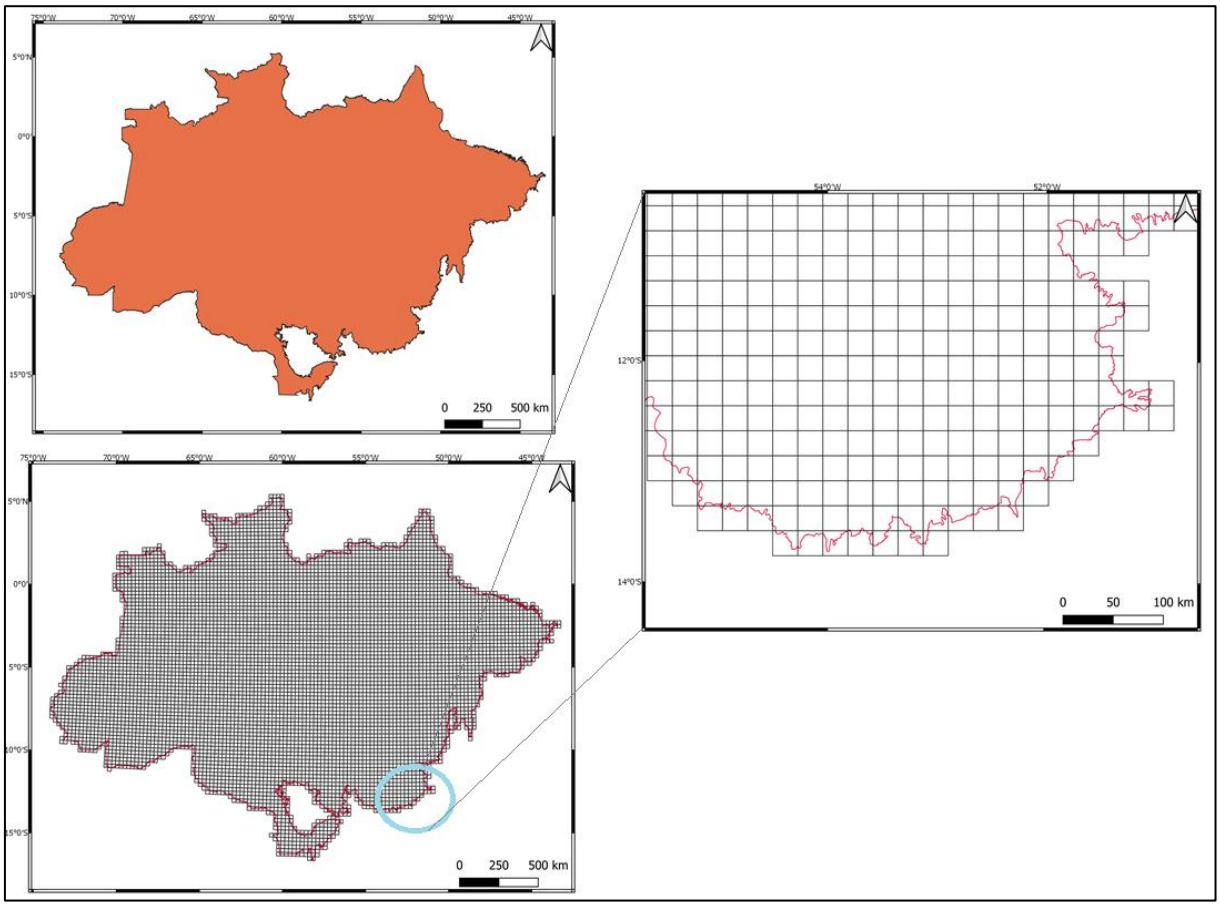
A grade de processamento consiste na subdivisão do bioma Amazônia em células de tamanho fixo (Figura 4). Considerando a demanda computacional, optou-se por utilizar células de 25 x 25 km.

Figura 3 - Esquema geral da base de dados



Fonte: elaboração própria (2021).

Figura 4 - Exemplo da criação de espaço celular com tamanho 25 x 25 Km

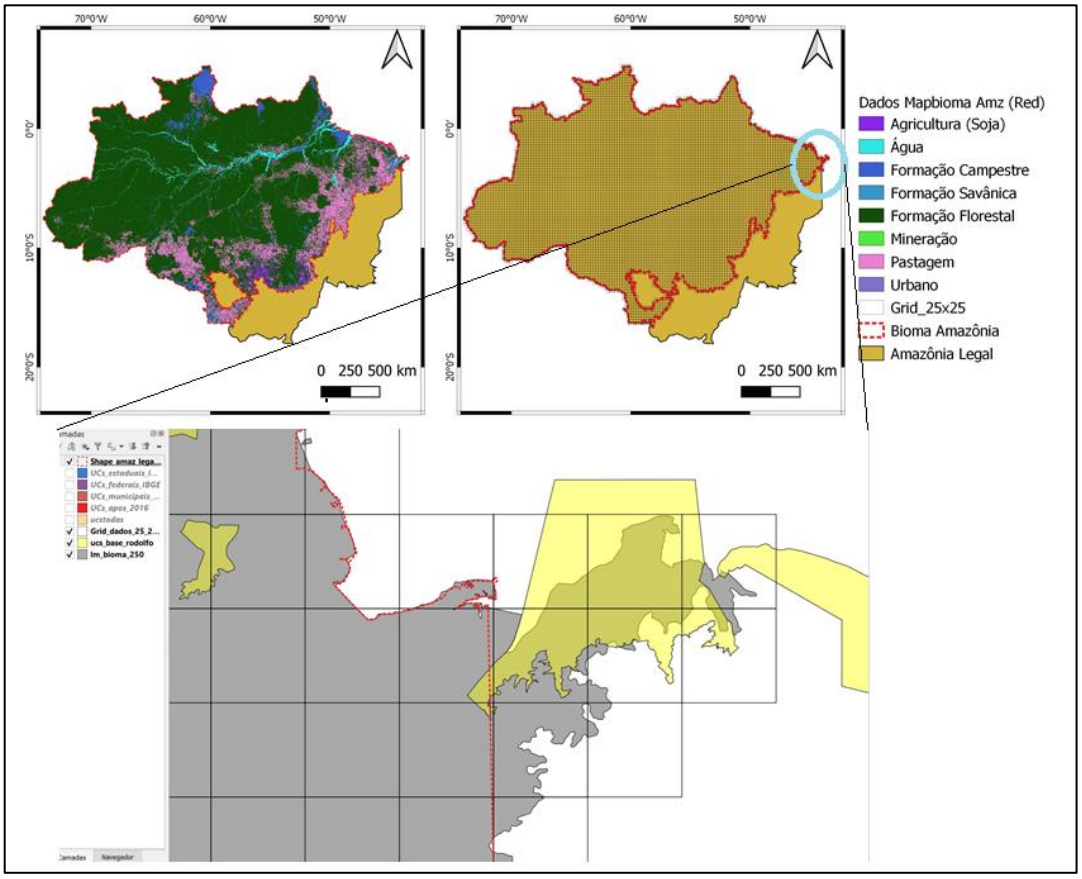


Fonte: elaboração própria (2021).

3.1.2 Projeção e limites

A base de dados foi montada a partir da projeção SIRGAS 5880 (https://epsg.io/5880) com distorção de 1m e coordenadas UTM. A base de dados utilizada por Jaffe et al. (2021) foi montada com base na projeção WGS84, sendo que as distâncias e áreas em graus foram transformadas para metros por meio de uma transformação simples, considerando a distância de 1 grau na latitude 0 (equador). Desta maneira, todas as células que estavam fora desta latitude apresentavam distorções em suas áreas. Outra atualização da base de dados foi o uso do limite do Bioma Amazônia para padronizar todos as camadas. A base anterior utiliza o limite da Amazônia Legal brasileira em algumas camadas, gerando algumas células sem dados no nordeste da Amazônia (Figura 5).

Figura 5 - Limites da Amazônia Legal e Amazônia Bioma utilizados por Jaffe et al. (2021), bem como exemplos de células que tiveram seus atributos calculados incorretamente pela confusão dos limites.



Fonte: elaboração própria (2021).

3.1.3 Atributos da base de dados

Os atributos da base de dados estão descritos na Tabela 1 e foram calculados para os anos de 2015 até 2019 (individualmente). Os metadados e descrição de cálculo de cada atributo estão no apêndice A.

Tabela 1 **- Atributos da base de dados (TV – TerraView)**

Dado	Nome	Software	Unidade
Área Terras Indígenas	Ar_IL	Qgis/R	Km
Distância Terras Indígenas	Dst_IL	Qgis/R	Km
Porcentagem Terras Indígenas	Pc_IL	Qgis/R	% [0-100]
Área UC	Ar_PA	Qgis/R	Km^2
Distância UC	Dst_UC	Qgis/R	Km
Porcentagem UC	Pc_UC	Qgis/R	% [0-100]
Soma Hidrovias	Sum_Wat	Qgis/R	Km
Distância Hidrovias	Dst_Wat	Qgis/R	Km
Soma Rios	Sum_Riv	TV/Qgis/R	Km
Distância Rios	Dst_Riv	TV/Qgis/R	Km
Soma Estradas Federais + Estaduais	Sum_FR_SR	Qgis/R	Km
Distância Estradas (Todas)	Dst_AR	Qgis/R	Km
IDH Médio (IDHM)	IDH	Qgis/R	[0-1]
IDH Renda (IDHM_Rn)	IDH_Rn	Qgis/R	[0-1]
IDH Educação (IDHMEdc)	IDH_Ed	Qgis/R	[0-1]
IDH Longevidade (IDHM_Ln)	IDH_Lg	Qgis/R	[0-1]
População	POP	Qgis/R	Numérico
PIB	GDP	Qgis/R	Numérico x1000
PIB Per Capta	GDPPC	Qgis/R	Numérico
Índice Aridez	ARI	GEE/Qgis	Numérico
Altitude	Elevation	GEE/Qgis/ArcGis	Numérico
Declividade	Slope	GEE/Qgis/ArcGis	Numérico
Proporção Natural	Pc_Nat	GEE/Qgis/ArcGis	% [0-100]

Proporção Água	Pc_Wat	GEE/Qgis/ArcGis	% [0-100]
Proporção Agricultura	Pc_Agr	GEE/Qgis/ArcGis	% [0-100]
Proporção Pastagem	Pc_Pst	GEE/Qgis/ArcGis	% [0-100]
Proporção Urbano	Pc_Urb	GEE/Qgis/ArcGis	% [0-100]
Proporção Mineração	Pc_Min	GEE/Qgis/ArcGis	% [0-100]
Distância Urbano	Dst_Urb	GEE/Qgis	Km
Distância Mineração	Dst_Min	GEE/Qgis	Km
Distância Pastagem	Dst_Pst	GEE/Qgis	Km
Distância Agricultura	Dst_Agr	GEE/Qgis[4]	Km
Forest Edge Density	FED	GEE/Qgis/ArcGis/R[4]	m/km^2
Distância P. Madeireiros	Dst_PM	Qgis	Km
Área Assentamentos	Ar_Ast	Qgis/R	Km^2
Distância Assentamentos	Dst_Ast	Qgis/R	Km
Porcentagem Assentamentos	Pc_Ast	Qgis/R	% [0-100]
Desmatamento	Def	Qgis/ArcGis[2]	% [0-100]

Fonte: elaboração própria (2021).

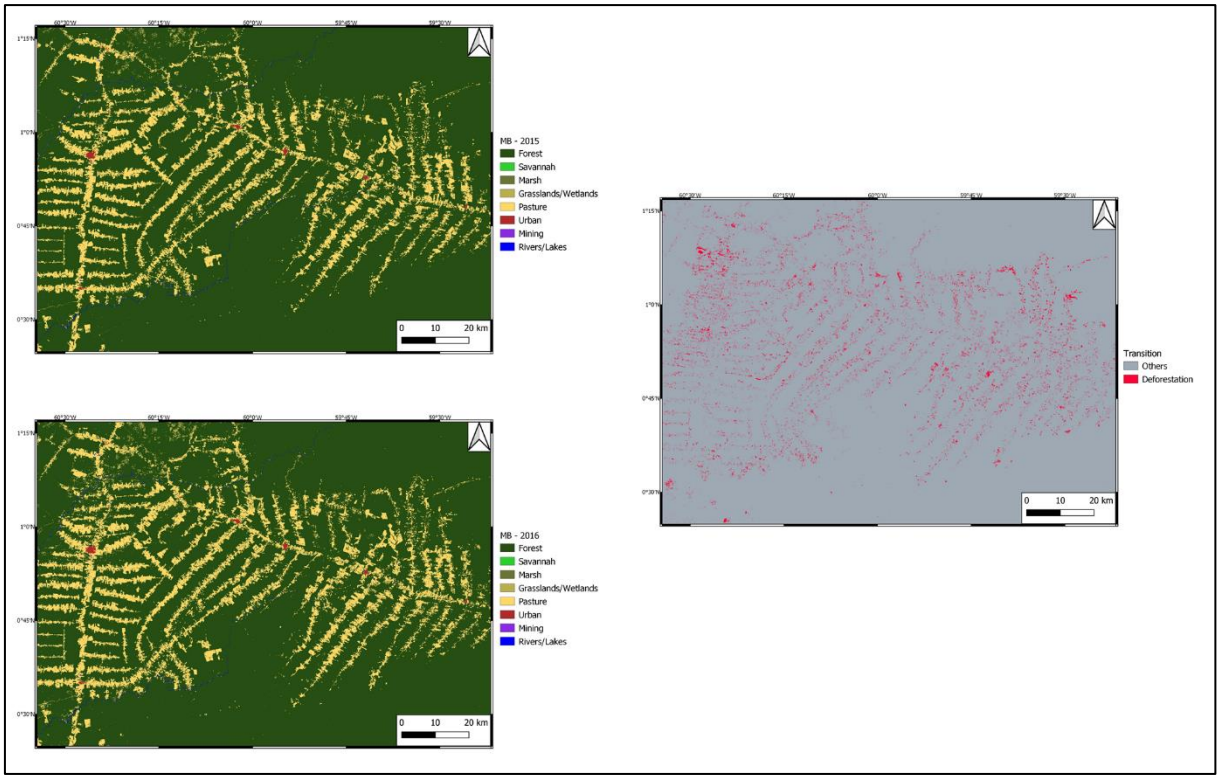
O desmatamento anual foi calculado por meio da diferença entre os mapas de uso e cobertura do MapbiomasMapBiomas (Figura 6). As conversões de uso e cobertura da terra (natural para antrópico) apresentadas na Tabela 2 foram consideradas como desmatamento.

Tabela 2 - Conversões consideradas como desmatamento

Natural **(t)**	→	Antrópico **(t+1)**
Floresta		Floresta Plantada
Savana		Pastagem
Áreas pantanosas		Urbano
Áreas alagadas		Mineração
Campestre		Cana
		Soja
		Outras lavouras temporárias

Fonte: elaboração própria (2021).

Figura 6 - Exemplo do cálculo do desmatamento entre os anos de 2015 e 2016.



Fonte: elaboração própria (2021).

3.2. Modelagem do desmatamento

Esta seção mostra as principais etapas da fase de modelagem, sendo que inicialmente foi realizada uma avaliação e remoção de atributos correlacionados do conjunto de dados. Posteriormente foi montada uma estratégia de treinamento similar para os modelos de Random Forest e Spatial Random Forest, gerando conjuntos de dados utilizados para treinamento, teste, validação e predição futura do desmatamento. Uma metodologia análoga foi utilizada para os modelos INLA, sendo que os resultados foram avaliados por meio de medidas estatísticas como o RMSE (*Root Mean Square Error*).

3.2.1 Remoção de atributos correlacionados

Para avaliação da correlação entre os atributos, foi utilizada a correlação de Pearson (Benesty et al., 2008), a qual pode ser descrita da seguinte formula:

$$Y_i = \frac{\sum_{i=1}^{n}(x_i - x_m) * (y_i - y_m)}{\sqrt{\sum_{i=1}^{n}(x_i - x_m)^2} * \sqrt{\sum_{i=1}^{n}(y_i - y_m)^2}} \quad (1)$$

Onde *n* representa o tamanho amostral (número de registros); x_i e y_i são os valores dos atributos para uma determinada amostra i; e x_m e y_m são a média amostral dos atributo x e y, respectivamente.

O software R foi utilizado para realizar essa avaliação com o pacote corrplot (https://cran.r-project.org/web/packages/corrplot/index.html).

3.2.2 Treinamento e validação dos modelos Random Forest e Spatial Random Forest

Foram gerados conjuntos de treinamento, teste, validação e predição para os modelos do tipo Random Forest e Spatial Random Forest.

Os conjuntos de treinamento são a base para que os modelos sejam desenvolvidos e aprendam as regras de classificação. Já o conjunto de testes é utilizado em um primeiro momento para avaliação das regras de classificação obtidas na etapa anterior, sendo utilizado para calibrar os parâmetros do classificador e realizar análises de quais são os atributos mais relevantes na modelagem.

O conjunto de validação consiste na avaliação das predições do modelo com um conjunto que não foi utilizado para o desenvolvimento do modelo, sendo que estas medidas são as medidas de desempenho mais confiáveis, evitando processos de sobreajuste no treinamento do modelo. Por fim, o conjunto de predição consiste em um conjunto de dados sem a variável resposta, apenas com as variáveis preditoras.

Neste projeto foi avaliado o desmatamento futuro considerando uma janela temporal de 1 ano, logo as condições preditoras do ano T_{-1} são utilizadas para determinar o desmatamento no tempo T_0, seguindo o recomendado por Jaffe et al. (2021). O conjunto de dados utilizado para treinar os modelos continha variáveis preditoras de 2016 com o desmatamento de 2017. O conjunto de teste tinha as variáveis preditoras de 2017 e o desmatamento de 2018. O conjunto de validação tinha as variáveis preditoras de 2018 e o desmatamento de 2019 e o conjunto de predição tinha as variáveis preditoras de 2019, visando obter o desmatamento do ano de 2020.

No processo de modelagem do Random Forest, ainda foi realizado um processo de calibração de variáveis internas do modelo, como o número total de

árvores de decisão da floresta (ntree), o número de atributos aleatórios por nó (mtry) e profundidade máxima da floresta (max_nodes). Por fim a seleção de atributos foi realizada seguindo as medidas de desempenho Mean Decrease Accuracy e Gini Impurity Index, descritas no pacote caret (https://topepo.github.io/caret/)

3.2.3 Modelagem do tipo INLA

A modelagem dou tipo INLA utilizou a mesma base de modelagem do Random Forest, utilizando os valores ótimos dos parâmetros de entrada obtidos por Jaffe et al. (2021).

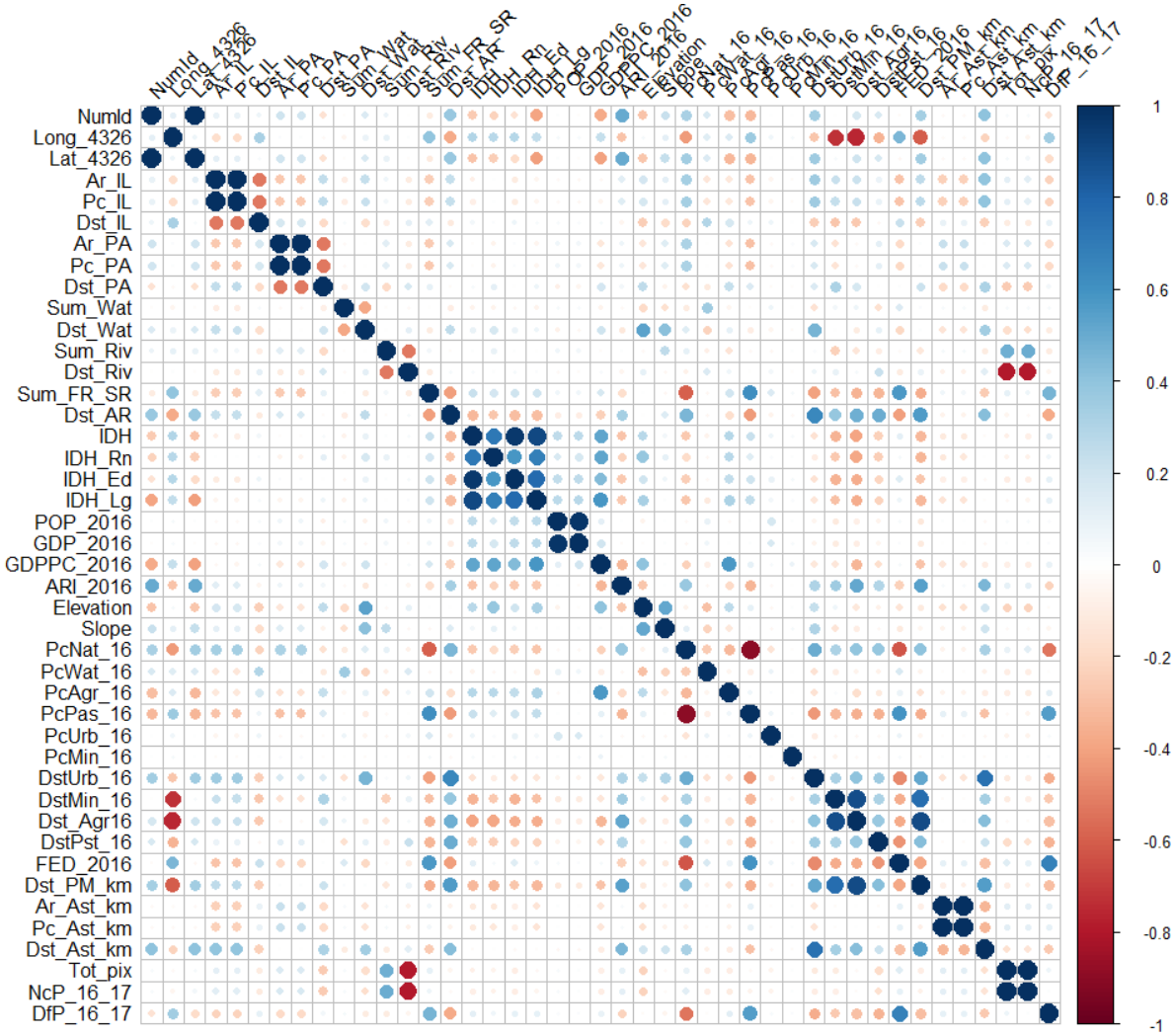
4 MODELAGEM PILOTO NA AMAZÔNIA

Essa seção apresenta os resultados preliminares de modelagem na Amazônia considerando o espaço celular de 25x25 km, todo o processo de calibração dos modelos, seleção de atributos e também a comparação entre 4 modelos de desmatamento, sendo eles do tipo INLA, Random Forest e Spatial Random Forest (gerado com 2 tipos de kernels).

4.1 REMOÇÃO DE ATRIBUTOS CORRELACIONADOS:

A avaliação de correlação entre os atributos foi responsável pela remoção de 3 atributos sendo eles a porcentagem de Terras Indígenas (Pc_IL), a porcentagem de áreas protegidas (Pc_PA) e a porcentagem de Assentamentos Rurais (Pc_AST). Estes atributos obtiveram correlação próxima a 1 com os seguintes atributos, respectivamente: total de área de terras indígenas (Ar_IL), total de área Áreas protegidas (Ar_PA) e total de área de Assentamentos Rurais (Ar_AST). Estes atributos foram removidos antes do processo de calibração dos modelos e avaliação de atributos mais importantes. A Figura 8 mostra a correlação entre todos os atributos do conjunto de dados.

Figura 8 - Exemplo da abordagem CLUE-S utilizada por Aguiar et al. (2016) para modelagem futura do desmatamento.



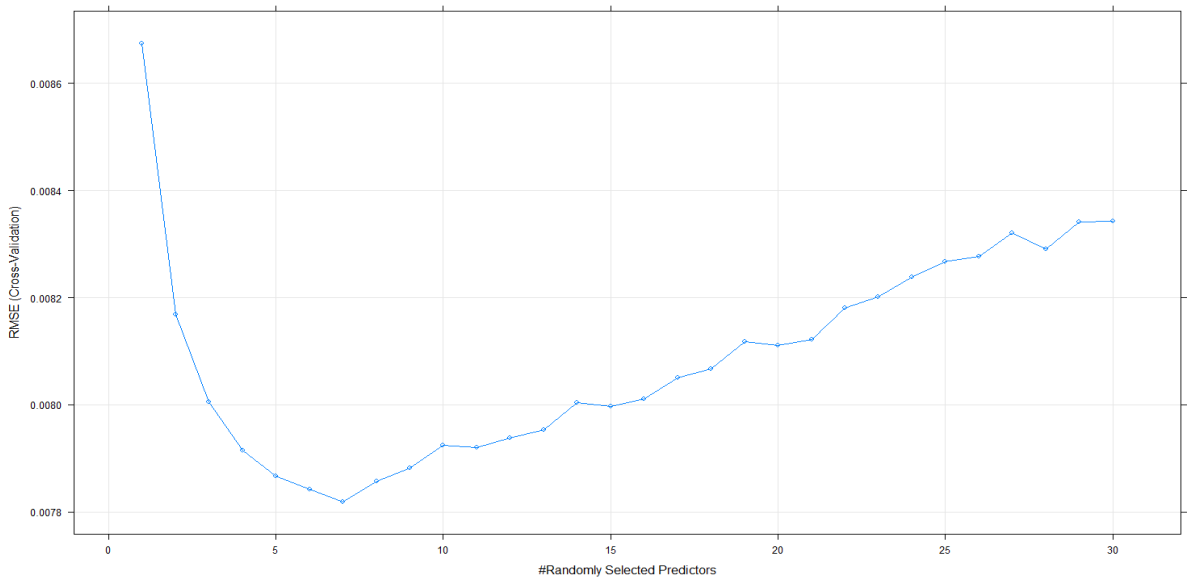
Fonte: elaboração própria (2021).

4.2 CALIBRAÇÃO DOS MODELOS

4.2.1 Calibração dos modelos Random Forest

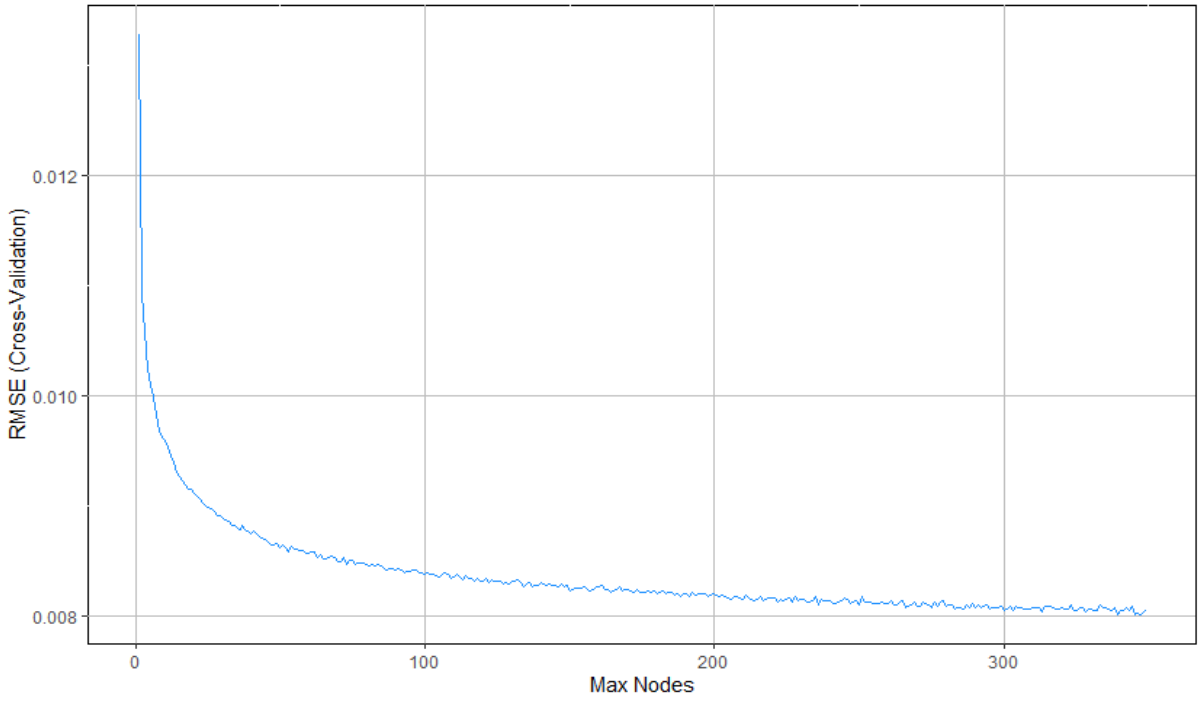
Os resultados da calibração para o modelo Random Forest estão nas Figuras 9, 10 e 11. O Processo de calibração demorou cerca de 26 horas e foi baseado nos menores valores de RMSE obtidos em cada teste. Foram avaliados valores de 1-30 para o parâmetro mtry (número aleatórios de atributos selecionados) e o melhor valor obtido foi de 7. Para os valores de profundidade da árvore (max_nodes) foram testados valores de 1-300, onde 300 significa que não é aplicado nenhum processo de poda em cada árvore de decisão gerada (melhor resultado). Já o atributo de número de árvoresárvores por floresta foi avaliado de 1 – 2000, sendo o valor ótimo fixado em 1200.

Figura 7 - Calibração do parâmetro mtry para valores de 1-30 (valor ótimo = 7)



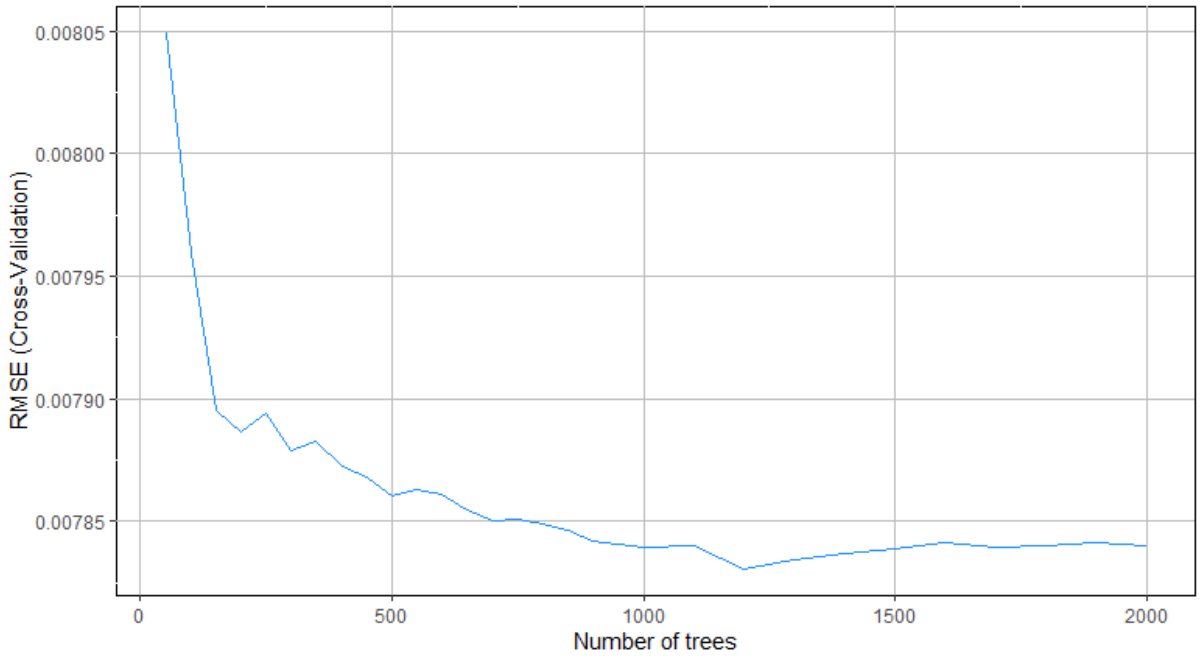
Fonte: elaboração própria (2021).

Figura 8 - Calibração da profundidade de cada árvore da floresta (valor ótimo = 300)



Fonte: elaboração própria (2021).

Figura 9 - Valor do número de árvores na floresta em relação ao RMSE (valor ótimo = 1200)

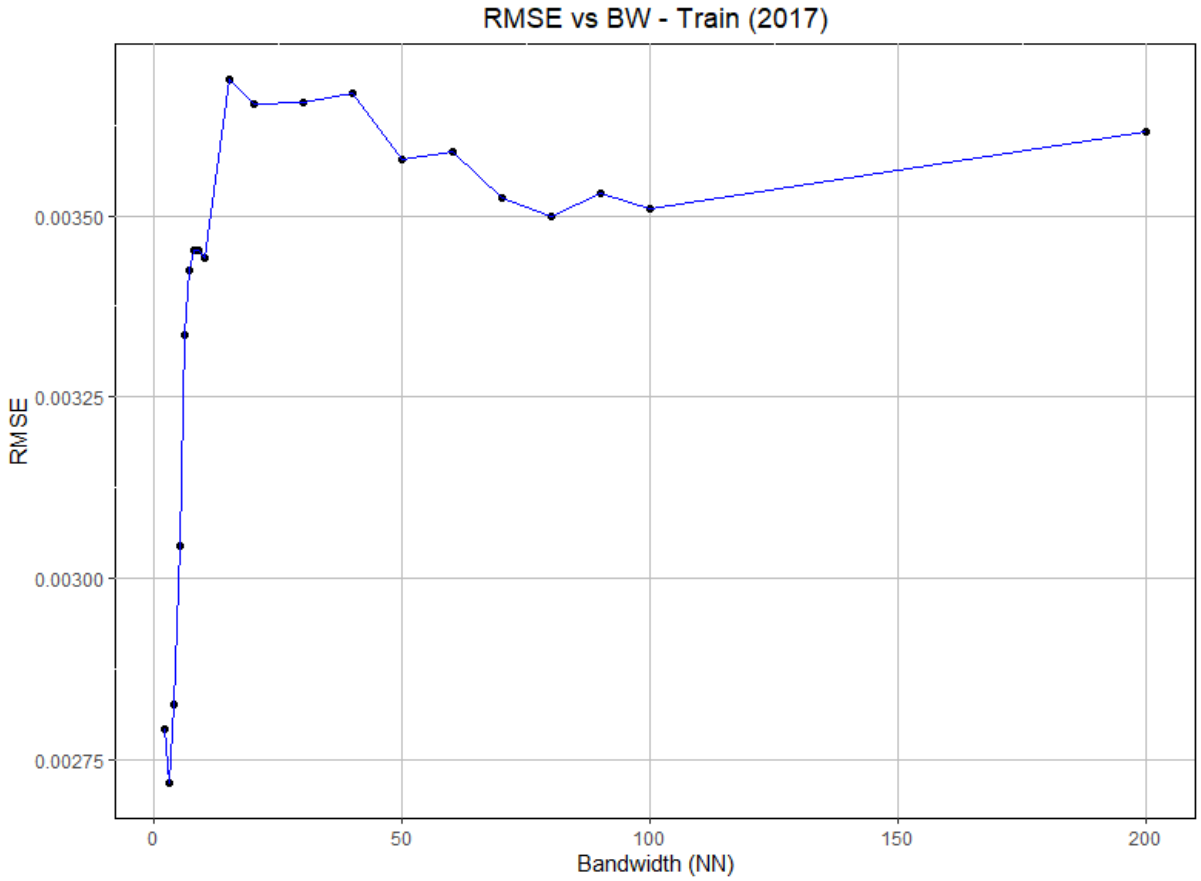


Fonte: elaboração própria (2021).

4.2.2 Calibração dos modelos Spatial Random Forest com kernel adaptativo

O resultado da calibração para o modelo Spatial Random Forest com Kernel adaptativo está na Figura 12. Para este modelo foram considerados os parâmetros ótimos da seção anterior, sendo que o número de vizinhos mais próximos (valor do kernel) foi calibrado de 1 a 200, sendo escolhido o valor em que o modelo apresentasse o menor RMSE. Os valores de 1-10 apresentaram RMSE muito baixos quando comparados com os demais, indicando que poderia haver um processo de overfitting nos modelos, por isso optou-se por trabalhar com os valores de 80 ou 100.

Figura 10 - Calibração do parâmetro de vizinhos mais próximos para o Spatial Random Forest (variação do kernel de 1-200)

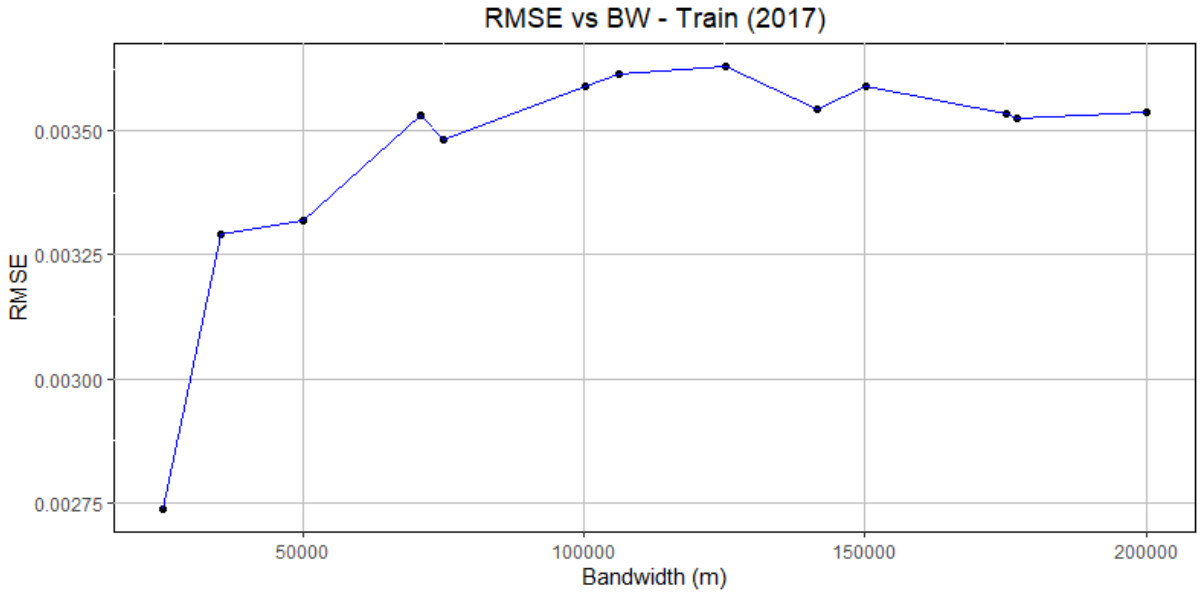


Fonte: elaboração própria (2021).

4.2.3 Calibração dos modelos Spatial Random Forest com kernel fixo.

O resultado da calibração para o modelo Spatial Random Forest com Kernel fixo está na Figura 13. Para este modelo foram considerados os parâmetros ótimos da seção anterior, sendo que o valor do kernel representa um raio, em metros, em que deverão ser considerados todos os vizinhos para cálculo do Spatial Random Forest. Foram avaliados valores de 25.000 até 200.000. O processo de calibração demorou cerca de 12 horas. O valor ótimo escolhido para teste foi 200.000, pelo mesmo motivo apresentado na seção anterior.

Figura 11 - Calibração do parâmetro de vizinhos mais próximos para o Spatial Random Forest (variação do kernel de 1-200)



Fonte: elaboração própria (2021).

4.3 AVALIAÇÃO DOS ATRIBUTOS MAIS IMPORTANTES:

Os melhores atributos para a modelagem foram avaliados com base no decréscimo dos valores de MSR (mean square error) causados pela remoção do atributo no processo de modelagem (Figuras 14 e 15). Os resultados para o modelo Spatial Random Forest foi o mesmo quando considerado o kernel fixo ou adaptativo.

Como resultados, três atributos foram identificados entre os quatro mais importantes em ambos os modelos (Random Forest e Spatial Random Forest). Um deles foi a presença e o avanço da pastagem como forma de uso do solo, conforme analisado por Ometto et al. (2015). Este atributo está representado neste estudo pela variável Pc_Pst (Porcentagem de pastagem em uma célula) e também foi apontado como um dos atributos mais importante para modelagem do desmatamento.

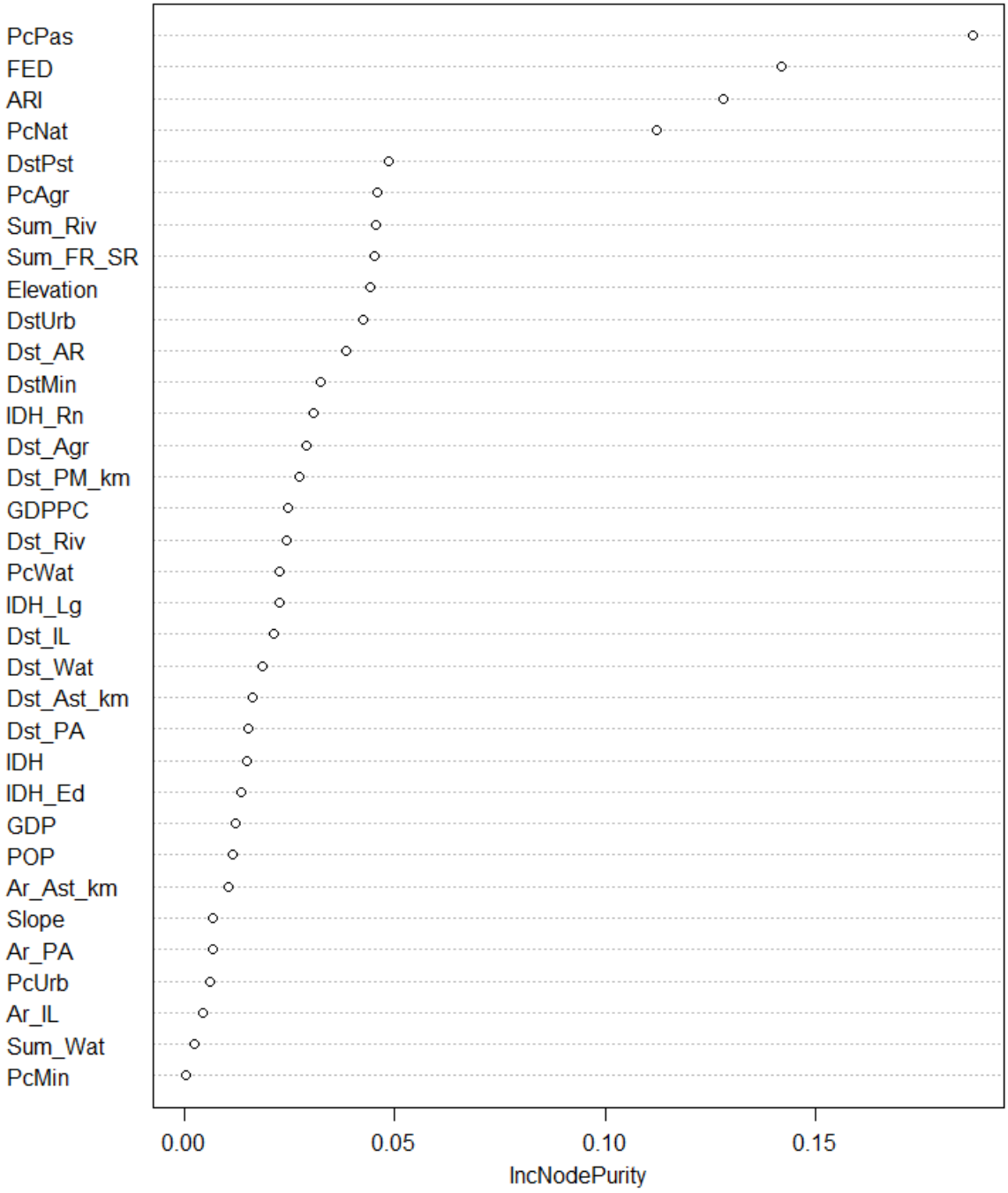
Outro atributo apontado como importante foi o FED (Forest Edge Density), o qual representa um perímetro de floresta em relação a área de uma célula. Conforme mencionado por Jaffe et al. (2021) este atributo é uma medida de fragmentação da paisagem, sendo que paisagens mais desmatadas e mais fragmentadas tendem a continuar este processo e, consequentemente, gerar mais desmatamento.

Considerando as mudanças de uso e cobertura do solo apresentados pelo projeto MapBiomas, a principal alteração do uso e cobertura do solo na Amazônia são causadas pela conversão de florestas em pastagens. Desta maneira o atributo de

Pc_Nat (porcentagem de áreas naturais em uma célula) funciona como inverso do atributo Pc_Pst.

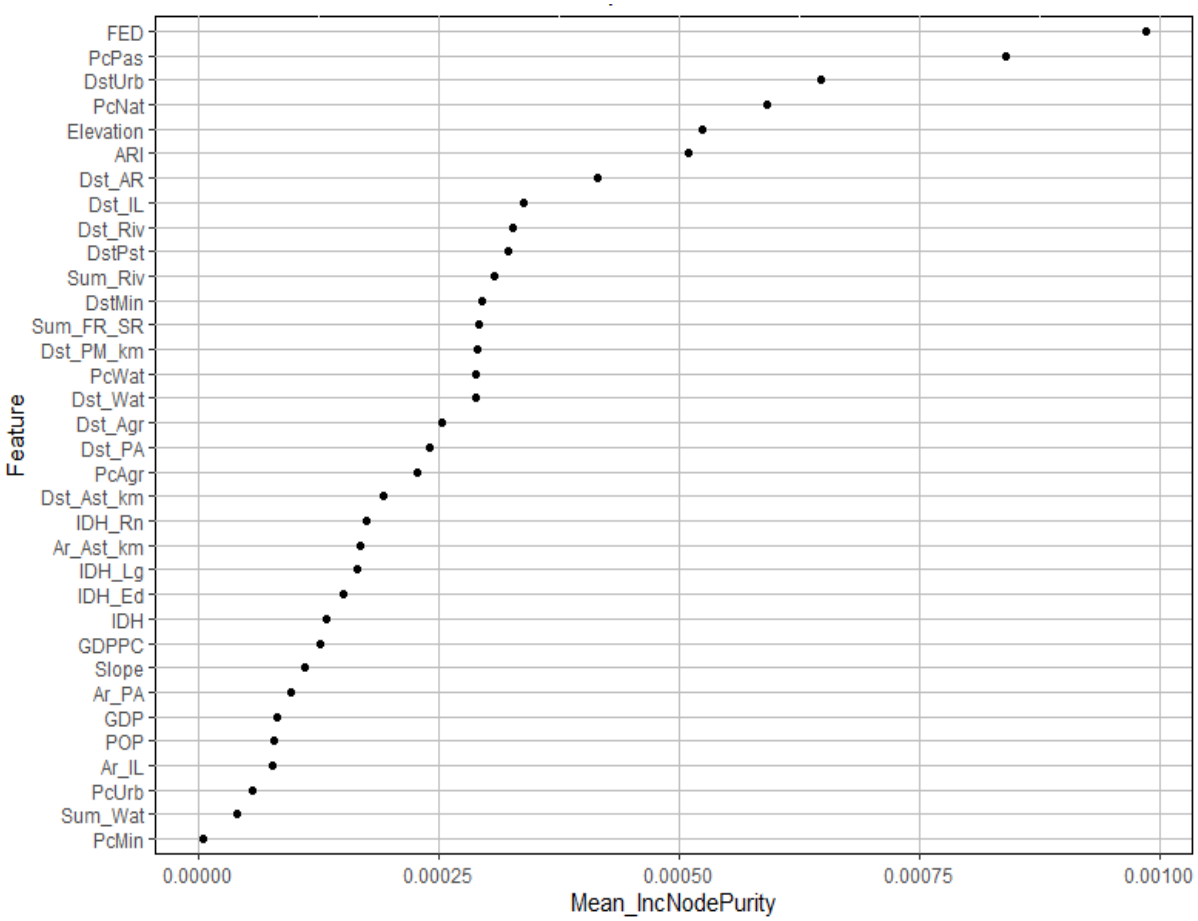
O índice de aridez (ARI) também apareceu como um importante fator em ambos os modelos, enquanto outras variáveis se destacaram dependendo do modelo. Os atributos relacionados as estradas (distância e soma), comumente apontados como um importante driver de desmatamento, ficaram em 11º e 8º lugares para o RF, e em 7º e 13º para o SpRF (de 34 atributos). O uso de informações atualizadas de estradas, que incluam as estradas não pavimentadas e vicinais, pode melhorar a modelagem da influência desta variável no desmatamento.

Figura 12 - Avaliação dos melhores atributos para a modelagem em Random Forest com base na métrica de impureza dos nos (GINI index)



Fonte: elaboração própria (2021).

Figura 13 - Avaliação dos melhores atributos para a modelagem em Spatial Random Forest com base na métrica de impureza dos nos (GINI index)



Fonte: elaboração própria (2021).

4.4 RESULTADOS DA PREDIÇÃO E COMPARAÇÃO DE MODELOS DE DESMATAMENTO.

A Tabela 3 apresenta os valores do erro médio quadrático para os 4 modelos desenvolvidos e para todos os conjuntos de dados analisados (treinamento, teste e validação). Estes valores são analisados para comparar os modelos em diversas situações de performance, sendo que o valor para o conjunto de validação pode ser considerado o mais importante.

No treinamento, o modelo INLA obteve um RMSE de 0, devido a este ser desenvolvido com base nos valores preditores do conjunto de treinamento. Todavia, quando os conjuntos de teste e validação foram utilizados para gerar predições do desmatamento, o RMSE foi o pior dentre todos os modelos. Esse resultado mostra que este tipo de modelagem está aquém de outras técnicas em relação a modelagem do desmatamento da Amazônia e seus resultados devem ser analisados com cautela.

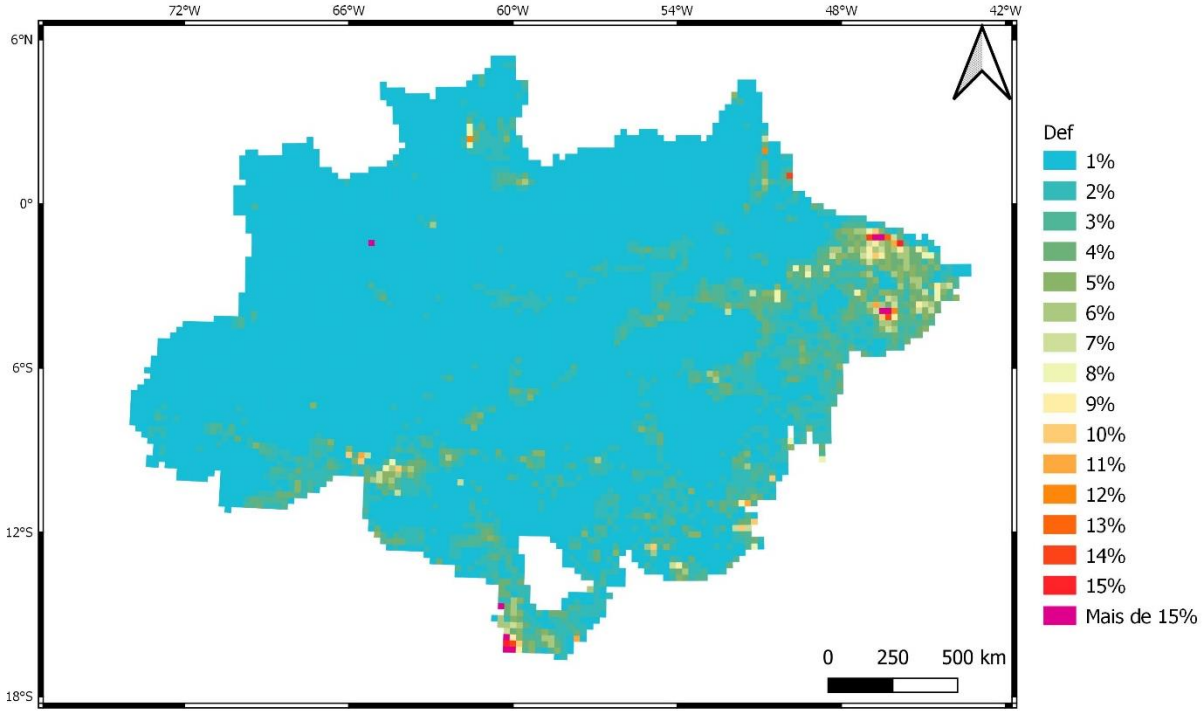
Os modelos do tipo INLA predisseram um desmatamento na Amazônia de 35.123 km² para 2020(Figura 16).

Tabela 3 - Valores de RMSE para os conjuntos de treinamento, teste e validação para os 4 modelos desenvolvidos.

Ano	Random forest	Spatial random forest (adaptativo)	Spatial random forest (fixo)	INLA
2017 - Treinamento	8,0	3,6	3,5	0,0
2018 - Teste	11,4	11,7	11,9	15,9
2019 - Validação	8,8	9,3	9,5	13,5

Fonte: elaboração própria (2021).

Figura 14 - Previsão de desmatamento para o ano de 2020 com dados do INLA.



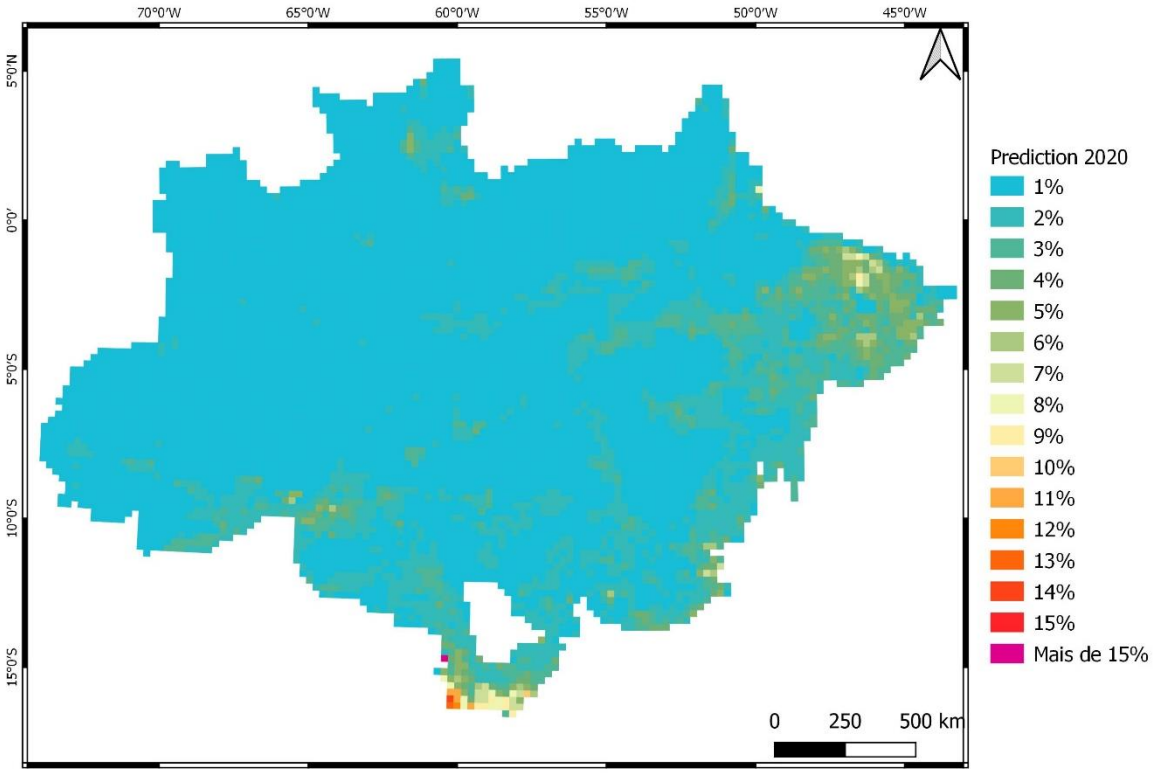
Fonte: elaboração própria (2021).

Os modelos desenvolvidos com a técnica de Spatial Random Forest, seja com o Kernel adaptativo ou fixo, obtiveram valores praticamente iguais de RMSE. Para uma análise mais profunda se estes valores são equivalentes seria necessário repetir

a modelagem por *n* vezes, a fim de se obter uma média e desvio padrão. Com estes valores seria possível fazer um teste estatístico de equivalência das médias nos dois modelos. Mesmo com os menores valores de RMSE no treinamento, o conjunto de validação apresentou um RMSE maior do que os modelos em Random Forest.

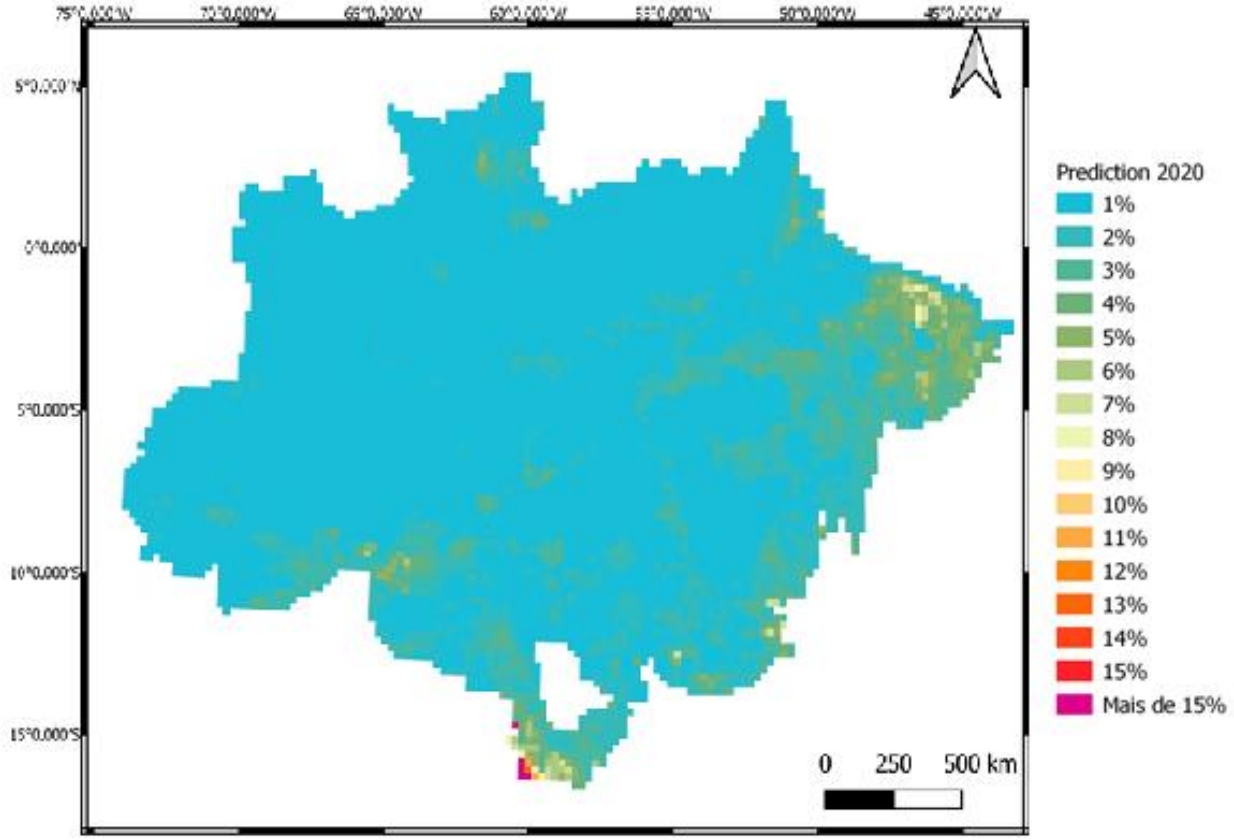
O uso de uma ou outra técnica dependerá de ajustes finos e do procedimento de seleção de variáveis, uma vez que os valores de desmatamento preditos com estes modelos foram muito próximos. O Random Forest fez a previsão de 31.815 km^2 e o Spatial Random Forest de 31.716 km^2 de desmatamento (Figuras 17 e 18).

Figura 15 - Previsão de desmatamento para o ano de 2020 com dados do Random Forest.



Fonte: elaboração própria (2021).

Figura 16 - Previsão de desmatamento para o ano de 2020 com dados do Spatial Random Forest.



Fonte: elaboração própria (2021).

5 CONCLUSÕES E TRABALHOS FUTUROS

Uma das principais contribuições deste trabalho foi a construção de uma base de dados atualizada, trazendo melhorias em relação a base de dados utilizada por Jaffe et al. (2021). Para trabalhos futuros, a inclusão de uma janela temporal de 5 anos nos dados do Mapbiomas, afim de confirmar uma transição como desmatamento, ou o uso de outras fontes de dados de desmatamento, a exemplo dos dados do PRODES (Programa oficial de desmatamento do governo), pode melhorar as estimativas de desmatamento. O uso de uma grade com células menores pode alterar a importância dos drivers de desmatamento indicados melhorar o refinamento da predição.

A porcentagem de pastagem em uma célula e a densidade de perímetro florestal, que é um indicativo da fragmentação, apareceram como os atributos mais importantes para modelagem do desmatamento. O índice de aridez também apareceu como um importante fator em ambos os modelos, enquanto outras variáveis se

destacaram dependendo do modelo. O uso de informações atualizadas de estradas, que incluam as estradas não pavimentadas e vicinais, pode melhorar a modelagem da influência desta variável no desmatamento.

Com relação à modelagem, pode-se verificar que, independentemente dos dados utilizados, as técnicas de Random Forest e Spatial Random Forest apresentam erros menores nas predições realizadas do que os modelos do tipo INLA. O processo de seleção de atributos mostrado pode ser essencial para melhorar ainda mais as predições do modelo, sendo que diversos atributos podem ser descartados, contribuindo para uma melhora na taxa de acerto das predições e um menor erro médio quadrático. Os resultados de predição de desmatamento de 2020 devem ser comparados com o desmatamento real para verificação dos resultados e os dados reais de desmatamento podem ser utilizados para gerar novas predições futuras. Todos os modelos utilizados indicaram os extremos leste e sul da Amazônia como as áreas com maior probalidade de desmatamento.

Diversas aplicações que visam a predição de dados futuros, como o desmatamento (Aguiar et al., 2016), crescimento urbano (Ahmed et al., 2014) e intensificação de agricultura (Britz et al., 2011) utilizam abordagens do tipo CLUE (Conversion of Land Use and its Effects) e suas ramificações (Verburg et al., 2002; Verburg et al., 2010). A abordagem CLUE é recomendada para trabalhos futuros que busquem melhorar a predição do desmatamento por permitindo gerar dados de treinamento com deslocamento temporal com base em técnicas de regressão espacial.

REFERÊNCIAS

AHMED, S. J.; BRAMLEY, G.; VERBURG, P. H. Key Driving factors influencing urban growth: Spatial-statistical modelling with Clue-s. **In**: DEWAN, A.; CORNER, R. (eds.). **Dhaka megacity:** geospatial perspectives on urbanization, environment and health. Springer Netherlands: 2014. p. 123-145. DOI 10.1007/978-94-007-6735-5_7.

ARAGÃO, L. E. O. C. The rainforest's water pump. **Nature**, v. 489, p. 217-218, 2012.

BENESTY, J.; CHEN, J.; HUANG, Y.; COHEN, I. Pearson correlation coefficient. **In:** _____. **Noise reduction in speech processing**. Berlin, Heidelberg: Springer, 2009. p. 1-4.

BLANGIARDO, M.; CAMELETTI, M. **Spatial and spatio-temporal Bayesian models with R-INLA.** John Wiley & Sons, 2015.

BRASIL. Ministério do Meio Ambiente (MMA). **O nível de referência de emissões florestais do Brasil para pagamentos por resultados de redução de emissões provenientes do desmatamento no bioma Amazônia**. Brasília, DF: MMA, 2017.

BRIENEN, R. J. W. *et al*. Long-term decline of the Amazon carbon sink. **Nature,** v. 519, p. 344–348, 2015.

BRITZ, W.; VERBURG, P. H.; LEIP, A. Modelling of land cover and agricultural change in Europe: Combining the CLUE and CAPRI-Spat approaches. Agriculture, ecosystems & environment. 2011 Jul 1;142(1-2):40-50.

GAO J, LI S. Detecting spatially non-stationary and scale-dependent relationships between urban landscape fragmentation and related factors using Geographically Weighted Regression. **Appl Geogr**., v. 31, p. 292–302, 2011.

GATTI, L, V.; *et al.* Amazonia as a carbon source linked to deforestation and climate change. **Nature**, v. 595, n. 7867, p. 388-393, jul., 2021.

GOLLINI, I.; LU, B.; CHARLTON, M.; BRUNSDON, C.; HARRIS, P. **GWmodel:** an R Package for Exploring Spatial Heterogeneity using Geographically Weighted Models. 2013.

GÓMEZ-RUBIO, V. **Bayesian inference with INLA.** CRC Press: 2020.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **Biomas e Sistema Costeiro-Marinho do Brasil:** compatível com a escala 1:250.000. Rio de Janeiro: IBGE, 2019. Disponível em: https://biblioteca.ibge.gov.br/visualizacao/livros/liv101676.pdf. Acesso em: fevereiro de 2021.

IMAZON. **Sad Alerta.** Boletim do desmatamento. [20--?]. Disponível em: https://imazon.org.br/categorias/sad-alerta/. Acesso em: fevereiro de 2021.

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS. **Terra Brasilis.** 2021. Disponível em: http://terrabrasilis.dpi.inpe.br/en/home-page/. Acesso em: fevereiro de 2021.

JAFFÉ, R.; *et al.* Forecasting deforestation in the Brazilian Amazon to prioritize conservation efforts. **Environmental Research Letters**, v. 16, jul. 2021.

KALOGIROU S, GEORGANOS S. 2018. "SpatialML." R Foundation for Statistical Computing.

KRAINSKI, E.; *et al.* **Advanced spatial modeling with stochastic partial differential equations using R and INLA.** Chapman and Hall/CRC, 2018.

MALHI, Y.; *et al*. The regional variation of aboveground live biomass in old-growth Amazonian forests. **Glob. Change Biol**., v. 12, p. 1107-1138, 2006.

MAPBIOMAS. **Alerta.** [20--?]. Disponível em: http://alerta.mapbiomas.org/en. Acesso em: fevereiro de 2021.

MARTINO. S.; RIEBLER, A. Integrated nested Laplace approximations (INLA). **In: Wiley StatsRef:** Statistics Reference Online. John Wiley & Sonsn: 2014.

MAURANO, L. E. P.; ESCADA, M. I. S.; RENNO, C. D. Padrões espaciais de desmatamento e a estimativa da exatidão dos mapas do PRODES para Amazônia Legal Brasileira. **Ciência Florestal**, v. 29, p.1763-1775, 2019.

OSHAN, T. M.; *et al.* MGWR: A Python Implementation of Multiscale Geographically Weighted Regression for Investigating Process Spatial Heterogeneity and Scale. **ISPRS Int. J. Geo-Inf.**, v. 8, n. 6, p. 269, 2019.

BRASIL. Ministério do Meio Ambiente (MMA). Decreto 9.578, de 22 de novembro de 2018. **Consolida atos normativos editados pelo Poder Executivo ...** 2020a. Disponível em: http://www.planalto.gov.br/ccivil_03/_Ato2015-2018/2018/Decreto/D9578.htm. Acesso em: 15 jan. 2021.

BRASIL. Ministério do Meio Ambiente (MMA). **Plano nacional para controle do desmatamento ilegal e recuperação da vegetação nativa**. 2020b. Disponível em: https://www.gov.br/planalto/pt-br/conheca-a-vice-presidencia/nota-a-imprensa/anexo-ao-resumo-informativo-no-3_de-29-5-2020.pdf. Acesso em: 15 jan. 2021.

MORAGA, P. **Geospatial health data:** Modeling and visualization with R-INLA and shiny. Chapman and Hall/CRC, 2019.

OBORN, I.; *et al.* **Five scenarios for 2050:** conditions for agriculture and land use. [S.l.]: Swedish University of Agricultural Sciences, 2011.

OMETTO, J. P.; AGUIAR, A. P.; MARTINELLI, L. A. Amazon deforestation in Brazil: effects, drivers and challenges. **Carbon Management**., v. 2, n. 5, p. 575-85, out. 2011.

PHILLIPS, O. L.; BRIENEN, R. J. W. Carbon uptake by mature Amazon forests has mitigated Amazon nations' carbon emissions. **Carbon Balance Manag**., v. 12, n. 1, 2017.

RUE, H.; HELD, L. **Gaussian Markov random fields:** theory and applications. CRC press, 2005.

RUE, H.; MARTINO, S.; CHOPIN, N. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. **Journal of the Royal Statistical Society: Series B (Statistical Methodology)**, v. 71, n. 2, p. 319-392, 2009.

SANTOS, F.; GRAW, V.; BONILLA, S. A geographically weighted random forest approach for evaluate forest change drivers in the Northern Ecuadorian Amazon. **PLoS ONE,** v. 14, n. 12, p. e0226224, 2019. DOI 10.1371/journal.pone.0226224.

SILVA JUNIOR, C. H. L.; PESSÔA, A. C. M.; CARVALHO, N. S.; REIS, J. B. C.; ANDERSON, L. O.; ARAGÃO, L. E. C. O. The Brazilian Amazon deforestation rate in 2020 is the greatest of the decade. **Nature Ecology & Evolution**, v. 5, n. 2, p. 144-145, 2021.

SPRACKLEN, D. V.; ARNOLD, S. R.; TAYLOR, C. M. Observations of increased tropical rainfall preceded by air passage over forests. **Nature**, v. 489, p. 282–285, 2012.

STAAL, A.; *et al*. Forest-rainfall cascades buffer against drought across the Amazon. **Nat. Clim. Chang**., v. 8, p. 539-543, 2018.

VERBURG, P. H.; OVERMARS, K. P. Combining top-down and bottom-up dynamics in land use modeling: exploring the future of abandoned farmlands in Europe with the Dyna-CLUE model. **Landscape ecology,** v. 24, n.9, p. 1167-81, 2009.

VERBURG, P. H.; SOEPBOER, W.; VELDKAMP, A.; LIMPIADA, R.; ESPALDON, V.; MASTURA, S. S. Modeling the spatial dynamics of regional land use: the CLUE-S model. **Environmental management**, v. 30, n. 3, p. 391-405, 2002.

WORLD WIDE FUND FOR NATURE. **Living Amazon Report 2016:** risk and resilience in a new area. Gland, Switzerland: WWF, 2016.

APÊNDICES

Apêndice A – Metadados e cálculo de atributos da base de dados

TERRAS INDÍGENAS

Camada: TI_5880.shp

EPSG: 5880

Pré-processamento realizado: Reprojetar de EPSG:4674 para EPSG:5880, Clip pelo limite do bioma Amazônia e remoção de atributos (ficou somente o geom e as Terras Indígenas Homologadas)

Processamento para cálculo de distância e área: Rodar a ferramenta de análise de sobreposição no QGis, o resultado sai em m^2 e % da área do grid. A distância do centroide do grid para a TI mais próxima foi calculada pelo plugin NNjoin do Qgis. Transformar para Km^2 via R ou fieldcalc no Qgis.

Descrição rápida: Terras indígenas FUNAI

Fonte para download (dado bruto):

http://www.funai.gov.br/index.php/shape

(Reserva Indígena)

Descrição estendida:

Dados produzidos pela FUNAI (Polígonos e Pontos das terras indígenas brasileiras), e utilizados na elaboração do shape de terras indígenas com a melhor base oficial disponível.

Referência principal (mais próximo):

Título: O uso do gvSIG na construção do Sistema de Informação Geográfica da Fundação Nacional do Índio – Funai.

Editor: FUNAI, Patrícia Cayres Ramos

Ano: 2011

Descrição física: 8 p.

http://downloads.gvsig.org/download/events/jornadas-lac/3as-jornadas-lac/articles/Article-gvsig_SIG_Fundacao_Nacional_Indio_Funai%20.pdf

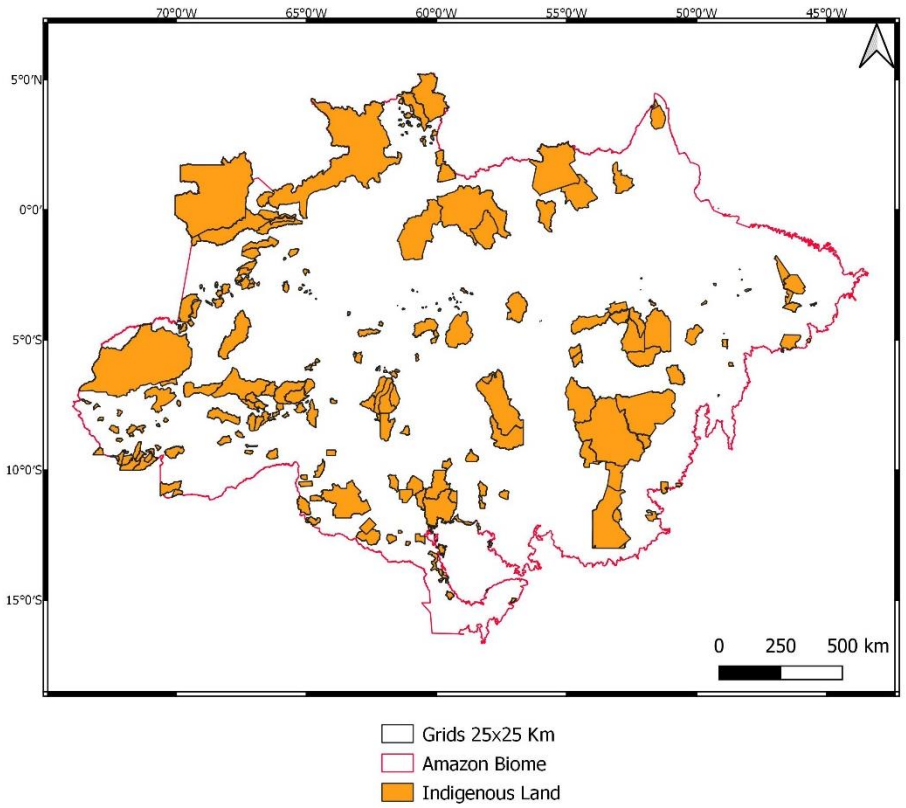


Figura 17: Dados das Terras Indígenas (FUNAI)

UNIDADES DE CONSERVAÇÃO

Camada: UCs_todas_5880.shp

EPSG: 5880

Pré-processamento realizado: Reprojetar de EPSG:4674 para EPSG:5880, Clip pelo limite do bioma Amazônia, remoção de atributos (ficou somente o geom e a data de criação), remoção das sobreposições – Rever futuramente se compensa deixar o tipo da UC.

Processamento para cálculo de distância e área: Rodar a ferramenta de análise de sobreposição no QGis, o resultado sai em m^2 e % da área do grid. A distância do centroide do grid para a TI mais próxima foi calculada pelo plugin NNjoin do Qgis. Transformar para Km^2 via R ou fieldcalc no Qgis.

Descrição rápida: Áreas protegidas MMA

Fonte para download (dado bruto):

http://mapas.mma.gov.br/i3geo/datadownload.htm

(i3Geo > Areas_Especiais > Unidades_de_Conservacao > Unidades_de_Conservacao(todas))

Descrição estendida:

Unidades de Conservação (UC) do Brasil, que finalizaram o processo de cadastramento no CNUC (Cadastro Nacional de Unidades de Conservação), estando assim de acordo com a legislação do SNUC (lei nº 9.985/2000).

Referência principal:

http://mapas.mma.gov.br/geonetwork/srv/br/metadata.show?id=1250

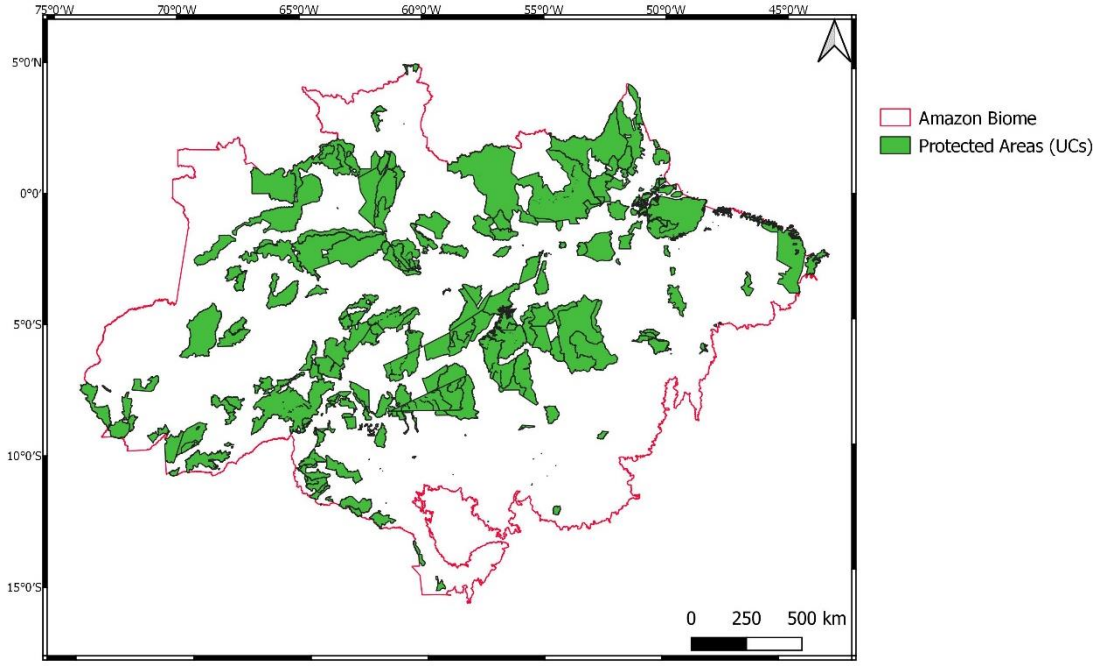


Figura 18: Unidades de Conservação obtidas do MMA.

HIDROVIAS

Camada: Hidrovias_5880.shp

EPSG: 5880

Pré-processamento realizado: Reprojetar de EPSG:4674 para EPSG:5880 e cortar para o limite do bioma. Os atributos foram apagados e foi deixado somente a coluna geom.

Processamento para o cálculo de soma de perímetro e distância do centroide: Usar soma de comprimentos de linha para somas as linhas em cada grid. Usar o

plugin NNJoin do Qgis para calcular a distância do centroide do grid para a linha de hidrovia. Os dados saem em m^2, transformer em Km^2 via R ou fieldcalc no Qgis.

Descrição rápida: Hidrovias pré-processadas pelo MapBiomas.

Fonte para download (dado bruto):

https://mapbiomas.org/dados-de-infraestrutura?cama_set_language=pt-BR

(Transportes > Hidrovia > Download)

Descrição estendida:

Via de navegação dentro de um rio, lagoa ou canal artificial com dimensões e parâmetros padronizados, segundo critérios de engenharia. Hidrovias e vias navegáveis interiores (trechos de navegação inexpressiva foram filtrados).

Referência principal:

Título: Infrastructure Layers (Transportation, Energy and Mining) – Appendix (Collection 5 Version 1).

Editor: MapBiomas

Ano: 2019

Descrição física: 10 p.

https://mapbiomas-br-site.s3.amazonaws.com/20200827_MapBiomas_InfrastructureLayers_ATBD_col5_v1_ENG__1_.pdf

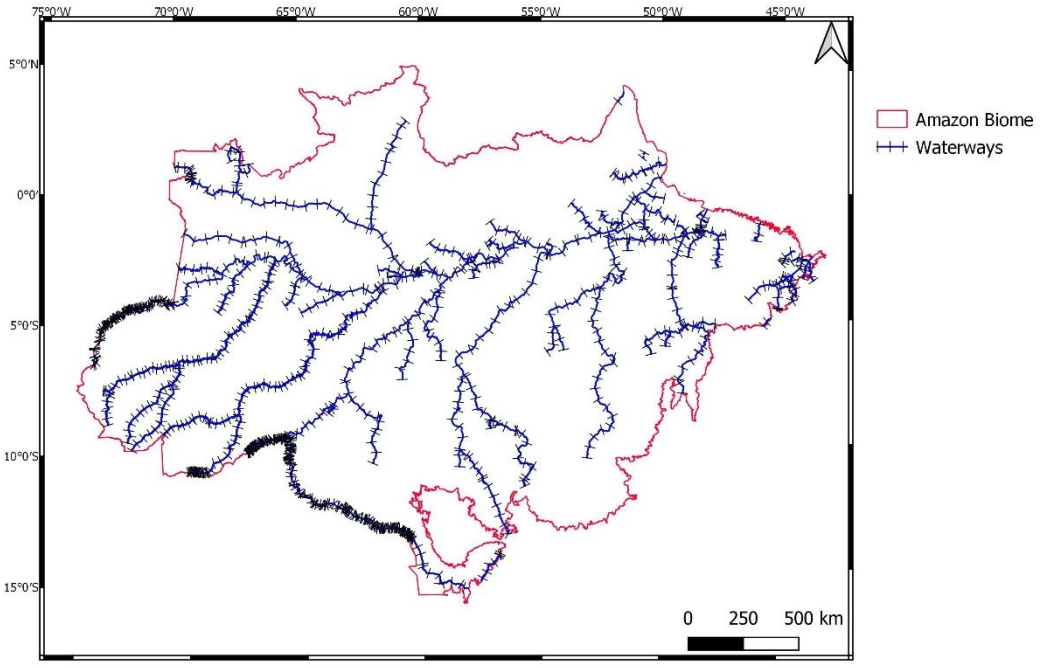


Figura 19: Shapefile de Hidrovias processado pela iniciativa do MapBiomas

RIOS

Camada: hid_trecho_drenagem_5880_make_valid.shp

EPSG: 5880

Pré-processamento realizado: Reprojetar de EPSG:4674 para EPSG:5880 e cortar para o limite do bioma. Os atributos foram apagados e foi deixado somente a coluna geom. Rodar Validador de geometria no terraview para eliminar erros.

Processamento para o cálculo de soma de perímetro e distância do centroide: Usar soma de comprimentos de linha para somas as linhas em cada grid. Usar o plugin NNJoin do Qgis para calcular a distância do centroide do grid para a linha de rios. Os dados saem em m^2, transformar em Km^2 via R ou fieldcalc no Qgis.

Descrição rápida: Hidrografia IBGE

Fonte para download (dado bruto):

https://www.ibge.gov.br/geociencias/cartas-e-mapas/bases-cartograficas-continuas/15759-brasil.html?=&t=downloads

(bc250 > Shapefile > bc250_shapefile_06_11_2019.zip)

Descrição estendida:

A BC250 é um conjunto de dados geoespaciais de referência, estruturados em bases de dados digitais, permitindo uma visão integrada do território nacional nesta escala, sendo de grande importância para projetos de planejamento regional, de cunho ambiental e de gestão do território. Os elementos cartográficos representados nessa base de dados possuem correspondência com informações da realidade física do território, simplificadas para a escala de mapeamento a que este produto foi especificado, neste caso 1:250.000. Elementos de dimensões inferiores às previstas nas especificações técnicas não estão contemplados, bem como denominações e classificações que promovam um grau de detalhamento incompatível com a escala.

Referência principal:

Título: BASE CARTOGRÁFICA CONTINUA DO BRASIL, ESCALA 1:250.000 - BC250

Editor: IBGE, DIRETORIA DE GEOCIÊNCIAS COORDENAÇÃO DE CARTOGRAFIA

Ano: 2019

Descrição física: 28 p.

https://geoftp.ibge.gov.br/cartas_e_mapas/bases_cartograficas_continuas/bc250/versao2019/informacoes_tecnicas/Documentacao_bc250_v2019.pdf

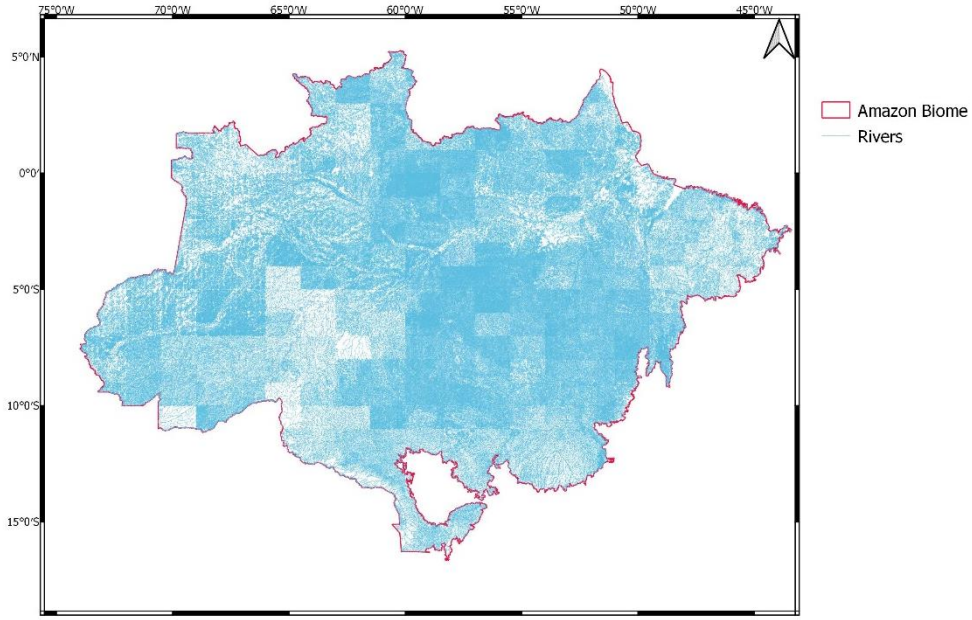


Figura 20: Dados de Corpos d'agua do IBGE.

ESTRADAS FEDERAIS

Camada: Rodovias_Federais_5880.shp

EPSG: 5880

Pré-processamento realizado: Reprojetar de EPSG:4674 para EPSG:5880 e cortar para o limite do bioma. Os atributos foram apagados e foi deixado somente a coluna geom.

Processamento para o cálculo de soma de perímetro e distância do centroide: Usar soma de comprimentos de linha para somas as linhas em cada grid. Usar o plugin NNJoin do Qgis para calcular a distância do centroide do grid para a linha de rodovia. Os dados saem em m^2, transformer em Km^2 via R ou fieldcalc no Qgis.

Descrição rápida: Rodovias Federais pré-processadas pelo MapBiomas.

Fonte para download (dado bruto):

https://mapbiomas.org/dados-de-infraestrutura?cama_set_language=pt-BR

(Transportes > Rodoviário > Download)

Descrição estendida:

Estrada de rodagem sob responsabilidade federal do DNIT - Departamento Nacional de Infraestrutura de Transportes. Rodovias em operação ou em obra (rodovias planejadas foram filtradas).

Referência principal:

Título: Infrastructure Layers (Transportation, Energy and Mining) – Appendix (Collection 5 Version 1).

Editor: MapBiomas

Ano: 2019

Descrição física: 10 p.

https://mapbiomas-br-site.s3.amazonaws.com/20200827_MapBiomas_InfrastructureLayers_ATBD_col5_v1_ENG__1_.pdf

ESTRADAS ESTADUAIS

Camada: Rodovia_estadual_5880.shp

EPSG: 5880

Pré-processamento realizado: Reprojetar de EPSG:4674 para EPSG:5880 e cortar para o limite do bioma. Os atributos foram apagados e foi deixado somente a coluna geom.

Processamento para o cálculo de soma de perímetro e distância do centroide: Usar soma de comprimentos de linha para somas as linhas em cada grid. Usar o plugin NNJoin do Qgis para calcular a distância do centroide do grid para a linha de rodovia. Os dados saem em m^2, transformar em Km^2 via R ou fieldcalc no Qgis.

Descrição rápida: Rodovias Estaduais pré-processadas pelo MapBiomas.

Fonte para download (dado bruto):

https://mapbiomas.org/dados-de-infraestrutura?cama_set_language=pt-BR

(Transportes > Rodoviário > Download)

Descrição estendida:

Estrada de rodagem sob responsabilidade federal do DNIT - Departamento Nacional de Infraestrutura de Transportes. Rodovias em operação ou em obra (rodovias planejadas foram filtradas).

Referência principal:

Título: Infrastructure Layers (Transportation, Energy and Mining) – Appendix (Collection 5 Version 1).

Editor: MapBiomas

Ano: 2019

Descrição física: 10 p.

https://mapbiomas-br-site.s3.amazonaws.com/20200827_MapBiomas_InfrastructureLayers_ATBD_col5_v1_ENG__1_.pdf

ESTRADAS SEM PAVIMENTO

Camada: estradas_imazon_5880.shp

EPSG: 5880

Pré-processamento realizado: Reprojetar de EPSG: 4618 para EPSG:5880 e cortar para o limite do bioma. Os atributos foram apagados e foi deixado somente a coluna geom. Verificar por NAs e erros de topologia no TA.

Processamento para o cálculo de soma de perímetro e distância do centroide: Usar soma de comprimentos de linha para somas as linhas em cada grid. Usar o plugin NNJoin do Qgis para calcular a distância do centroide do grid para a linha de rodovia. Os dados saem em m^2, transformer em Km^2 via R ou fieldcalc no Qgis.

Descrição rápida: Rodovias Estaduais pré-processadas pelo MapBiomas.

Fonte para download (dado bruto):

https://www.imazongeo.org.br/files/uploads/bases/sad/estradas_bioma_2012_geo.zip

(Necessário acessar o webgis do Imazon para download)

Descrição estendida:

Os dados de rodovias mapeados pelo Imazon até o ano de 2012. O mapeamento de estradas teve início na região centro-oeste do estado do Pará, para o ano de 1985 até 2001 e, posteriormente, se estendeu para o limite do bioma. Foi inicialmente

realizado com imagens Landsat e a ultima data de atualização dos dados é o ano de 2012.

Referência principal:

BRANDÃO JÚNIOR, A. O.; SOUZA JÚNIOR, C. M. Mapping unofficial roads with Landsat images: a new tool to improve the monitoring of the Brazilian Amazon rainforest. International Journal of Remote Sensing, v. 27, n. 1, p. 177–189, 2006.

POPULAÇÃO, PIB E IDH

Camada: Municipios_IDH_POP_PIB_5880.shp

EPSG: 5880

Pré-processamento realizado: Atualizar cada tabela baixada de forma a deixar o dado de interesse (PIB, IDH e População) e o código go município. Utilizar o R para agregar todos os atributos no shape de municípios. Utilizar o shape de centróides para replicar os atributos para cada célula e agregar com o shape de células depois.

Descrição rápida: Dados socioeconômicos como PIB, IDH e população.

Fonte para download (dado bruto):

IDH → http://www.atlasbrasil.org.br/ranking

População → https://www.ibge.gov.br/estatisticas/downloads-estatisticas.html

PIB → https://www.ibge.gov.br/estatisticas/economicas/contas-nacionais/9088-produto-interno-bruto-dos-municipios.html?=&t=resultados

Descrição estendida:

Os dados de IDH médio são provenientes do Atlas Brasil para o Ano de 2010. Já os dados de população e PIB são derivados e tabela do IBGE ano a ano. O PIB municipal deve ser multiplicado por 1000 para o valor total.

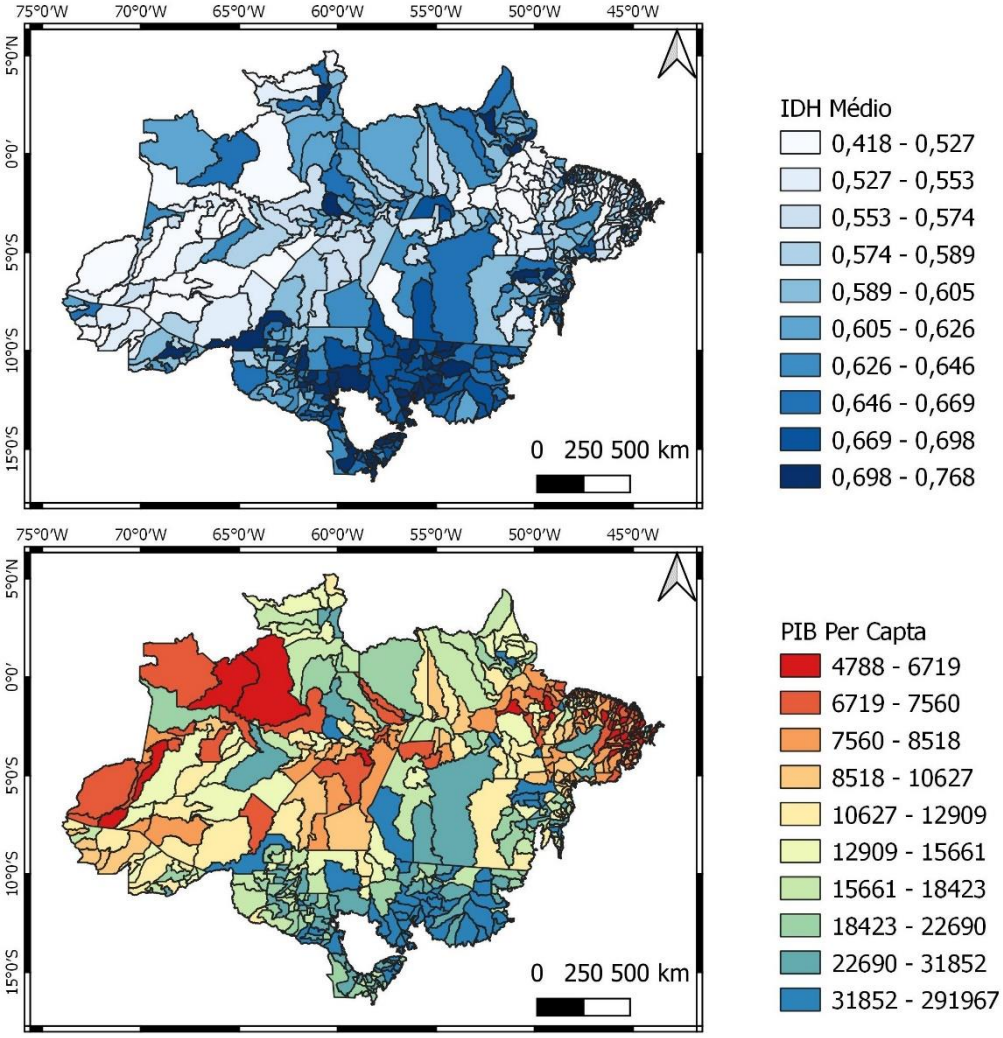


Figura 21: IDH E PIB calculados por município dentro do bioma Amazonian.

ÍNDICE DE ARIDEZ

Camada: Aridity_XXXX.tif

XXXX = [2015, 2016, 2017, 2018, 2019]

EPSG: 5880

Pré-processamento realizado: Extrair as imagens de aridez ano a ano no GEE.

Processamento: Rodar estatísticas zonais para o grid (Mediana). Arrumar as cels com Nodata (rios, lagos e oceanos) com os seguintes vizinhos mais próximos (características similares).

Faltante --> Recebe

C114L71 --> C115L71

C113L71 --> C115L71

C100L93 --> C99L93

C109L74 --> C108L74

C106L77 --> C105L77

C106L76 --> C105L76

Descrição rápida: Índice de aridez da UNESCO

Fonte para download (dado bruto): Gerado no GEE

Descrição estendida:

Os dados de aridez dependem de variáveis como a precipitação (P) e a evapotranspiração potencial para uma determinada janela temporal. O índice de aridez (IA) pode ser calculado pelo quociente P/ETP e classificado de acordo com os valores:

IA < 0,03 → Zona Super árida

0,03 < IA < 0,20 → Zona árida

0,20 < IA < 0,50 → Zona semi-árid

0,20 < IA < 0,50 → Zona semi-árida

0,50 < IA < 0,75 → Zona sub húmida

0,75 < IA → Zona húmida

Referência principal:

United Nations Educational, Scientific and Cultural Organization (UNESCO). Map of the World Distribution of
Arid Regions: Explanatory Note; MAB Technical Notes; UNESCO: Paris, France, 1979.

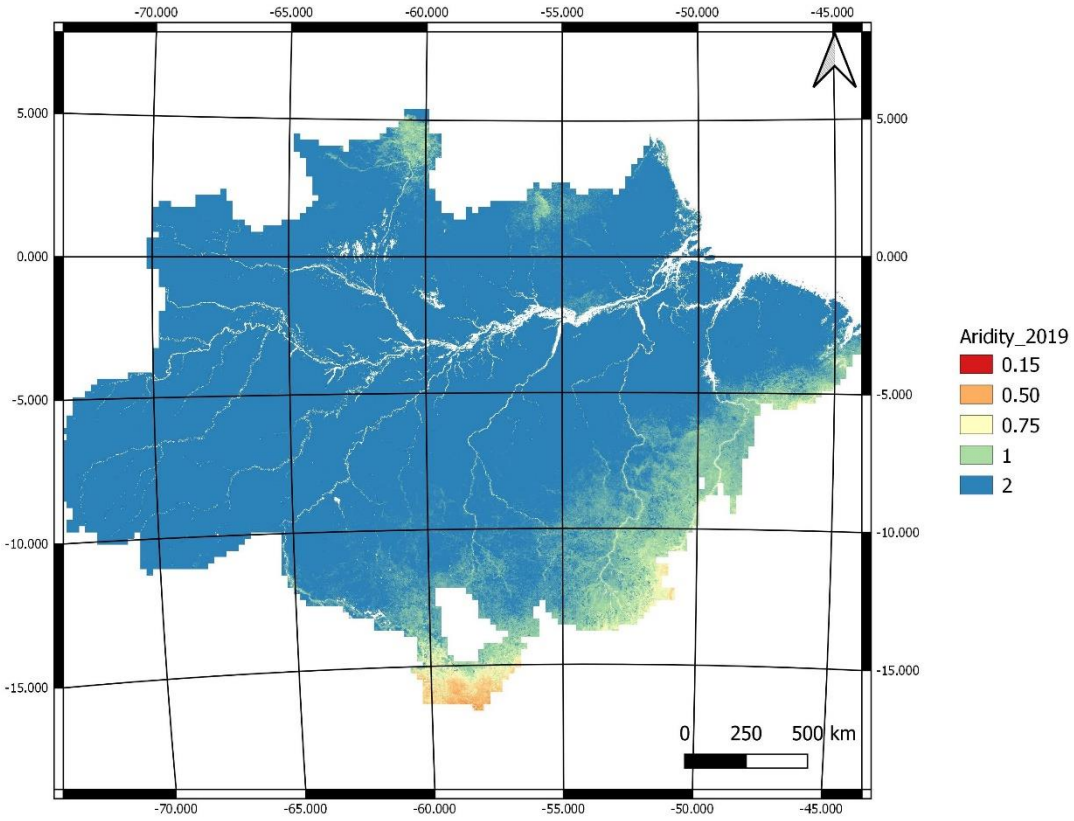


Figura 22: Índice de aridez gerado pelo Google Earth Engine de acordo com a metodologia da FAO.

SRTM

Camada: altitude.tif e declividade.tif

EPSG: 5880

Pré-processamento realizado: Extrair as imagens do SRTM no GEE.

Processamento: Rodar estatísticas zonais para o grid (Mediana). Mosaicar se precisar (o ArcGis mosaica bem mais rápido que o Qgis)

Descrição rápida: Dados de Altitude e Declividade do SRTM

Fonte para download (dado bruto):

Gerado no GEE (CGIAR/SRTM90_V4)

Descrição estendida:

The Shuttle Radar Topography Mission (SRTM) digital elevation dataset was originally produced to provide consistent, high-quality elevation data at near global

scope. This version of the SRTM digital elevation data has been processed to fill data voids, and to facilitate its ease of use.

Referência principal:

Jarvis, A., H.I. Reuter, A. Nelson, E. Guevara. 2008. Hole-filled SRTM for the globe Version 4, available from the CGIAR-CSI SRTM 90m Database: http://srtm.csi.cgiar.org.

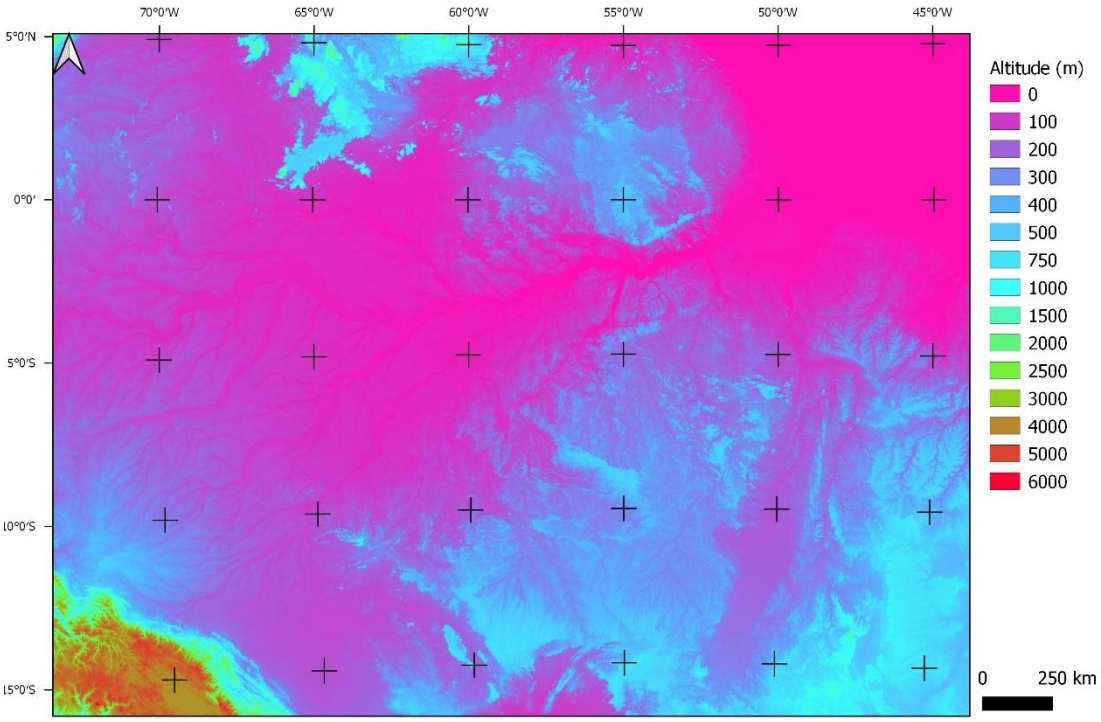


Figura 23: Dado de Altitude gerado no Google Earth Engine utilizando o SRTM com 90m de resolução espacial.

USO E COBERTURA MAPBIOMAS

Camada: Mapbiomas_XXXX_reclass.tif

XXXX = [2016,2017,2018,2019]

EPSG: 5880

Pré-processamento realizado: Reprojetar de EPSG:4326 para EPSG:5880 e reclassificar o raster de acordo com a tabela abaixo.

Processamento para o cálculo proporção de cada classe: Usar o histograma zonal do qgis, calcular o total de pixels excluindo nodata, recalcular as % de cada classe.

Valor	Valor

0	Nodata	0	Nodata
3	1.1.1. Formação Florestal	1	Natural
4	1.1.2. Formação Savânica	1	Natural
5	1.1.3. Mangue	1	Natural
9	1.2. Floresta Plantada	3	Agricultura
11	2.1. Campo Alagado e Área Pantanosa	1	Natural
12	2.2. Formação Campestre	1	Natural
15	3.1. Pastagem	4	Pastagem
20	3.2.1.2. Cana	3	Agricultura
23	4.1. Praia e Duna	1	Natural
24	4.2. Infraestrutura Urbana	5	Urbano
25	4.4. Outras Áreas não Vegetadas	5	Urbano
30	4.3. Mineração	6	Mineração
32	2.3. Apicum	1	Natural
33	5.1. Rio, Lago e Oceano	2	Agua
39	3.2.1.1. Soja	3	Agricultura
41	3.2.1.3. Outras Lavouras Temporárias	3	Agricultura

Descrição rápida: Porcentagem de cobertura com dados do Mapbiomas

Fonte para download (dado bruto):

https://mapbiomas.org/colecoes-mapbiomas-1?cama_set_language=pt-BR

Descrição estendida:

Dados de uso e cobertura do solo do mapbiomas reclassificados em classes mais simples (Natural, Pastagem, Agricultura, Urbano, Mineração e Agua)

Referência principal:

Título: MapBiomas General “Handbook” Algorithm Theoretical Basis Document (ATBD) Collection 5 Version 1.0. Editor: MapBiomas. Ano: 2020. Descrição física: 48 p.

https://mapbiomas-br-site.s3.amazonaws.com/ATBD_Collection_5_v1.pdf

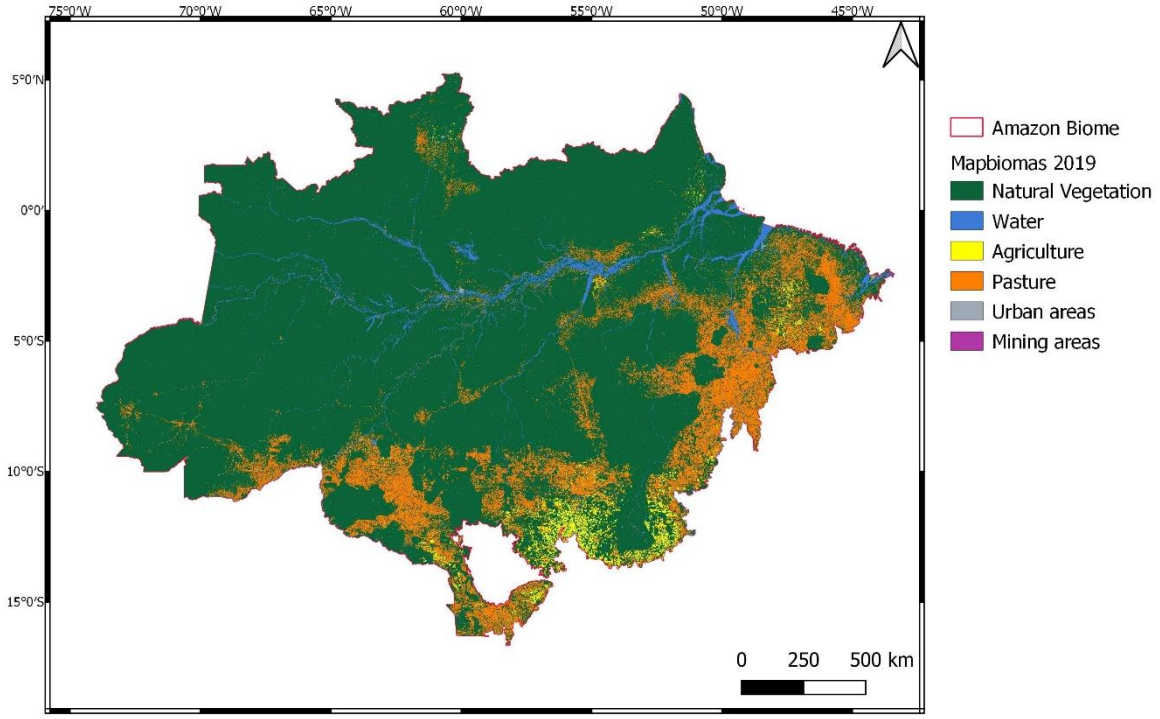


Figura 24: Uso e cobertura da iniciativa do Mapbiomas com classes agrupadas.

DISTÂNCIA URBANO

Camada: Grid_25_25_aux_Dist_Urb_19_18_17_16.shp

EPSG: 5880

Pré-processamento realizado: Usar o GEE para exportar os shape da classe do mapbiomas. Clipar pelo limite do bioma e reprojetar para 5880.

Processamento para o cálculo da distância do centroide: Usar o plugin NNJoin para calcular a distância do centroide do grid.

Descrição rápida: Distância do centroide da célula até o urbano mais próximo.

Fonte para download (dado bruto):

https://mapbiomas.org/colecoes-mapbiomas-1?cama_set_language=pt-BR

Descrição estendida:

Dados de uso e cobertura do solo do mapbiomas no formato em shape foram utilizados para o calculo da distância.

Referência principal:

Título: MapBiomas General “Handbook” Algorithm Theoretical Basis Document (ATBD) Collection 5 Version 1.0

Editor: MapBiomas

Ano: 2020

Descrição física: 48 p.

https://mapbiomas-br-site.s3.amazonaws.com/ATBD_Collection_5_v1.pdf

DISTÂNCIA MINERACAO

Camada: Grid_25_25_aux_Dist_Min_19_18_17_16.shp

EPSG: 5880

Pré-processamento realizado: Usar o GEE para exportar os shape da classe do mapbiomas. Clipar pelo limite do bioma e reprojetar para 5880.

Processamento para o cálculo da distância do centroide: Usar o plugin NNJoin para calcular a distância do centroide do grid.

Descrição rápida: Distância do centroide da célula até a mineração mais próxima.

Fonte para download (dado bruto):

https://mapbiomas.org/colecoes-mapbiomas-1?cama_set_language=pt-BR

Descrição estendida:

Dados de uso e cobertura do solo do mapbiomas no formato em shape foram utilizados para o cálculo da distância.

Referência principal:

Título: MapBiomas General “Handbook” Algorithm Theoretical Basis Document (ATBD) Collection 5 Version 1.0

Editor: MapBiomas

Ano: 2020

Descrição física: 48 p.

https://mapbiomas-br-site.s3.amazonaws.com/ATBD_Collection_5_v1.pdf

DISTÂNCIA AGRICULTURA

Camada: Grid_25_25_aux_dist_agri_19_18_17_16.shp

EPSG: 5880

Pré-processamento realizado: Usar o GEE para exportar os shape da classe do mapbiomas. Clipar pelo limite do bioma e reprojetar para 5880.

Processamento para o cálculo da distância do centroide: Usar o plugin NNJoin para calcular a distância do centroide do grid.

Descrição rápida: Distância do centroide da célula até a agricultura mais próxima.

Fonte para download (dado bruto):

https://mapbiomas.org/colecoes-mapbiomas-1?cama_set_language=pt-BR

Descrição estendida:

Dados de uso e cobertura do solo do mapbiomas no formato em shape foram utilizados para o cálculo da distância.

Referência principal:

Título: MapBiomas General “Handbook” Algorithm Theoretical Basis Document (ATBD) Collection 5 Version 1.0

Editor: MapBiomas

Ano: 2020

Descrição física: 48 p.

https://mapbiomas-br-site.s3.amazonaws.com/ATBD_Collection_5_v1.pdf

FOREST EDGE DENSITY

Camada: Grid_25_25_aux_for_edge_16_17_18_19.shp

EPSG: 5880

Pré-processamento realizado: Usar o GEE para exportar os shape da classe de florestas do mapbiomas (ou utilizar o arcgis para vetorizar o valor 3, foi mais rápido e utilizar o simplify polygon!!!!!!!!!!!!!!!!!!)

Como se trata de um raster existe várias “escadinhas” que não representam corretamente uma linha reta e acrescem demais o perímetro de floresta, o método de simplificar os pols transforma isso em uma linha:

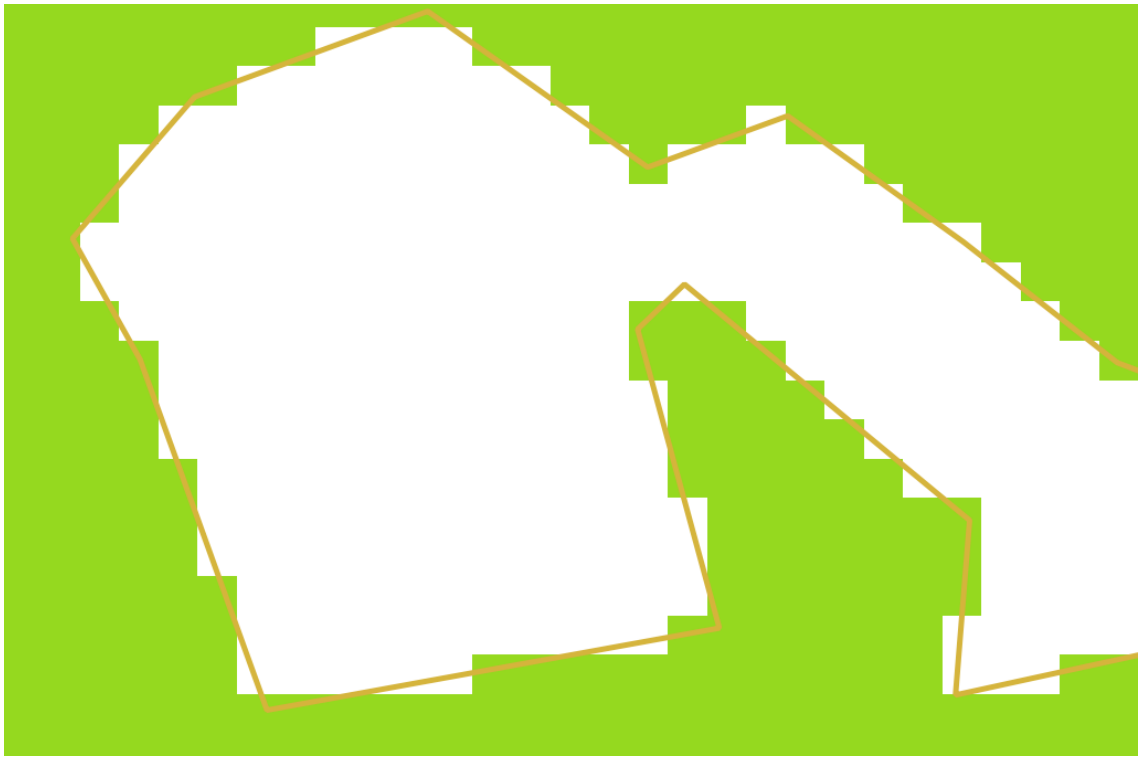


Figura 25: Exemplo de cálculo para a variável Forest Edge Density: Polígono simplificado em marrom e a borda do raster de Floresta em verde.

Processamento para o cálculo do forest edge: Transformar os polígonos para polilines, reprojetar pra 5880, calcular a soma das linhas por grid e converter para m (essa etapa demora dias no QGis), pra contornar o mais fácil a fazer é dar um intersect no shape dos grids e das linhas isso vai quebrar todas as linhas dentro de um grid calcular o shape lenght de cada linha e agregar a soma por id no R e gerar um shape de saída. calcular a área do grid e dividir pela soma das linhas (m/km2) não esquecer de multiplicar a área por 10^(-6)

Descrição rápida: Forest Edge Density

Fonte para download (dado bruto):

https://mapbiomas.org/colecoes-mapbiomas-1?cama_set_language=pt-BR

Descrição estendida:

Cálculo do perímetro de floresta por unidade de área, foi considerado somente a classe florestal do mapbiomas para o cálculo.

Referência principal:

Título: MapBiomas General “Handbook” Algorithm Theoretical Basis Document (ATBD) Collection 5 Version 1.0

Editor: MapBiomas

Ano: 2020

Descrição física: 48 p.

https://mapbiomas-br-site.s3.amazonaws.com/ATBD_Collection_5_v1.pdf

POLOS MADEIREIROS

Camada: Polos_madeireiros_5880.shp

EPSG: 5880

Pré-processamento realizado: Baixar o dado do Imazon e reprojetar para 5880

Processamento para o cálculo da distância do centroide: Usar o plugin NNJoin para calcular a distância do centroide do grid.

Descrição rápida: Distância do centroide da célula até o polo madeireiro mais próximo.

Fonte para download (dado bruto):

https://imazongeo.org.br/#/

(downloads > Polos Madeireiros)

Descrição estendida:

Dados em pontos com os principais polos de processamento de madeira para o bioma Amazônia.

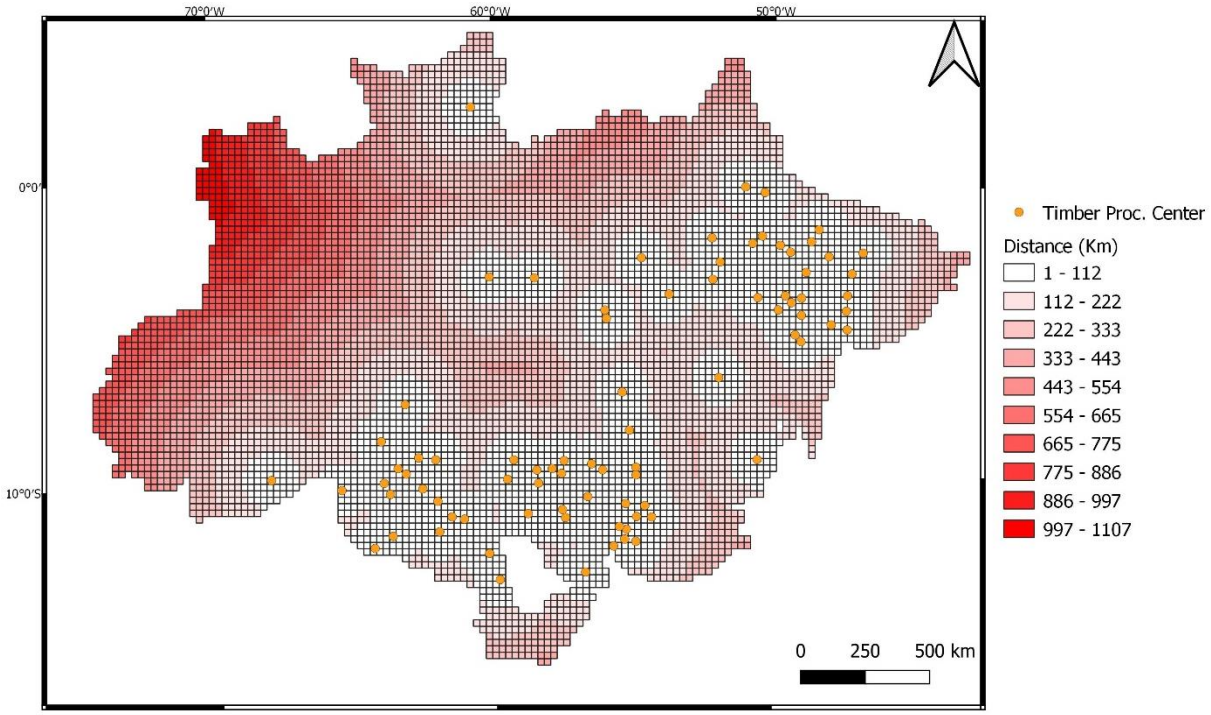


Figura 26: Distancia de polos madeireiros para as células de 25 x 25 Km

ASSENTAMENTOS RURAIS

Camada: Assentamento_Brasil_clean5880.shp

EPSG: 5880

Pré-processamento realizado: Baixar dados do incra, remover assentamentos em obtenção, clipar pelo bioma e reprojetar para 5880.

Processamento para cálculo de distância e área: Rodar a ferramenta de análise de sobreposição no QGis, o resultado sai em m^2 e % da área do grid. A distância do centroide do grid para o AST mais próximo foi calculada pelo plugin NNjoin do Qgis. Transformar para Km^2 via R ou fieldcalc no Qgis.

Descrição rápida: Assentamentos Rurais Incra

Fonte para download (dado bruto):

https://certificacao.incra.gov.br/csv_shp/export_shp.py

Descrição estendida:

Polígonos referentes aos assentamentos rurais do Instituto Nacional de Colonização e Reforma Agrária (INCRA)

https://acervofundiario.incra.gov.br/acervo/acv.php

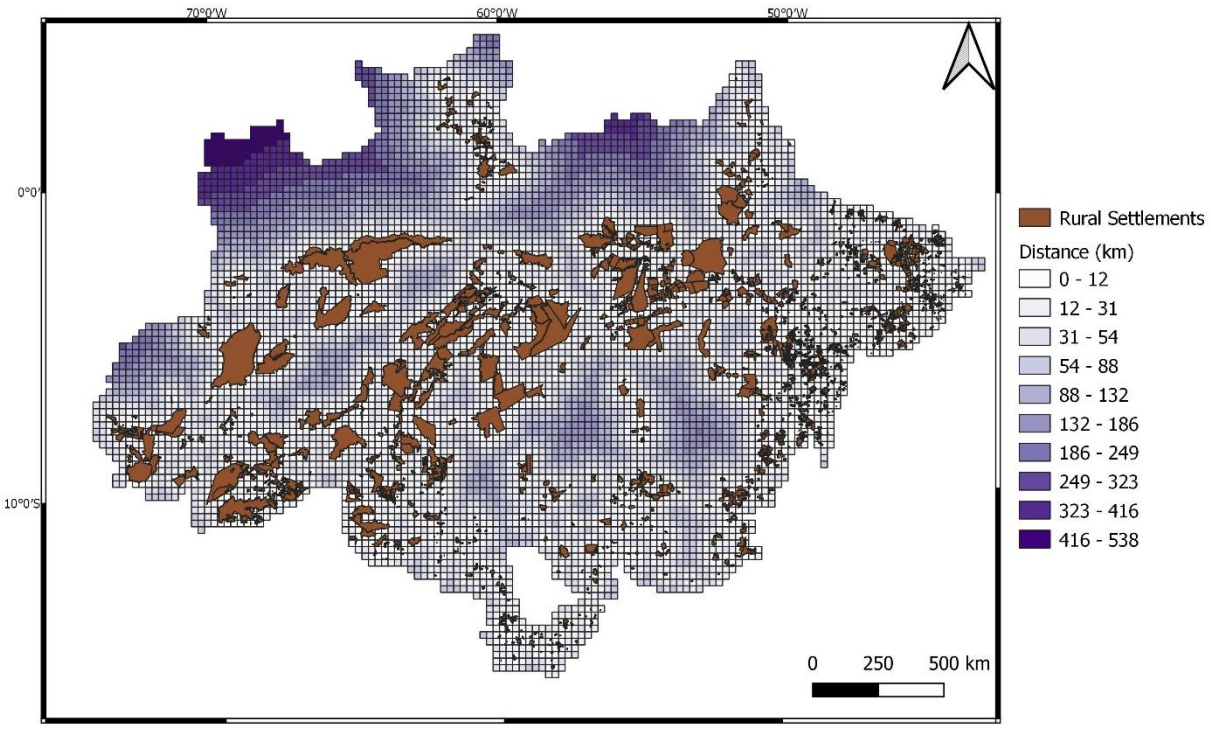


Figura 27: Distancia de Assentamentos rurais para as células de 25 x 25 Km

BIOMA AMAZÔNIA

Camada: Bioma_Amz_5880.shp

EPSG: 5880

Pré-processamento realizado: Reprojetar de EPSG:4674 para EPSG:5880

Descrição rápida: Limite do Bioma Amazônia em escala 1:250.000

Fonte para download (dado bruto):

https://www.ibge.gov.br/geociencias/informacoes-ambientais/15842-biomas.html?=&t=downloads

(vetores > Biomas_250mil.zip)

Descrição estendida:

O mapa Biomas e sistema costeiro-marinho do Brasil é compatível com a escala 1:250 000 e contempla aperfeiçoamentos na representação dos limites e incorpora atualizações e avanços conceituais e tecnológicos, com destaque para o aumento da resolução, tanto espacial quanto espectral e temporal, das imagens orbitais. Ele busca, assim, atender as expectativas de diferentes setores da sociedade

interessados em um mapeamento mais detalhado que o oferecido no Mapa de biomas do Brasil: primeira aproximação, divulgado, em 2004, na escala 1:5.000 000, em cooperação com o Ministério do Meio Ambiente.

Referência principal:

Título: Biomas e sistema costeiro-marinho do Brasil: compatível com a escala 1:250 000

Editor: IBGE, Coordenação de Recursos Naturais e Estudos Ambientais

Ano: 2019

Descrição física: 164 p.

https://biblioteca.ibge.gov.br/index.php/biblioteca-catalogo?view=detalhes&id=2101676

Apêndice B – Códigos utilizados na modelagem

Random Forest

```
# Carregar bibliotecas

library(sf)
library(raster)
library(dplyr)
library(spData)
library(corrplot)
library(rgdal)
library(tidyverse)
library(Hmisc)
library(tictoc)
library(rsample)
library(randomForest)
library(ranger)
library(caret)
library(ggplot2)
library(randomForestExplainer)
library(e1071)
library(FSelector)
library(ggplot2)
library(reshape2)

#limpar ambiente
cat("\f")
rm(list = ls())

#Setar a pasta
setwd("E:/Posdoc_ITV/Base_qgis/999_Master_Join/Modelagem_001/backup")
getwd()

# Carregar o shape de 2016_2017 (Já removidos: Pc_IL, Pc_PA, Pc_Ast, Lat, Long) - Treinamento
# Transforma em Df, remove a col geom (os nomes dos atributos estão iguais ja)
shape_16_17 <- st_read("Cond_2016_def_2017.shp")
str(shape_16_17)
shape_16_17_df <- data.frame(shape_16_17)
shape_16_17_df <- shape_16_17_df[-c(37)]
str(shape_16_17_df)

# Carregar o shape de 2017_2018 (Já removidos: Pc_IL, Pc_PA, Pc_Ast, Lat, Long) - Teste
shape_17_18 <- st_read("Cond_2017_def_2018.shp")
str(shape_17_18)
shape_17_18_df <- data.frame(shape_17_18)
shape_17_18_df <- shape_17_18_df[-c(37)]
str(shape_17_18_df)

# Carregar o shape de 2018_2019 (Já removidos: Pc_IL, Pc_PA, Pc_Ast, Lat, Long) - Validação
shape_18_19 <- st_read("Cond_2018_def_2019.shp")
str(shape_18_19)
shape_18_19_df <- data.frame(shape_18_19)
shape_18_19_df <- shape_18_19_df[-c(37)]
str(shape_18_19_df)


# Carregar o shape de 2019 (Previsão 2020)
predict_20 <- st_read("predict_2020_cond_2019.shp")
str(predict_20)
predict_20_df <- data.frame(predict_20)
predict_20_df <- predict_20_df[-c(36)]
str(predict_20_df)


# avaliação inicial de correlação de pearson (por esse que removi as colunas acima)
# não eh necessario rodar nesse script
#mydata.cor = cor(shape_16_17_df)
#corrplot(mydata.cor, tl.col = "black", tl.srt = 45, method = "square",number.cex=0.65, addCoef.col="black")
```

```
#criar train/test/validation para testar inicialmente o Random Forest

TrainSet <- shape_16_17_df
TestSet  <- shape_17_18_df
ValidSet <- shape_18_19_df



##############
############## TUNING SIMPLES #####APENAS PARA TESTE RAPIDO DO MODELO/CODIGO
##############
#Tuning do mtry pela função do caret

tic()
set.seed(123)

Tune <- tuneRF(
  x          = TrainSet[-c(1,36)],
  y          = TrainSet$Def,
  ntreeTry   = 1000,
  mtryStart  = 10,
  stepFactor = 1.5,
  improve    = 0.005,
  trace      = TRUE, # to show real-time progress
  plot       = TRUE
)
toc()

Tune

##############
############## TUNING DETALHADO
##############
############## Bastante demorado, para teste do codigo rodar o simples

####tunar o modelo por grid - mtry
tic()
trControl <- trainControl(method = "cv",
                          number = 10,
                          search = "grid")

set.seed(1234)
tuneGrid <- expand.grid(.mtry = c(1: 30))
rf_mtry <- train(Def~.,
                 data = TrainSet[-c(1)],
                 method = "rf",
                 metric = "RMSE",
                 tuneGrid = tuneGrid,
                 trControl = trControl,
                 importance = TRUE,
                 )
print(rf_mtry)
plot(rf_mtry)
toc()
best_mtry <- rf_mtry$bestTune$mtry

#save(rf_mtry, file = "rf_mtry.rda")
load(file = "rf_mtry.rda")

####tunar o modelo por maxnodes, com o best mtry
tic()
store_maxnode <- list()
tuneGrid <- expand.grid(.mtry = best_mtry)
for (maxnodes in c(1: 1000)) {
  set.seed(1234)
  rf_maxnode <- train(Def~.,
                      data = TrainSet[-c(1)],
                      method = "rf",
                      metric = "RMSE",
                      tuneGrid = tuneGrid,
                      trControl = trControl,
                      importance = TRUE,
                      maxnodes = maxnodes
                      )
  current_iteration <- toString(maxnodes)
  store_maxnode[[current_iteration]] <- rf_maxnode
```

```
}
results_mtry <- resamples(store_maxnode)
summary(results_mtry)
toc()

#Ativar abaixo somente para plotar os dados
#options(max.print=1000000)
#save(results_mtry, file = "results_mtry.rda")
#tune_max_nodes <- read.csv("tuning_max_nodes.csv", header = TRUE, sep = ";")
#str(tune_max_nodes)
#ggplot(data = tune_max_nodes, aes(x=ï..RMSE, y=Mean)) + geom_line(colour = "dodgerblue") +
#  labs( y = "RMSE (Cross-Validation)", x = "Max Nodes") +
#  theme(panel.background = element_rect(fill = "white",
#                                        colour = "black",
#                                        size = 0.5, linetype = "solid"),
#        panel.grid.major = element_line(size = 0.5, linetype = 'solid',
#                                        colour = "gray"))


####tunar o modelo por ntrees - com mtry anterior, o maxnodes tem q colocar direto no modelo
#250,300,350,400,450,500,600,700,800,900,1000,1200,1400,1600,1800,2000)

tic()
store_maxtrees <- list()
for (ntree in c(1600, 1700, 1800, 1900, 2000)) {
  set.seed(5678)
  rf_maxtrees <- train(Def~.,
                       data = TrainSet[-c(1)],
                       method = "rf",
                       metric = "RMSE",
                       tuneGrid = tuneGrid,
                       trControl = trControl,
                       importance = TRUE,
                       ntree = ntree)
  key <- toString(ntree)
  store_maxtrees[[key]] <- rf_maxtrees
}
results_tree <- resamples(store_maxtrees)
summary(results_tree)
toc()

#Ativar abaixo somente para plotar os dados
#options(max.print=1000000)
#tune_ntree <- read.csv("tuning_ntrees.csv", header = TRUE, sep = ";")
#str(tune_ntree)
#ggplot(data = tune_ntree, aes(x=ï..ntree, y=mean)) + geom_line(colour = "dodgerblue") +
#  labs( y = "RMSE (Cross-Validation)", x = "Number of trees") +
#  theme(panel.background = element_rect(fill = "white",
#                                        colour = "black",
#                                        size = 0.5, linetype = "solid"),
#        panel.grid.major = element_line(size = 0.5, linetype = 'solid',
#                                        colour = "gray"))


#############construir o modelo calibrado###############
#Treina um modelo do Random Forest no arquivo de treinamento
#avalia a importancia das variáveis e os resultados no conjunto de teste

tic()

Model1 <- randomForest(
  formula = Def ~ .,
  data = TrainSet[-c(1)], # remove o Id do treinamento
  importance = TRUE, # Se colocar TRUE calcula tanto o INCNODEPURITY quanto o %INCMSE
  ntree = 1200,
  mtry = 7)

Model1

print(Model1)
toc()

#importancia de variaveis

varImpPlot(Model1,
           sort = TRUE,
```

```
        main = 'Variable Importance - Cond 2018',
        n.var = 34) #Conjunto tem 36 variaveis (tira o ID e a resposta)


#achar o numero da arvore com menor MSE - Nem precisa rodar
#which.min(model1$mse)
#sqrt(model1$mse[which.min(model1$mse)])

#importancia de variaveis pela Taxa de Ganho de Info (GainRatio - Entropia)

weights <- data.frame(gain.ratio(Def~., TrainSet[-c(1)]))
str(weights)
weights2 <- melt(as.matrix(weights))

#Ativar abaixo somente para plotar os dados
#ggplot(data = weights2, aes(reorder(Var1, value), value)) + geom_point() + coord_flip() +
#  labs( y = "Gain Ratio Value", x = "Feature", title = "Gain Ratio Variable Importance - Cond 2018") +
#  theme(panel.background = element_rect(fill = "white",
#                                        colour = "black",
#                                        size = 0.5, linetype = "solid"),
#        panel.grid.major = element_line(size = 0.5, linetype = 'solid',
#                                        colour = "gray"),
#        plot.title = element_text(hjust = 0.5))


#Não utilizado
#seleção de subconjunto otimo pelo Correlation Feature Selection (CFS)
#subset <- cfs(Def~., TrainSet[-c(1)])
#f <- as.simple.formula(subset, "Def")
#print(subset)

##### Faz a predição do Modelo acima para os dados de treinamento,
##### validação e predição 2020

#dividir o conjunto de teste com os dados e com a variável resposta
x_test <- TestSet[setdiff(names(TestSet), "Def")]
y_test <- TestSet$Def

x_valid <- ValidSet[setdiff(names(ValidSet), "Def")]
y_valid <- ValidSet$Def

x_2020  <- predict_20_df

# predição com o conjunto de teste
pred_test = predict(Model1, x_test[-c(1)])

# predição com o conjunto de validação
pred_val = predict(Model1, x_valid[-c(1)])

# predição com o conjunto de validação
pred_2020 = predict(Model1, x_2020[-c(1)])

# salvando as variaveis de saida para integrar com o shape depois

options(scipen=999)

Output_predictions = as.data.frame(cbind(NumId = TrainSet$NumId,
                          train_value = Model1$y,
                          pred_train = Model1$predicted,
                          id_test = x_test$NumId,
                          test_value = y_test,
                          pred_test = pred_test,
                          id_validation = x_valid$NumId,
                          valid_value = y_valid,
                          pred_valid = pred_val,
                          id_pred = x_2020$NumId,
                          Pred_2020 = pred_2020))

str(Output_predictions)

write.csv(Output_predictions, "resultados_17_18_19_20.csv")

summary(Output_predictions)


#MSE E RMSE de treinamento pelo arquivo de saida (para comparar tirando direto dos modelos)
```

```
MSE_train_output = (sum((Output_predictions$train_value -
Output_predictions$pred_train)^2))/length(Output_predictions$pred_train)
MSE_train_output

RMSE_train_output = sqrt(MSE_train_output)
RMSE_train_output

#MSE E RMSE de teste pelo arquivo de saida (para comparar tirando direto dos modelos)
MSE_test_output = (sum((Output_predictions$test_value -
Output_predictions$pred_test)^2))/length(Output_predictions$pred_test)
MSE_test_output

RMSE_test_output = sqrt(MSE_test_output)
RMSE_test_output

#MSE E RMSE de teste pelo arquivo de saida (para comparar tirando direto dos modelos)
MSE_valid_output = (sum((Output_predictions$valid_value -
Output_predictions$pred_valid)^2))/length(Output_predictions$pred_valid)
MSE_valid_output

RMSE_valid_output = sqrt(MSE_valid_output)
RMSE_valid_output

##### Aki nao precisa rodar era só para ver se estava certo com os dados de cima
# RMSE do modelo no treinamento ok, bateu com o de cima
#MSE_Train = mean(Model1$mse)
#MSE_Train
#
#RMSE_Train = sqrt(MSE_Train)
#RMSE_Train


# estatísticas da performance do algoritmo no cojnuto de teste (ok bateu com la em cima)
# soma dos erros quadráticos
#MSE_test = (sum((y_test - pred_test)^2))/length(pred_test)
#MSE_test

# erro quadrático médio
#RMSE_test = sqrt(MSE_test)
#RMSE_test

# estatísticas da performance do algoritmo no cojnuto de validacao, ok com o de cima tb
# soma dos erros quadráticos
#MSE_valid = (sum((y_valid - pred_val)^2))/length(pred_val)
#MSE_valid

# erro quadrático médio
#RMSE_valid = sqrt(MSE_valid)
#RMSE_valid


# Plot predictions vs Training data
ggplot(Output_predictions,aes(pred_train, train_value)) +
  geom_point(color = "darkred", alpha = 0.5) +
  geom_smooth(method=lm)+
  ggtitle("RF: prediction vs training data (RMSE 8,09*10^(-3)") +
  xlab("prediction ") +
  ylab("Training data") +
  theme(plot.title = element_text(color="darkgreen",size=18,hjust = 0.5),
        axis.text.y = element_text(size=12),
        axis.text.x = element_text(size=12,hjust=.5),
        axis.title.x = element_text(size=14),
        axis.title.y = element_text(size=14))

# Plot predictions vs Test data
ggplot(Output_predictions,aes(pred_test, test_value)) +
  geom_point(color = "darkred", alpha = 0.5) +
  geom_smooth(method=lm)+
  ggtitle("RF: prediction vs test data (RMSE 11,39*10^(-3))") +
  xlab("prediction ") +
  ylab("Test data") +
  theme(plot.title = element_text(color="darkgreen",size=18,hjust = 0.5),
        axis.text.y = element_text(size=12),
        axis.text.x = element_text(size=12,hjust=.5),
        axis.title.x = element_text(size=14),
        axis.title.y = element_text(size=14))
```

```
# Plot predictions vs validation data
ggplot(Output_predictions,aes(pred_valid, valid_value)) +
  geom_point(color = "darkred", alpha = 0.5) +
  geom_smooth(method=lm)+
  ggtitle("RF: prediction vs validation data (RMSE 8,82*10^(-3))") +
  xlab("prediction ") +
  ylab("Validation data") +
  theme(plot.title = element_text(color="darkgreen",size=18,hjust = 0.5),
        axis.text.y = element_text(size=12),
        axis.text.x = element_text(size=12,hjust=.5),
        axis.title.x = element_text(size=14),
        axis.title.y = element_text(size=14))


##########TENHO QUE VER COMO ENCAIXAR ESSA PARTE JUNTO COM A DE CIMA, SE NAO FICA SEPARADO
MESMO
##########MAS EH COMPUTACIONALMENTE RUIM
##########ACHO QUE TEM Q FAZER O TEST JUNTO PARA PLOTAR AS RF

Model1_v2 <- randomForest(
  formula = Def ~ .,
  data = TrainSet[-c(1)],# remove o Id do treinamento
  ytest = y_test,
  xtest = x_test[-c(1)],
  importance = TRUE, # Se colocar TRUE calcula tanto o INCNODEPURITY quanto o %INCMSE
  ntree = 1200,
  mtry = 7)

Model1_v2


##############Testando no conjunto de validação

Model1_v3 <- randomForest(
  formula = Def ~ .,
  data = TrainSet[-c(1)],# remove o Id do treinamento
  ytest = y_valid,
  xtest = x_valid[-c(1)],
  importance = TRUE, # Se colocar TRUE calcula tanto o INCNODEPURITY quanto o %INCMSE
  ntree = 1200,
  mtry = 7)

Model1_v3


# extract OOB & validation errors
oob <- sqrt(Model1_v3$mse)
test_error <- sqrt(Model1_v2$test$mse)
validation <- sqrt(Model1_v3$test$mse)



# compare error rates
tibble::tibble(
  `Training set error` = oob,
  `Test set error` = test_error,
  `Validation set error` = validation,
  ntrees = 1:Model1_v3$ntree
) %>%
  gather(Metric, RMSE, -ntrees) %>%
  ggplot(aes(ntrees, RMSE, color = Metric)) +
  geom_line() +
  scale_y_continuous() +
  xlab("Number of trees")


#####avaliações estatisticas

summary(Output_predictions$train_value)
summary(log10(Output_predictions$train_value*100))
summary(log10(Output_predictions$train_value*100+1))
plot(density(log10(Output_predictions$train_value*100+1)))
plot(density(sqrt(Output_predictions$train_value)))

qplot(x = train_value, data = Output_predictions)
```

```
qplot(x = log10(train_value+1), data = Output_predictions)
qplot(x = log(train_value+1), data = Output_predictions)
qplot(x = log(train_value+ sqrt(train_value^2 +1)), data = Output_predictions)
qplot(x = 1/train_value, data = Output_predictions)
qplot(x = 1/(sqrt(train_value)), data = Output_predictions)
qplot(x = sqrt(train_value), data = Output_predictions)
qplot(x = 1/log10(train_value+1), data = Output_predictions)


ln(x + \sqrt(x^2 + 1))

test_log_transform <- log10(Output_predictions$train_value)


hist(Output_predictions$train_value, breaks=50, col="red")
plot(density(Output_predictions$train_value))

hist((log10(Output_predictions$train_value + 1)), breaks=50, col="red")
plot(density(log(Output_predictions$train_value)))


ggplot(data = Output_predictions, aes(x = pred_train, y = train_value)) +
  geom_point() +
  scale_x_log10() + scale_y_log10()
```

Spatial Random Forest com Kernel Adaptativo

```
# Carregar bibliotecas

library(sf)
library(raster)
library(dplyr)
library(rgdal)
library(tidyverse)
library(tictoc)
library(randomForest)
library(caret)
library(ggplot2)
library(SpatialML)
library(reshape2)
library(data.table)


#limpar ambiente
cat("\f")
rm(list = ls())

#Setar a pasta
setwd("E:/Posdoc_ITV/Base_qgis/999_Master_Join/Modelagem_002/backup")
getwd()

# Carregar o shape de 2016_2017 - Treinamento
# Gera uma Df com as coordenadas separadas pois eh necessário para avaliar o GRF
shape_16_17 <- st_read("Cond_2016_def_2017_coords.shp")
str(shape_16_17)
shape_16_17_df <- data.frame(shape_16_17)

coords_shape_16_17_df <- shape_16_17_df[c(39,40)]
names(coords_shape_16_17_df)[names(coords_shape_16_17_df) == 'Long_4326'] <- 'X'
names(coords_shape_16_17_df)[names(coords_shape_16_17_df) == 'Lat_4326'] <- 'Y'
str(coords_shape_16_17_df)

shape_16_17_df <- shape_16_17_df[-c(41,40,39,38,37)]
str(shape_16_17_df)


# Carregar o shape de 2017_2018 - Teste
# Gera uma Df com as coordenadas separadas pois eh necessário para avaliar o GRF
```

```
shape_17_18 <- st_read("Cond_2017_def_2018_coords.shp")
str(shape_17_18)
shape_17_18_df <- data.frame(shape_17_18)

coords_shape_17_18_df <- shape_17_18_df[c(39,40)]
names(coords_shape_17_18_df)[names(coords_shape_17_18_df) == 'Long_4326'] <- 'X'
names(coords_shape_17_18_df)[names(coords_shape_17_18_df) == 'Lat_4326'] <- 'Y'
str(coords_shape_17_18_df)

# Carregar o shape de 2018_2019 (Já removidos: Pc_IL, Pc_PA, Pc_Ast, Lat, Long) - Validação
# Gera uma Df com as coordenadas separadas pois eh necessário para avaliar o GRF

shape_18_19 <- st_read("Cond_2018_def_2019_coords.shp")
str(shape_18_19)
shape_18_19_df <- data.frame(shape_18_19)

coords_shape_18_19_df <- shape_18_19_df[c(39,40)]
names(coords_shape_18_19_df)[names(coords_shape_18_19_df) == 'Long_4326'] <- 'X'
names(coords_shape_18_19_df)[names(coords_shape_18_19_df) == 'Lat_4326'] <- 'Y'
str(coords_shape_18_19_df)

# Carregar o shape de 2019 (Previsão 2020)
# Gera uma Df com as coordenadas separadas pois eh necessário para avaliar o GRF

predict_20 <- st_read("predict_2020_cond_2019_coords.shp")
str(predict_20)
predict_20_df <- data.frame(predict_20)

coords_predict_20_df <- predict_20_df[c(39,40)]
names(coords_predict_20_df)[names(coords_predict_20_df) == 'Long_4326'] <- 'X'
names(coords_predict_20_df)[names(coords_predict_20_df) == 'Lat_4326'] <- 'Y'
str(coords_predict_20_df)


#############construir o GRF###############
#https://www.tandfonline.com/doi/full/10.1080/10106049.2019.1595177
#Treina um modelo do Random Forest no arquivo de treinamento
#avalia a importancia das variáveis e os resultados no conjunto de teste

options(scipen=999999)
memory.limit (9999999999)

tic()

grf <- grf(Def ~ Ar_IL + Dst_IL + Ar_PA + Dst_PA +
        Sum_Wat + Dst_Wat + Sum_Riv + Dst_Riv +
        Sum_FR_SR + Dst_AR + IDH + IDH_Rn + IDH_Ed + IDH_Lg +
        POP + GDP + GDPPC + ARI + Elevation + Slope +
        PcNat + PcWat + PcAgr + PcPas + PcUrb + PcMin +
        DstUrb + DstMin + Dst_Agr + DstPst + FED +
        Dst_PM_km + Ar_Ast_km + Dst_Ast_km,          #Variavel alvo ~ Preditores (tem q por um a um)
        dframe=shape_16_17_df,                       #Df de entrada
        bw=100,                                  #Parametro do modelo ()
        kernel="adaptive",                          #Parametro do modelo () pode mudar para "fixed" e colocar um raio ao invés do
numero de vizinhos
        coords=coords_shape_16_17_df,                    #Df com coordenadas
        mtry = 7,                                #Igual Caret (obtido da calibracao do RF normal)
        ntree = 1200,                              #Igual Caret (obtido da calibracao do RF normal)
        )

toc()

#calibration of the bw (started with mtry and ntree from normal RF)
#has to be done manually (feito de 0-200)

#avaliando o modelo nos dados de 2018 (100% do modelo local)
tic()
pred_18 <- predict.grf(grf, shape_17_18_df, x.var.name="Long_4326", y.var.name="Lat_4326", local.w=1, global.w=0)
toc()
tic()
pred_19 <- predict.grf(grf, shape_18_19_df, x.var.name="Long_4326", y.var.name="Lat_4326", local.w=1, global.w=0)
toc()
tic()
pred_20 <- predict.grf(grf, predict_20_df, x.var.name="Long_4326", y.var.name="Lat_4326", local.w=1, global.w=0)
toc()
```

```
#gera um csv com as predições
predictions <- as.data.frame(cbind( NumId = shape_16_17_df$NumI,
                                    train_value = grf$Global.Model$y,
                                    train_pred_global = grf$Global.Model$predicted,
                                    train_pred_local = grf$LGofFit$LM_yfitPred,
                                    train_2018 = shape_17_18_df$Def,
                                    pred_2018_local = pred_18,
                                    train_2019 = shape_18_19_df$Def,
                                    pred_2019_local = pred_19,
                                    pred_2020_local = pred_20))

#MSE E RMSE de treinamento pelo arquivo de saida (para comparar tirando direto dos modelos)
RSS_train_output = (sum((predictions$train_value - predictions$train_pred_local)^2))
RSS_train_output

MSE_train_output = RSS_train_output/length(predictions$train_pred_local)
MSE_train_output

RMSE_train_output = sqrt(MSE_train_output)
RMSE_train_output

#MSE E RMSE de treinamento pelo arquivo de saida (para comparar tirando direto dos modelos)
RSS_train_output_2018 = (sum((predictions$train_2018 - predictions$pred_2018_local)^2))
RSS_train_output_2018

MSE_train_output_2018 = RSS_train_output_2018/length(predictions$pred_2018_local)
MSE_train_output_2018

RMSE_train_output_2018 = sqrt(MSE_train_output_2018)
RMSE_train_output_2018

#MSE E RMSE de treinamento pelo arquivo de saida (para comparar tirando direto dos modelos)
RSS_train_output_2019 = (sum((predictions$train_2019 - predictions$pred_2019_local)^2))
RSS_train_output_2019

MSE_train_output_2019 = RSS_train_output_2019/length(predictions$pred_2019_local)
MSE_train_output_2019

RMSE_train_output_2019 = sqrt(MSE_train_output_2019)
RMSE_train_output_2019

########salva o csv com as predições de saida#######

write.csv(predictions, "resultados_GRF_bw_100.csv")

######salva o modelo de random forest#####

save(grf,file = "grf_bw100.rda")

#ate aki ok, vamos testar um predict:

######plotar os resultados do treinamento (todos salvos em csv)
######precisa do csv da etapa anterior!!!

bw_train <- data.frame(read.csv("bw_cal_train.csv", header = TRUE, sep = ';'))

#tempo calibracao
sum(bw_train$time)/3600

#plot
ggplot(data = bw_train, aes(x = bw_train$Bw.4326., y = bw_train$RMSE)) + geom_point() +
  geom_line(color = "blue") +
labs( y = "RMSE", x = "Bandwidth (NN)", title = "RMSE vs BW - Train (2017)") +
    theme(panel.background = element_rect(fill = "white",
                                          colour = "black",
                                          size = 0.5, linetype = "solid"),
          panel.grid.major = element_line(size = 0.5, linetype = 'solid',
                                          colour = "gray"),
          plot.title = element_text(hjust = 0.5))



bw_val <- data.frame(read.csv("bw_cal_test.csv", header = TRUE, sep = ';'))
#tempo perdicao 2018
sum(bw_val$time_18)/3600
```

```
#tempo perdicao 2019
sum(bw_val$time19)/3600
#tempo perdicao 2020
sum(bw_val$time20)/3600



ggplot(data = bw_val, aes(x = bw_val$Bw.4326., y = bw_val$RMSE_2018_set)) + geom_point() +
  geom_line(color = "blue") +
  labs( y = "RMSE", x = "Bandwidth (NN)", title = "RMSE vs BW - Test (2018)") +
  theme(panel.background = element_rect(fill = "white",
                                        colour = "black",
                                        size = 0.5, linetype = "solid"),
        panel.grid.major = element_line(size = 0.5, linetype = 'solid',
                                        colour = "gray"),
        plot.title = element_text(hjust = 0.5))



ggplot(data = bw_val, aes(x = bw_val$Bw.4326., y = bw_val$RMSE_2019_set)) + geom_point() +
  geom_line(color = "blue") +
  labs( y = "RMSE", x = "Bandwidth (NN)", title = "RMSE vs BW - Valid (2019)") +
  theme(panel.background = element_rect(fill = "white",
                                        colour = "black",
                                        size = 0.5, linetype = "solid"),
        panel.grid.major = element_line(size = 0.5, linetype = 'solid',
                                        colour = "gray"),
        plot.title = element_text(hjust = 0.5))


#importancia de variaveis

Var_importance <- data.frame(row.names(grf$LocalModelSummary$l.IncMSE), grf$LocalModelSummary$l.IncMSE,
grf$LocalModelSummary$l.IncNodePurity)

ggplot(data = Var_importance, aes(reorder(row.names.grf.LocalModelSummary.l.IncMSE., Mean), Mean)) + geom_point() +
coord_flip() +
  labs( y = "Mean_IncMSE", x = "Feature", title = "Variable Importance - Train 2017") +
  theme(panel.background = element_rect(fill = "white",
                                        colour = "black",
                                        size = 0.5, linetype = "solid"),
        panel.grid.major = element_line(size = 0.5, linetype = 'solid',
                                        colour = "gray"),
        plot.title = element_text(hjust = 0.5))


ggplot(data = Var_importance, aes(reorder(row.names.grf.LocalModelSummary.l.IncMSE., Mean.1), Mean.1)) + geom_point() +
coord_flip() +
  labs( y = "Mean_IncNodePurity", x = "Feature", title = "Variable Importance - Train 2017") +
  theme(panel.background = element_rect(fill = "white",
                                        colour = "black",
                                        size = 0.5, linetype = "solid"),
        panel.grid.major = element_line(size = 0.5, linetype = 'solid',
                                        colour = "gray"),
        plot.title = element_text(hjust = 0.5))

#exportar a importância de variaveis por celula:


Local_IncMSE_2017 <- data.frame(grf$Local.Pc.IncMSE, shape_16_17$NumId)
Local_NodePur_2017 <- data.frame(grf$Local.IncNodePurity, shape_16_17$NumId)

write.csv(Local_IncMSE_2017, "Local_IncMSE_2017.csv")
write.csv(Local_NodePur_2017, "Local_NodePur_2017.csv")


####Seleciona a variavel com maior IncMSE

DT <- data.table(Local_IncMSE_2017)

  DT[, col_max := colnames(.SD)[max.col(.SD, ties.method = "first")], .SDcols = c(1:34)]

  write_csv(DT, file = "Local_IncMSE_2017_DT.csv")


####Seleciona a variavel com mair NodePur
```

```
DT2 <- data.table(Local_NodePur_2017)

DT2[, col_max := colnames(.SD)[max.col(.SD, ties.method = "first")], .SDcols = c(1:34)]

write_csv(DT2, file = "Local_NodePur_2017_DT.csv")
```

Spatial Random Forest com Kernel Fixo

```
# Carregar bibliotecas

library(sf)
library(raster)
library(dplyr)
library(rgdal)
library(tidyverse)
library(tictoc)
library(randomForest)
library(caret)
library(ggplot2)
library(SpatialML)
library(reshape2)
library(data.table)


#limpar ambiente
cat("\f")
rm(list = ls())

#Setar a pasta
setwd("E:/Posdoc_ITV/Base_qgis/999_Master_Join/Modelagem_002/backup")
getwd()

# Carregar o shape de 2016_2017 - Treinamento
# Gera uma Df com as coordenadas separadas pois eh necessário para avaliar o GRF
shape_16_17 <- st_read("Cond_2016_def_2017_coords.shp")
str(shape_16_17)
shape_16_17_df <- data.frame(shape_16_17)

coords_shape_16_17_df <- shape_16_17_df[c(38,37)]
names(coords_shape_16_17_df)[names(coords_shape_16_17_df) == 'Long_5880'] <- 'X'
names(coords_shape_16_17_df)[names(coords_shape_16_17_df) == 'Lat_5880'] <- 'Y'
str(coords_shape_16_17_df)

shape_16_17_df <- shape_16_17_df[-c(41,40,39,38,37)]
str(shape_16_17_df)


# Carregar o shape de 2017_2018 - Teste
#shape_17_18_df <- shape_17_18_df[-c(37)]
#str(shape_17_18_df)

shape_17_18 <- st_read("Cond_2017_def_2018_coords.shp")
str(shape_17_18)
shape_17_18_df <- data.frame(shape_17_18)

coords_shape_17_18_df <- shape_17_18_df[c(38,37)]
names(coords_shape_17_18_df)[names(coords_shape_17_18_df) == 'Long_5880'] <- 'X'
names(coords_shape_17_18_df)[names(coords_shape_17_18_df) == 'Lat_5880'] <- 'Y'
str(coords_shape_17_18_df)

# Carregar o shape de 2018_2019 (Já removidos: Pc_IL, Pc_PA, Pc_Ast, Lat, Long) - Validação

shape_18_19 <- st_read("Cond_2018_def_2019_coords.shp")
str(shape_18_19)
shape_18_19_df <- data.frame(shape_18_19)

coords_shape_18_19_df <- shape_18_19_df[c(38,37)]
```

```
names(coords_shape_18_19_df)[names(coords_shape_18_19_df) == 'Long_5880'] <- 'X'
names(coords_shape_18_19_df)[names(coords_shape_18_19_df) == 'Lat_5880'] <- 'Y'
str(coords_shape_18_19_df)

# Carregar o shape de 2019 (Previsão 2020)

predict_20 <- st_read("predict_2020_cond_2019_coords.shp")
str(predict_20)
predict_20_df <- data.frame(predict_20)

coords_predict_20_df <- predict_20_df[c(37,36)]
names(coords_predict_20_df)[names(coords_predict_20_df) == 'Long_5880'] <- 'X'
names(coords_predict_20_df)[names(coords_predict_20_df) == 'Lat_5880'] <- 'Y'
str(coords_predict_20_df)


#############construir o GRF###############
#https://www.tandfonline.com/doi/full/10.1080/10106049.2019.1595177
#Treina um modelo do Random Forest no arquivo de treinamento
#avalia a importancia das variáveis e os resultados no conjunto de teste

options(scipen=999999)
memory.limit (9999999999)

tic()

grf <- grf(Def ~ Ar_IL + Dst_IL + Ar_PA + Dst_PA +
        Sum_Wat + Dst_Wat + Sum_Riv + Dst_Riv +
        Sum_FR_SR + Dst_AR + IDH + IDH_Rn + IDH_Ed + IDH_Lg +
        POP + GDP + GDPPC + ARI + Elevation + Slope +
        PcNat + PcWat + PcAgr + PcPas + PcUrb + PcMin +
        DstUrb + DstMin + Dst_Agr + DstPst + FED +
        Dst_PM_km + Ar_Ast_km + Dst_Ast_km,          #Variavel alvo ~ Preditores (tem q por um a um)
        dframe=shape_16_17_df,                       #Df de entrada
        bw=200100,                                   #Parametro do modelo ()
        kernel="fixed",                              #Parametro do modelo ()
        coords=coords_shape_16_17_df,                #Df com coordenadas
        mtry = 7,
        ntree = 1200,
        )

toc()

#calibration of the bw (started with mtry and ntree from normal RF)
#has to be done manually

#avaliando o modelo nos dados de 2018 (100% do modelo local)
tic()
pred_18 <- predict.grf(grf, shape_17_18_df, x.var.name="Long_5880", y.var.name="Lat_5880", local.w=1, global.w=0)
toc()
tic()
pred_19 <- predict.grf(grf, shape_18_19_df, x.var.name="Long_5880", y.var.name="Lat_5880", local.w=1, global.w=0)
toc()
tic()
pred_20 <- predict.grf(grf, predict_20_df, x.var.name="Long_5880", y.var.name="Lat_5880", local.w=1, global.w=0)
toc()


predictions <- as.data.frame(cbind( NumId = shape_16_17_df$NumI,
                                    train_value = grf$Global.Model$y,
                                    train_pred_global = grf$Global.Model$predicted,
                                    train_pred_local = grf$LGofFit$LM_yfitPred,
                                    train_2018 = shape_17_18_df$Def,
                                    pred_2018_local = pred_18,
                                    train_2019 = shape_18_19_df$Def,
                                    pred_2019_local = pred_19,
                                    pred_2020_local = pred_20))

#MSE E RMSE de treinamento pelo arquivo de saida (para comparar tirando direto dos modelos)
RSS_train_output = (sum((predictions$train_value - predictions$train_pred_local)^2))
RSS_train_output

MSE_train_output = RSS_train_output/length(predictions$train_pred_local)
MSE_train_output

RMSE_train_output = sqrt(MSE_train_output)
```

```
RMSE_train_output

#MSE E RMSE de treinamento pelo arquivo de saida (para comparar tirando direto dos modelos)
RSS_train_output_2018 = (sum((predictions$train_2018 - predictions$pred_2018_local)^2))
RSS_train_output_2018

MSE_train_output_2018 = RSS_train_output_2018/length(predictions$pred_2018_local)
MSE_train_output_2018

RMSE_train_output_2018 = sqrt(MSE_train_output_2018)
RMSE_train_output_2018

#MSE E RMSE de treinamento pelo arquivo de saida (para comparar tirando direto dos modelos)
RSS_train_output_2019 = (sum((predictions$train_2019 - predictions$pred_2019_local)^2))
RSS_train_output_2019

MSE_train_output_2019 = RSS_train_output_2019/length(predictions$pred_2019_local)
MSE_train_output_2019

RMSE_train_output_2019 = sqrt(MSE_train_output_2019)
RMSE_train_output_2019

########salva o csv com as predições de saida#######

write.csv(predictions, "resultados_GRF_bw_200km.csv")

######salva o modelo de random forest#####

save(grf,file = "grf_bw200km.rda")

#ate aki ok, vamos testar um predict:

######plotar os resultados do treinamento (todos salvos em csv)

bw_train <- data.frame(read.csv("bw_cal_train_km.csv", header = TRUE, sep = ';'))
#tempo calibracao
sum(bw_train$time)/3600
#plot
ggplot(data = bw_train, aes(x = Bw.4326., y = RMSE)) + geom_point() +
  geom_line(color = "blue") +
labs( y = "RMSE", x = "Bandwidth (Km)", title = "RMSE vs BW - Train (2017)") +
    theme(panel.background = element_rect(fill = "white",
                                          colour = "black",
                                          size = 0.5, linetype = "solid"),
          panel.grid.major = element_line(size = 0.5, linetype = 'solid',
                                          colour = "gray"),
          plot.title = element_text(hjust = 0.5))



bw_val <- data.frame(read.csv("bw_cal_test.csv", header = TRUE, sep = ';'))
#tempo perdicao 2018
sum(bw_val$time_18)/3600
#tempo perdicao 2019
sum(bw_val$time19)/3600
#tempo perdicao 2020
sum(bw_val$time20)/3600



ggplot(data = bw_val, aes(x = Bw.4326., y = RMSE_2018_set)) + geom_point() +
  geom_line(color = "blue") +
  labs( y = "RMSE", x = "Bandwidth (Km)", title = "RMSE vs BW - Test (2018)") +
  theme(panel.background = element_rect(fill = "white",
                                        colour = "black",
                                        size = 0.5, linetype = "solid"),
        panel.grid.major = element_line(size = 0.5, linetype = 'solid',
                                        colour = "gray"),
        plot.title = element_text(hjust = 0.5))



ggplot(data = bw_val, aes(x = Bw.4326., y = RMSE_2019_set)) + geom_point() +
  geom_line(color = "blue") +
  labs( y = "RMSE", x = "Bandwidth (Km)", title = "RMSE vs BW - Valid (2019)") +
  theme(panel.background = element_rect(fill = "white",
```

```
                                colour = "black",
                                size = 0.5, linetype = "solid"),
        panel.grid.major = element_line(size = 0.5, linetype = 'solid',
                                colour = "gray"),
        plot.title = element_text(hjust = 0.5))


#importancia de variaveis

Var_importance <- data.frame(row.names(grf$LocalModelSummary$l.IncMSE), grf$LocalModelSummary$l.IncMSE,
grf$LocalModelSummary$l.IncNodePurity)

ggplot(data = Var_importance, aes(reorder(row.names.grf.LocalModelSummary.l.IncMSE., Mean), Mean)) + geom_point() +
coord_flip() +
  labs( y = "Mean_IncMSE", x = "Feature", title = "Variable Importance - Train 2017") +
  theme(panel.background = element_rect(fill = "white",
                                colour = "black",
                                size = 0.5, linetype = "solid"),
        panel.grid.major = element_line(size = 0.5, linetype = 'solid',
                                colour = "gray"),
        plot.title = element_text(hjust = 0.5))


ggplot(data = Var_importance, aes(reorder(row.names.grf.LocalModelSummary.l.IncMSE., Mean.1), Mean.1)) + geom_point() +
coord_flip() +
  labs( y = "Mean_IncNodePurity", x = "Feature", title = "Variable Importance - Train 2017") +
  theme(panel.background = element_rect(fill = "white",
                                colour = "black",
                                size = 0.5, linetype = "solid"),
        panel.grid.major = element_line(size = 0.5, linetype = 'solid',
                                colour = "gray"),
        plot.title = element_text(hjust = 0.5))

#exportar a importância de variaveis por celula:


Local_IncMSE_2017 <- data.frame(grf$Local.Pc.IncMSE, shape_16_17$NumId)
Local_NodePur_2017 <- data.frame(grf$Local.IncNodePurity, shape_16_17$NumId)

write.csv(Local_IncMSE_2017, "Local_IncMSE_2017_km.csv")
write.csv(Local_NodePur_2017, "Local_NodePur_2017_km.csv")


####Seleciona a variavel com mair IncMSE

DT <- data.table(Local_IncMSE_2017)

  DT[, col_max := colnames(.SD)[max.col(.SD, ties.method = "first")], .SDcols = c(1:34)]

  write_csv(DT, file = "Local_IncMSE_2017_DT_Km.csv")


####Seleciona a variavel com mair NodePur

  DT2 <- data.table(Local_NodePur_2017)

  DT2[, col_max := colnames(.SD)[max.col(.SD, ties.method = "first")], .SDcols = c(1:34)]

  write_csv(DT2, file = "Local_NodePur_2017_DT_Km.csv")
```

INTEGRATED NESTE LAPLACE APPROXIMATIONS

```
## Clean workspace
rm(list=ls())
gc()

## Source utilities
library(tidyverse)
library(lares) # devtools::install_github("laresbernardo/lares")
library(sp)
library(sf)
library(rgdal)
```

```
library(raster)
library(tictoc)
library(INLA)
library(viridis)
library(gridExtra)
library(scales)
#source("Deforestation_modutil_v4.R")
library(lme4)
library(MuMIn)
library(gstat)
library(usdm)



## Clean workspace
rm(list=ls())
gc()

## Source utilities

#opcoes de memoria/notacao cientifica
options(scipen=999999)
memory.limit (9999999999)

#Setar a pasta
setwd("E:/Posdoc_ITV/Base_qgis/999_Master_Join/Modelagem_003/backup")
getwd()


#### Load Deforestation dataset 2016-2017
shape_16_17 <- st_read("Cond_2016_def_2017_coords_NDef.shp")
str(shape_16_17)
shape_16_17_df <- data.frame(shape_16_17) %>% rename(grid.ID = NumId) %>%
  rename (Def_Pix = DfP_16_17) %>%
  dplyr::select(-c(Def, Long_5880, Lat_5880, geometry)) %>%
  mutate(grid.ID = as.factor(grid.ID))

## Scale predictor variables
##shape_16_17_df[, c(2:35)] <- scale(shape_16_17_df[, c(2:35)])

#### Load Deforestation dataset 2017-2018
shape_17_18 <- st_read("Cond_2017_def_2018_coords_NDef.shp")
str(shape_17_18)
shape_17_18_df <- data.frame(shape_17_18) %>% rename(grid.ID = NumId) %>%
  rename (Def_Pix = DfP_17_18) %>%
  dplyr::select(-c(Def, Long_5880, Lat_5880, geometry)) %>%
  mutate(grid.ID = as.factor(grid.ID))

#### Load Deforestation dataset 2018-2019
shape_18_19 <- st_read("Cond_2018_def_2019_coords_NDef.shp")
str(shape_18_19)
shape_18_19_df <- data.frame(shape_18_19) %>% rename(grid.ID = NumId) %>%
  rename (Def_Pix = DfP_18_19) %>%
  dplyr::select(-c(Def, Long_5880, Lat_5880, geometry)) %>%
  mutate(grid.ID = as.factor(grid.ID))

#### Load Prediction dataset 2019-2020
shape_19_20 <- st_read("predict_2020_cond_2019_coords.shp")
str(shape_19_20)
shape_19_20_df <- data.frame(shape_19_20) %>% rename(grid.ID = NumId) %>%
  dplyr::select(-c(Long_5880, Lat_5880, geometry)) %>%
  mutate(grid.ID = as.factor(grid.ID))



######## Predict reference data
# https://www.paulamoraga.com/book-geospatial/sec-geostatisticaldataexamplespatial.html
mesh <- inla.mesh.2d(loc = shape_16_17_df[, c("Long_4326", "Lat_4326")], max.edge=c(1, 4),cutoff=1, offset=c(-0.1,-0.2))
plot(mesh, asp=1)
points(shape_16_17_df[, c("Long_4326", "Lat_4326")], col="red", pch=19)

## Formulas - After cross-validation
hyper.prec <- list(theta = list(prior="pc.prec", param = c(1, 0.05)))

#### Load best model formulas ----
#formula <- y ~ -1 + Intercept +
```

```
#  f(inla.group(FED), model='rw2', hyper=hyper.prec, scale.model = TRUE) +
#  Ar_IL + Dst_IL + Ar_PA +
#  Dst_PA + Sum_Wat + Dst_Wat +
#  f(field, model = spde) + f(grid.ID, model = "iid")

formula <- y ~ -1 + Intercept +
  FED + Ar_IL +Dst_IL +Ar_PA +Dst_PA +Sum_Wat +Dst_Wat +
  Sum_Riv +Dst_Riv +Sum_FR_SR +Dst_AR +IDH +IDH_Rn +IDH_Ed +
  IDH_Lg +POP +GDP +GDPPC +ARI +Elevation +Slope +
  PcNat +PcWat +PcAgr +PcPas +PcUrb +PcMin +DstUrb +
  DstMin +Dst_Agr +DstPst +Dst_PM_km +Ar_Ast_km +Dst_Ast_km +
  f(field, model=spde) + f(grid.ID, model='iid') ## w accounts for SA, grid.ID accounts for OD


#names(shape_16_17_df)

# Define penalised complexity priors for random field.
spde <- INLA::inla.spde2.matern(mesh, alpha=2)
field.indices <- inla.spde.make.index("field", n.spde = mesh$n)

# Make the A matrices
Aest <- INLA::inla.spde.make.A(mesh, loc = as.matrix(shape_16_17_df[, c("Long_4326", "Lat_4326")]))
Apred <- INLA::inla.spde.make.A(mesh, loc = as.matrix(shape_19_20_df[, c("Long_4326", "Lat_4326")]))

# Stack for estimation
stk.est <- inla.stack(tag = "estimation", ## tag
                      data = list(y = shape_16_17_df$Def_Pix, ## Number of deforested pixels
                                  Ntrials = shape_16_17_df$Tot_pix), ## Total number of pixels per grid
                      A = list(Aest, 1),  ## Projector matrix for space, fixed.
                      effects = list(field = field.indices,
                                     cbind(Intercept = 1, shape_16_17_df)))

# Stack for prediction
stk.pred <- inla.stack(tag = "prediction", ## tag
                       data = list(y = NA, ## Number of deforested pixels
                                   Ntrials = NA), ## Total number of pixels per grid
                       A = list(Apred, 1),  ## Projector matrix for space, fixed.
                       effects = list(field = field.indices,
                                      cbind(Intercept = 1, shape_19_20_df)))

# Full stack
stk.full <- inla.stack(stk.est, stk.pred)

# Run the model
tic()
m <- inla(formula, family = "binomial",
          data = INLA::inla.stack.data(stk.full),
          Ntrials = INLA::inla.stack.data(stk.full)$Ntrials, # can also simply use Ntrials
          #control.family = list(link = "logit"), # does not change predictions
          control.predictor = list(compute = TRUE, link = 1, A = inla.stack.A(stk.full)))
#control.inla = list(int.strategy = "eb", strategy = "gaussian")) # This makes things quicker. Might be a good idea for cv and then
do the most accurate you can for the final model
toc()
save(m, file="INLA_2019_Def_2020.Rdata")
#load(file="INLA_2016_Def_2017.Rdata")

#### Model coeficients ----
Index <- inla.stack.index(stk.full, tag  = "prediction")$data
pred <- m$summary.fitted.values[Index, "mean"]
#actual <- shape_19_20_df$Def_Pix / shape_19_20_df$Tot_pix
actual.index <- shape_19_20_df$grid.ID
hist(pred)

results <- cbind(actual.index, Index, pred)
results <- data.frame(results)

write.csv(results, "INLA_2019_Def_2020.csv")

#MSE E RMSE de treinamento pelo arquivo de saida (para comparar tirando direto dos modelos)
RSS_train_output = (sum((results$actual - results$pred)^2))
RSS_train_output

MSE_train_output = RSS_train_output/length(results$actual)
MSE_train_output

RMSE_train_output = sqrt(MSE_train_output)
```

RMSE_train_output

```
## Check results
summary(m)
m$summary.fixed[, c("mean", "0.025quant", "0.975quant")] ## Fixed effects
m$summary.fitted.values[, "mean"] ## Fitted values
m$summary.hyperpar[, c("mean", "0.025quant", "0.975quant")] ##  Hyperparameters
m$summary.random$field ##  Random effects
m$marginals.fixed ## Posterior marginal distributions for predictors
m$marginals.hyperpar ## Posterior marginal distributions for hyperparameters



# tem q puxar as funções do deforestation modutil
#### plot.mar.fixed ----
## Plot marginal distributions of fixed effects from inla model
plot.mar.fixed <- function(inla.model){
  varnames <- names(inla.model$marginals.fixed)
  for(i in 1: length(varnames)){
    var.mar <- data.frame(inla.model$marginals.fixed[i])
    plot(x = var.mar[,1], y=var.mar[, 2], type="l",
         xlab=paste(names(var.mar)[1]), ylab=paste(names(var.mar)[2]))
    abline(v=0, col="red")
  }
}

#### plot.mar.hyper ----
## Plot marginal distributions of hyperparameters from inla model
plot.mar.hyper <- function(inla.model){
  varnames <- names(inla.model$marginals.hyperpar)
  for(i in 1: length(varnames)){
    var.mar <- data.frame(inla.model$marginals.hyperpar[i])
    plot(x = var.mar[,1], y=var.mar[, 2], type="l",
         xlab=paste(names(var.mar)[1]), ylab=paste(names(var.mar)[2]))
  }
}


plot.mar.fixed(m) ## Custom function to plot marginal distributions of fixed effects
plot.mar.hyper(m) ## Custom function to plot marginal distributions of hyperparameters
```