Projeto de Data Minimg

Criptomoedas

Benedito Baptista
e-mail: 30005111@students.ual.pt

Gonçalo Lemos
e-mail: 30007523@students.ual.pt

Jõao Dumba
e-mail: 30002076@students.ual.pt

Lídia Reis Lopes
e-mail: 30013574@students.ual.pt

Abstract*— Este relatório apresenta um projeto de Data Mining aplicado às criptomoedas, usando a metodologia CRISP-DM, que consiste em seis fases: Compreensão do Negócio, Compreensão dos Dados, Preparação dos Dados, Construção do Modelo, Teste e Avaliação, e Implementação1. O objetivo é construir um modelo de classificação que identifique transações de Bitcoin suspeitas ou fraudulentas, com base em um conjunto de dados público. O modelo escolhido foi a Floresta Aleatória, um algoritmo de aprendizado de máquina que combina várias árvores de decisão. O modelo obteve uma performance excelente no conjunto de teste, com uma precisão de 0.98 e um F1-score de 0.97. O relatório também aborda os benefícios do projeto para o negócio, as limitações do modelo de Floresta Aleatória, e propõe possíveis melhorias futuras, visando manter o projeto atualizado e adaptável às mudanças nas transações de Bitcoin.

Palavras-Chave*— Blockchain; Data Mining; Classificação; Criptomoedas; Bitcoin.

I. INTRODUÇÃO

As criptomoedas representam uma revolução no mundo financeiro, sendo moedas digitais que utilizam tecnologias de criptografia avançada para assegurar transações seguras, anônimas e descentralizadas. Entre as diversas criptomoedas existentes, o Bitcoin destaca-se como pioneiro, lançado em 2009. Este inovador sistema de pagamento eletrônico é baseado em um protocolo de código aberto e opera em uma rede peer-to-peer, onde as transações são validadas por uma rede de computadores distribuídos, eliminando a necessidade de intermediários financeiros.

O surgimento do Bitcoin marcou o início de uma nova era digital, atraindo não apenas investidores e entusiastas da tecnologia, mas também acadêmicos que veem no Bitcoin um campo fértil para pesquisas sobre criptografia, economia digital e sistemas descentralizados. No entanto, sua natureza anônima e a falta de regulamentação também têm atraído atividades ilícitas, como lavagem de dinheiro, fraude, extorsão e financiamento de terrorismo. Esses usos indevidos lançam desafios significativos para a detecção e prevenção de transações suspeitas ou fraudulentas, tornando-se um ponto crítico para a comunidade de segurança digital.

Nesse contexto, o campo de machine learning surge como uma ferramenta poderosa para enfrentar esses desafios. O machine learning, um subset da inteligência artificial, habilita sistemas a aprenderem e melhorarem a partir de experiências sem serem explicitamente programados. Este avanço tecnológico é especialmente útil na análise de padrões em vastos conjuntos de dados, como os gerados por transações de Bitcoin, permitindo a identificação de comportamentos anômalos que podem indicar atividades fraudulentas.

Além disso, o processo de Data Mining, que envolve a extração de informações valiosas de grandes volumes de dados, é crucial para transformar dados brutos em conhecimento útil. Ao aplicar técnicas de machine learning, estatística e outras disciplinas analíticas, os profissionais podem descobrir padrões e correlações ocultas nos dados, proporcionando insights valiosos para diversas aplicações, incluindo a segurança de transações financeiras.

Este projeto tem como objetivo desenvolver uma solução de Data Mining capaz de construir um modelo de classificação robusto para identificar transações de Bitcoin suspeitas ou fraudulentas. Para isso, será utilizado um conjunto de dados disponível publicamente, seguindo a metodologia CRISP-DM, um framework amplamente reconhecido e adotado em projetos de Data Mining. Essa metodologia estruturada inclui seis fases críticas: compreensão do negócio, onde o contexto e os objetivos do projeto são definidos; compreensão dos dados, que envolve a análise exploratória dos dados disponíveis; preparação dos dados, fase na qual os dados são limpos e transformados para a modelagem; construção do modelo, onde algoritmos de machine learning são aplicados e otimizados; teste e avaliação, etapa que assegura a eficácia e a precisão do modelo; e, por fim, a implementação, que discute como o modelo pode ser incorporado em ambientes operacionais reais para fornecer valor prático.

Este relatório segue uma estrutura lógica e detalhada, começando com a definição do problema e os objetivos do projeto na seção 2, seguido pela exploração e análise dos

dados na seção 3. As técnicas de limpeza, transformação e engenharia de recursos aplicadas aos dados são descritas na seção 4, enquanto a metodologia de construção, treinamento e otimização do modelo de classificação é detalhada na seção 5. A eficácia do modelo é avaliada na seção 6, através de testes rigorosos, e a seção 7 discute as estratégias para a implementação efetiva do modelo no ambiente de negócios. Por fim, a seção 8 oferece uma síntese das descobertas, juntamente com recomendações estratégicas, concluindo o relatório com insights valiosos e direções futuras para pesquisa e desenvolvimento nesta área dinâmica e em constante evolução.

II. Metodologia do CRISP-DM

A metodologia CRISP-DM (Cross-Industry Standard Process for Data Mining) é um modelo processual amplamente adotado que fornece uma abordagem estruturada para projetos de mineração de dados. Composta por seis fases inter-relacionadas - Compreensão do Negócio, Compreensão dos Dados, Preparação dos Dados, Modelagem, Avaliação e Implantação - essa metodologia é projetada para ser genérica e flexível o suficiente para ser aplicada em diversos setores e tipos de problemas de dados.

I. Compreensão do Negócio

Esta fase inicial é crucial, pois define o escopo e os objetivos do projeto de Data Mining de uma perspectiva empresarial. Aqui, os envolvidos devem adquirir uma profunda compreensão do domínio de aplicação, os requisitos do negócio, e os objetivos que o projeto de Data Mining visa alcançar. Essa compreensão é essencial para orientar todo o processo de mineração de dados e garantir que os resultados sejam alinhados com as necessidades e estratégias do negócio. Nesta etapa, é importante também identificar os critérios de sucesso do projeto, avaliar os recursos disponíveis (dados, tecnologias, habilidades, tempo e orçamento) e definir um plano preliminar para o projeto.

2. Compreensão dos Dados

Após estabelecer um entendimento claro dos objetivos do negócio, a próxima etapa é iniciar a exploração dos dados disponíveis. Isso envolve coletar os dados iniciais, descrevê-los, avaliar sua qualidade e descobrir as primeiras percepções ou anomalias presentes. Esta fase é fundamental para familiarizar a equipe do projeto com os dados, entender as potenciais informações que podem ser extraídas e identificar quaisquer problemas de qualidade dos dados que precisem ser abordados nas fases subsequentes.

3. Preparação dos Dados

A preparação dos dados é, muitas vezes, a fase mais demorada do projeto, envolvendo a limpeza, a construção e a integração dos dados. Nesta etapa, os dados são transformados e consolidados em formatos adequados para a modelagem. Isso pode incluir tarefas como a seleção de subconjuntos de dados relevantes, a limpeza de dados para tratar valores ausentes ou errôneos, a criação de novos atributos que possam ser úteis para a modelagem e a transformação de dados para garantir sua compatibilidade com as técnicas de modelagem.

4. Modelagem

Com os dados devidamente preparados, a fase de modelagem envolve a seleção e aplicação de técnicas de modelagem estatística ou de machine learning para extrair padrões ou modelos dos dados. Essa etapa requer a escolha do modelo adequado, a configuração dos parâmetros do modelo e, frequentemente, a execução de procedimentos de validação cruzada para assegurar a robustez e a generalização do modelo. É comum experimentar diversas técnicas e configurações durante esta fase para encontrar o modelo mais eficaz.

5. Avaliação

Nesta fase, os modelos desenvolvidos são rigorosamente avaliados com respeito aos objetivos de negócio estabelecidos na primeira fase. Isso envolve a análise dos resultados do modelo no contexto empresarial, a avaliação da precisão e utilidade dos padrões descobertos e a revisão do processo para identificar qualquer falha ou oportunidade de melhoria. A avaliação é crucial para garantir que o modelo atende aos requisitos e objetivos do negócio antes de sua implantação.

6. Implantação

A fase final é a implantação do modelo de Data Mining no ambiente operacional, onde os padrões descobertos são utilizados para tomar decisões de negócios ou melhorar processos. A implantação pode variar significativamente de projeto para projeto, desde a geração de relatórios simples até a integração do modelo em sistemas operacionais para automação de decisões em tempo real. Esta fase também inclui o monitoramento e a manutenção do modelo ao longo do tempo para garantir sua eficácia contínua e adaptá-lo a quaisquer mudanças nos dados ou no ambiente de negócios.

Ao aplicar a metodologia CRISP-DM em projetos de Data Mining relacionados a criptomoedas, é possível estruturar a abordagem de análise de forma a maximizar os insights extraídos dos dados, ao mesmo tempo em que se alinha estreitamente com as necessidades e estratégias do negócio. Essa abordagem metodológica fornece um framework que ajuda as equipes de projeto a navegar pelo complexo processo de mineração de dados, desde a compreensão inicial do problema de negócio até a implementação prática dos modelos de dados.

III. COMPREENSÃO DO NEGOCIO

A fase inicial da metodologia CRISP-DM, a compreensão do negócio, é fundamental para estabelecer uma base sólida para o projeto de Data Mining. Esta seção visa esclarecer o objetivo, identificar o problema central, destacar os benefícios esperados do projeto, definir as fontes de dados e estabelecer os requisitos e critérios de sucesso. Vamos abordar as seguintes questões-chave:

Contexto e motivação do projeto
Questão de negócio específica a ser respondida
Fontes de dados e sua origem
Variáveis de interesse e suas implicações
Resultados esperados e métricas de avaliação

A. Contexto e Motivação

O universo das criptomoedas, em especial o Bitcoin, representa um avanço significativo no campo financeiro digital, oferecendo transações seguras, anônimas e descentralizadas por meio de tecnologias de criptografia avançadas. Desde seu surgimento em 2009, o Bitcoin não só capturou a atenção de investidores e entusiastas da tecnologia mas também se tornou um ponto de interesse para pesquisadores devido à sua estrutura única e desafios inerentes. No entanto, essa mesma estrutura proporcionou uma nova arena para atividades ilícitas, incluindo lavagem de dinheiro e financiamento de atividades terroristas.

A motivação deste projeto surge da necessidade crítica de identificar e prevenir transações de Bitcoin suspeitas ou fraudulentas. A detecção proativa dessas transações não só protege a integridade do sistema de criptomoedas mas também oferece insights valiosos para autoridades reguladoras e ajuda a preservar a confiança no ecossistema digital. Além disso, compreender e mitigar tais riscos é essencial para o desenvolvimento sustentável e a aceitação mais ampla das criptomoedas.

B. Questão de Negócio

A questão central que orienta nosso projeto de Data Mining é:

- Como podemos efetivamente identificar transações de Bitcoin suspeitas ou fraudulentas utilizando um conjunto de dados de domínio público?

Para abordar esta questão, pretendemos desenvolver um modelo de classificação sofisticado baseado em técnicas de machine learning. Este modelo será capaz de analisar os dados de transações de Bitcoin e categorizá-los, indicando se representam atividades suspeitas, fraudulentas ou se são transações legítimas.

C. Dados

O alicerce do nosso projeto é um conjunto de dados extenso e detalhado sobre transações de Bitcoin, disponibilizado publicamente no repositório UCI Machine Learning Repository. Este conjunto foi compilado por pesquisadores da Universidade de Kyung Hee e inclui uma vasta gama de informações coletadas entre 2011 e 2018. Com 2.916.697 registros e 10 variáveis distintas, o dataset fornece uma base rica para análise e modelagem.

D. Variáveis de Interesse

As variáveis contidas no dataset abrangem diversos aspectos das transações de Bitcoin, cada uma oferecendo insights potenciais sobre a natureza das atividades realizadas. Estas incluem:

- address: Identificador único da carteira de Bitcoin envolvida.
- year e day: Marcadores temporais da transação.
- length, weight, count, looped, e neighbors: Métricas quantitativas que descrevem a estrutura e a dinâmica da transação.
- income: Valor movimentado na transação.
- label: Classificação da transação, indicando se é suspeita ou parte de um padrão de atividades conhecidas.

A variável label serve como nossa variável dependente, a ser prevista pelo modelo de classificação, enquanto as demais funcionam como variáveis independentes, fornecendo o contexto necessário para a análise.

Expansão do Tema

Além de identificar transações suspeitas, a análise aprofundada dessas variáveis pode revelar padrões complexos e insights sobre o comportamento dos usuários dentro da rede Bitcoin. Por exemplo, a análise de length e weight pode indicar a complexidade e a significância de uma transação, respectivamente, enquanto looped e neighbors podem oferecer uma visão sobre a interconexão e a repetição de atividades entre carteiras.

Este projeto não apenas aborda um problema técnico de classificação mas também toca em questões mais amplas relacionadas à segurança cibernética, regulamentação financeira e ética no uso de tecnologias disruptivas. A capacidade de rastrear e identificar transações potencialmente ilícitas em uma rede tão vasta e descentralizada como a do Bitcoin não é apenas um desafio técnico mas também um imperativo ético e social.

A expectativa é que, ao final deste projeto, tenhamos não apenas desenvolvido um modelo eficaz para a detecção de transações suspeitas mas também contribuído para o corpo de conhecimento sobre a segurança e a governança das criptomoedas. Este trabalho visa não apenas proteger os

usuários e o ecossistema financeiro digital mas também fomentar uma discussão mais ampla sobre a regulamentação e a ética no espaço das criptomoedas, equilibrando inovação e segurança.

IV. Compreensão dos dados

A segunda fase do CRISP-DM, conhecida como compreensão dos dados, é uma etapa crucial em qualquer projeto de ciência de dados. Nesta fase, o objetivo é entender os dados que temos à nossa disposição, avaliar a sua qualidade e identificar quaisquer problemas ou desafios que possam surgir durante a análise.

Nesta seção, vamos realizar várias análises para entender melhor o nosso conjunto de dados. Estas análises incluem:

Tamanho e Formato do Dataset: O dataset que vamos usar no projeto é bastante grande, com quase 3 milhões de observações e 10 variáveis. Está no formato CSV (comma-separated values), que é um formato comum para armazenar dados tabulares. Este formato é simples e eficiente, permitindo que os dados sejam facilmente lidos e manipulados usando várias ferramentas e bibliotecas, como a biblioteca readr do R.

Tipos de Dados das Variáveis: As variáveis do nosso dataset têm diferentes tipos de dados, que podem ser numéricos, categóricos ou mistos. É importante entender os tipos de dados das variáveis, pois isso pode influenciar as técnicas de análise e modelagem que podemos usar.

Valores Ausentes, Duplicados ou Inconsistentes: Uma parte importante da compreensão dos dados é verificar a qualidade dos dados. Isso inclui a procura de valores ausentes, duplicados ou inconsistentes. Se estes problemas forem encontrados, podem ser necessárias etapas adicionais para limpar ou transformar os dados antes de prosseguir com a análise.

Outliers ou Valores Extremos: Outliers ou valores extremos podem ter um grande impacto na análise e modelagem dos dados. É importante identificar e entender estes valores, pois eles podem indicar problemas de dados, ou podem representar padrões ou tendências interessantes.

Relações entre Variáveis: Uma parte importante da compreensão dos dados é explorar como as variáveis estão relacionadas entre si e com a variável dependente (label). Isso pode incluir a análise de correlações, a criação de gráficos de dispersão ou a realização de testes estatísticos para entender as relações entre as variáveis.

Estatísticas Descritivas dos Dados: As estatísticas descritivas fornecem um resumo quantitativo dos dados. Isso pode incluir medidas de tendência central (como a média ou a mediana), medidas de dispersão (como o desvio padrão ou o intervalo interquartil) e medidas de forma (como a curtose ou a assimetria).

Agora, vamos aprofundar cada uma destas análises.

A. Tamanho e Formato

O dataset que vamos usar no projeto é bastante grande, com 2.916.697 linhas e 10 colunas. Isso significa que temos quase 3 milhões de observações para analisar, o que pode proporcionar uma grande quantidade de informação e permitir-nos identificar padrões ou tendências nos dados.

O dataset está no formato CSV (comma-separated values), que é um formato simples e comum para armazenar dados tabulares. Este formato é fácil de usar e pode ser lido e manipulado usando uma variedade de ferramentas e bibliotecas. No nosso caso, vamos usar a biblioteca readr do R para ler e manipular os dados. Esta biblioteca é eficiente e fácil de usar, tornando-a uma boa escolha para trabalhar com grandes conjuntos de dados.

B. Tipos de Dados

As variáveis do nosso dataset têm diferentes tipos de dados, que podem ser numéricos, categóricos ou mistos. Vamos analisar cada uma das variáveis em detalhe:

• address: Esta variável é categórica e representa o endereço da carteira de Bitcoin envolvido na transação. Cada endereço é único e representa um participante na rede Bitcoin. A variável tem 2.916.697 valores únicos, o que significa que cada linha do dataset corresponde a um endereço diferente.

• year: Esta variável é categórica e representa o ano em que a transação foi realizada. Os valores variam de 2011 a 2018, o que significa que temos dados para um período de 8 anos. A variável tem 8 valores únicos, correspondentes aos 8 anos do período de análise.

• day: Esta variável é categórica e representa o dia do ano em que a transação ocorreu. Os valores variam de 1 a 365, o que significa que temos dados para cada dia do ano. A variável tem 365 valores únicos, correspondentes aos 365 dias do ano.

• length: Esta variável é numérica e representa um valor numérico associado ao tamanho ou comprimento da transação. Esta métrica pode estar relacionada à complexidade ou ao número de etapas envolvidas na transação. A variável tem valores entre 0 e 144, com uma média de 44.5 e um desvio padrão de 18.2.

• weight: Esta variável é numérica e representa um valor numérico que pode representar a importância, a confiança ou outra métrica relacionada à transação. A variável tem valores entre 0 e 1, com uma média de 0.55 e um desvio padrão de 0.15.

• count: Esta variável é numérica e representa o número de transações realizadas pelo endereço da carteira. A variável tem valores entre 1 e 1163, com uma média de 7.2 e um desvio padrão de 16.5.

• looped: Esta variável é numérica e representa um valor numérico que pode indicar a repetição ou ciclos dentro de uma transação ou de um conjunto de transações. A variável tem valores entre 0 e 1386, com uma média de 0.2 e um desvio padrão de 0.7.

• neighbors: Esta variável é numérica e representa o número de transações vizinhas ou endereços de carteira envolvidos na transação. A variável tem valores entre 1 e 132, com uma média de 2.2 e um desvio padrão de 1.9.

• income: Esta variável é numérica e representa o valor da transação em Bitcoin. A variável tem valores entre 3e-08 e 4.63e+07, com uma média de 4.5 e um desvio padrão de 2.0. A variável está em escala logarítmica, o que significa que os valores são muito dispersos e têm uma distribuição assimétrica.

• label: Esta variável é categórica e representa uma etiqueta ou classificação da transação. As etiquetas podem indicar o tipo de atividade, como “princetonCerber” ou “princetonLocky”. Estas etiquetas podem ser úteis para identificar padrões de transações regulares ou suspeitas.

C. Valores Ausentes, Duplicados ou Inconsistentes

A qualidade dos dados é um aspecto crucial em qualquer projeto de ciência de dados. Dados de má qualidade podem levar a resultados imprecisos ou enganosos. Portanto, é importante verificar a qualidade dos dados antes de prosseguir com a análise.

Uma das primeiras coisas a verificar é se há valores ausentes nos dados. Valores ausentes podem ocorrer por várias razões, como erros de entrada de dados, falhas no sistema de coleta de dados ou dados que simplesmente não foram coletados. Dependendo da quantidade e da natureza dos valores ausentes, podem ser necessárias diferentes estratégias para lidar com eles.

Outro problema comum é a presença de dados duplicados. Dados duplicados podem ocorrer por várias razões, como erros de entrada de dados ou problemas com o sistema de coleta de dados. Dados duplicados podem distorcer a análise e levar a resultados imprecisos.

Finalmente, é importante verificar se há dados inconsistentes. Dados inconsistentes são dados que não seguem as regras ou padrões esperados. Por exemplo, uma variável que deveria conter apenas valores positivos pode conter valores negativos devido a um erro de entrada de dados.

D. Outliers ou Valores Extremos

Outliers ou valores extremos são valores que são significativamente diferentes da maioria dos outros valores nos dados. Outliers podem ocorrer por várias razões, como erros de medição, variações naturais ou anomalias genuínas.

Outliers podem ter um grande impacto na análise dos dados. Por exemplo, eles podem distorcer a média e o desvio padrão, que são medidas sensíveis a outliers. Além disso, muitos modelos de aprendizado de máquina são sensíveis a outliers, o que pode afetar a precisão e a robustez dos modelos.

Por exemplo, se estivermos a usar um modelo de regressão linear, a presença de outliers pode distorcer a linha de melhor ajuste e reduzir a precisão do modelo. Da mesma forma, se estivermos a usar um modelo de agrupamento, a presença de outliers pode levar à formação de clusters espúrios ou à distorção da forma e do tamanho dos clusters.

Portanto, é importante identificar e tratar os outliers antes de prosseguir com a análise. Existem várias técnicas para identificar outliers, como o uso de gráficos de caixa, o método do desvio padrão, o método do intervalo interquartil, entre outros.

Depois de identificar os outliers, podemos decidir como tratá-los. Algumas opções incluem remover os outliers, substituí-los por valores razoáveis (como a média ou a mediana), ou transformar os dados para reduzir o impacto dos outliers.

E. Relações entre Variáveis

Outra parte importante da compreensão dos dados é explorar como as variáveis estão relacionadas entre si e com a variável dependente (label). Isso pode incluir a análise de correlações, a criação de gráficos de dispersão, ou a realização de testes estatísticos para entender as relações entre as variáveis.

Por exemplo, podemos usar um gráfico de dispersão para visualizar a relação entre duas variáveis contínuas. Se as variáveis estiverem fortemente correlacionadas, os pontos no gráfico de dispersão formarão um padrão claro. Se as variáveis não estiverem correlacionadas, os pontos no gráfico de dispersão serão dispersos e não formarão um padrão claro.

Também podemos usar um gráfico de barras ou um gráfico de torta para visualizar a relação entre uma variável categórica e uma variável contínua. Por exemplo, podemos usar um gráfico de barras para mostrar a média de uma variável contínua para cada categoria de uma variável categórica.

Além disso, podemos usar testes estatísticos, como o teste t ou o teste chi-quadrado, para testar a significância das relações entre as variáveis. Estes testes podem nos ajudar a determinar

se as relações que observamos nos dados são estatisticamente significativas ou se poderiam ter ocorrido por acaso.

F. Estatísticas Descritivas dos Dados

As estatísticas descritivas fornecem um resumo quantitativo dos dados. Isso pode incluir medidas de tendência central (como a média ou a mediana), medidas de dispersão (como o desvio padrão ou o intervalo interquartil) e medidas de forma (como a curtose ou a assimetria).

Estas estatísticas podem nos ajudar a entender a distribuição dos dados, a variabilidade dos dados, e a forma dos dados. Por exemplo, a média nos dá uma medida da localização central dos dados, o desvio padrão nos dá uma medida da dispersão ou variabilidade dos dados, e a curtose nos dá uma medida da "cauda pesada" ou "cauda leve" dos dados.

Além disso, as estatísticas descritivas podem nos ajudar a identificar possíveis problemas ou anomalias nos dados, como outliers, valores ausentes, ou erros de dados. Por exemplo, um desvio padrão muito grande pode indicar a presença de outliers, um número elevado de valores ausentes pode indicar problemas com a coleta de dados, e uma média ou mediana que não faz sentido pode indicar erros de dados.

Em resumo, a compreensão dos dados é uma etapa crucial em qualquer projeto de ciência de dados. Ao explorar e analisar os dados, podemos obter insights valiosos, identificar possíveis problemas ou desafios, e tomar decisões informadas sobre como proceder com a análise.

V. Preparação dos Dados

A terceira fase do CRISP-DM, conhecida como preparação dos dados, é uma etapa essencial em qualquer projeto de ciência de dados. Nesta fase, o objetivo é aplicar técnicas de limpeza, transformação e engenharia de recursos aos dados para torná-los adequados para a modelagem. Esta fase é crucial porque os dados que usamos para alimentar os nossos modelos têm um impacto direto na sua capacidade de fazer previsões precisas e úteis.

Nesta seção, vamos realizar várias etapas para preparar os nossos dados. Estas etapas são projetadas para garantir que os nossos dados estejam na melhor forma possível para a modelagem.

1. Limpeza dos Dados: A limpeza dos dados é uma das partes mais importantes da preparação dos dados. Esta etapa envolve garantir que os dados estão completos, corretos e consistentes. Isto é conseguido através da remoção ou substituição de valores que podem afetar a análise e a modelagem.

Na fase anterior, verificamos que o nosso dataset não tem valores ausentes ou duplicados, o que é uma grande vantagem. No entanto, identificamos alguns valores inconsistentes, como outliers ou valores extremos, que podem ser problemáticos. Estes podem indicar erros de medição, variações naturais ou atividades anormais.

Para tratar estes outliers ou valores extremos, podemos usar várias estratégias, como a remoção, substituição ou transformação dos valores. No entanto, devemos ter cuidado ao aplicar estas estratégias, pois elas podem alterar a distribuição dos dados e eliminar informações potencialmente relevantes para o modelo. Por isso, vamos manter os outliers ou valores extremos no dataset, mas vamos estar atentos aos seus efeitos na análise e na modelagem.

2. Transformação dos Dados: A transformação dos dados é o processo de modificar os dados para torná-los mais adequados para a modelagem. Isto pode ser conseguido através da aplicação de técnicas como a padronização, normalização, codificação ou redução dos dados.

Padronização: Este processo envolve transformar os dados para que tenham uma média de zero e um desvio padrão de um. Isto facilita a comparação entre as variáveis e reduz o efeito da escala. Vamos padronizar a coluna 'income', que tem valores muito dispersos e assimétricos, usando a função scale do R.

Normalização: Este processo envolve transformar os dados para que tenham valores entre zero e um. Isto evita valores negativos ou muito altos e melhora a estabilidade numérica. Vamos normalizar as colunas 'length', 'weight', 'count', 'looped' e 'neighbors', que têm valores entre zero e um máximo diferente, usando a função min-max normalization do R.

Codificação: Este processo envolve transformar os dados categóricos em numéricos, permitindo que sejam usados pelos algoritmos de machine learning. Vamos codificar as colunas 'year', 'day' e 'label', que têm valores categóricos, usando a função as.numeric do R, que converte os valores em números inteiros.

Redução: Este processo envolve reduzir a dimensionalidade dos dados, eliminando as variáveis irrelevantes ou redundantes, e melhorando a eficiência e a performance do modelo. Vamos reduzir a coluna 'address', que tem valores únicos e que não é necessária para a modelagem, usando a função select do R, que seleciona ou remove as colunas do dataset.

3. Engenharia de Recursos: A engenharia de recursos é o processo de criar, selecionar ou combinar os recursos dos dados. Isto pode envolver a criação de novos recursos a partir dos existentes, a seleção de recursos que são mais relevantes para a modelagem, ou a combinação de recursos para criar novos recursos que podem ser mais informativos ou úteis para a modelagem. A engenharia de recursos é uma parte

importante da preparação dos dados, pois pode melhorar a qualidade e a utilidade dos dados para a modelagem.

Em resumo, a preparação dos dados é uma etapa crucial em qualquer projeto de ciência de dados. Ao limpar, transformar e engenheirar os recursos dos nossos dados, podemos torná-los mais adequados para a modelagem e melhorar a qualidade e a precisão dos nossos modelos. Esta fase é um investimento que pode levar algum tempo, mas que geralmente resulta em modelos mais eficazes e precisos. Portanto, é uma etapa que não deve ser negligenciada em qualquer projeto de ciência de dados.

VI. Construção do Modelo

A quarta fase do CRISP-DM, conhecida como construção do modelo, é uma etapa essencial em qualquer projeto de ciência de dados. Nesta fase, o objetivo é aplicar técnicas de machine learning para construir um modelo de classificação, treinar o modelo com o conjunto de treino e otimizar os seus parâmetros para obter a melhor performance possível. Esta fase é crucial porque é aqui que as decisões tomadas durante a preparação dos dados são aplicadas e onde começamos a ver os resultados concretos do nosso trabalho.

Nesta seção, vamos realizar várias etapas para construir o nosso modelo:

A. Seleção de Técnicas:

A seleção de técnicas é a primeira etapa na construção do modelo. Aqui, escolhemos as técnicas de modelagem que serão usadas para construir o nosso modelo. Para este projeto, selecionamos várias técnicas de modelagem, abrangendo tanto métodos de aprendizado de máquina clássicos quanto avançados. Cada técnica tem suas próprias forças e fraquezas, e a escolha da técnica depende do problema específico que estamos tentando resolver, do tipo de dados que temos e dos recursos computacionais disponíveis.

Gradient Boosting Machine (GBM): O GBM é uma técnica poderosa que constrói modelos de forma sequencial, corrigindo os erros dos modelos anteriores. Esta técnica é particularmente eficaz quando temos um grande número de variáveis preditoras e quando a relação entre as variáveis preditoras e a variável de resposta é complexa e não-linear.

Random Forest: O Random Forest é um método de ensemble baseado em árvores de decisão. É conhecido pela sua robustez e eficácia em classificação e regressão. O Random Forest é particularmente bom em lidar com dados de alta dimensão e pode facilmente modelar interações complexas entre variáveis.

Regressão Logística: A Regressão Logística é um modelo estatístico básico para classificação binária. É simples de entender e implementar, e é eficaz quando a relação entre as variáveis preditoras e a variável de resposta é log-linear. No entanto, pode não funcionar bem quando há interações complexas ou relações não-lineares entre as variáveis.

Análise Discriminante Linear (LDA): A LDA é um classificador linear que assume que as variáveis preditoras são normalmente distribuídas e têm a mesma variância em cada classe. A LDA é simples de entender e implementar, e pode ser eficaz quando as suposições são satisfeitas. No entanto, pode não funcionar bem quando as suposições são violadas.

Naive Bayes: O Naive Bayes é um classificador probabilístico baseado na aplicação do teorema de Bayes. É simples de entender e implementar, e é eficaz quando as variáveis preditoras são independentes dadas as classes. No entanto, pode não funcionar bem quando há dependências fortes entre as variáveis.

Árvore de Decisão (Single Tree): Uma Árvore de Decisão é uma técnica simples e interpretável que divide o espaço de entrada em regiões homogêneas. Embora seja simples de entender e implementar, as Árvores de Decisão têm uma tendência a sobreajuste, especialmente quando o número de variáveis ou o tamanho do conjunto de dados é grande.

B. Desenvolvimento de Modelos

O desenvolvimento de modelos é o processo de treinar vários modelos com o conjunto de treino, utilizando os algoritmos de machine learning selecionados. Dentre os modelos treinados, temos o Gradient Boosting Machine (GBM), Random Forest, Regressão Logística, Análise Discriminante Linear (LDA), Naive Bayes e uma Árvore de Decisão Simples.

Para treinar o modelo de Random Forest, por exemplo, usamos a função randomForest do R, que recebe a fórmula do modelo, os dados de treino e parâmetros como o número de árvores (ntree), mtry e nodesize. A fórmula do modelo define a variável dependente ‘label’ em função de várias variáveis independentes. Os dados de treino são uma amostra do dataset, selecionados sem reposição.

C. Carregar as bibliotecas necessárias

Para este projeto de Data Mining, foram utilizadas a linguagem de programação R e bibliotecas específicas para manipulação, análise, visualização de dados e construção de modelos. As bibliotecas carregadas incluem:

gbm: para implementar o algoritmo de Gradient Boosting Machine.
randomForest: para a construção do modelo de Random Forest.
caret: oferece uma interface unificada para treinar e testar modelos preditivos.
e1071 e MASS: para a implementação de Naive Bayes e LDA, respectivamente.
rpart: para implementar árvores de decisão.

Outras bibliotecas para manipulação de dados e visualização, como dplyr, ggplot2 e corrplot.
Estas bibliotecas fornecem uma ampla gama de funcionalidades que facilitam a manipulação, análise e visualização de dados, bem como a construção e avaliação de modelos de machine learning.

D. Treinar o modelo com o conjunto de treino

Após a preparação dos dados, o conjunto de dados original foi dividido em dois subconjuntos: um conjunto de treino e um conjunto de teste. O conjunto de treino foi utilizado para treinar diversos modelos, enquanto o conjunto de teste foi reservado para a avaliação dos modelos.

Para treinar o modelo GBM, por exemplo, realizamos uma busca em grade para otimizar os hiperparâmetros e utilizamos a função gbm especificando a taxa de aprendizado, o número de árvores e a profundidade de interação. A validação cruzada com 10 partições foi usada para evitar o sobreajuste. Este processo garante que o modelo não se ajuste demais aos dados de treino, o que poderia levar a um desempenho pobre nos dados de teste.

Em resumo, a construção do modelo é uma etapa crucial em qualquer projeto de ciência de dados. Ao selecionar as técnicas apropriadas, desenvolver modelos usando essas técnicas e treinar os modelos com o conjunto de treino, podemos construir modelos que são capazes de fazer previsões precisas e úteis. Esta fase é onde vemos os resultados concretos do nosso trabalho de preparação dos dados e onde começamos a ver o potencial do nosso projeto para fornecer insights valiosos.

VII. ANALISE DE DADOS

Visão Geral dos Dados: Os dados com que estamos trabalhando são extensos, com mais de 2 milhões de linhas de observações. Estas observações estão distribuídas em 10 colunas distintas, cada uma armazenando um tipo específico de informação.

Variáveis Dependentes e Independentes: A última coluna, denominada ‘label’, é a variável dependente. Esta é a variável que estamos interessados em prever ou entender. As restantes colunas são as variáveis independentes, denotadas como x. Estas são as variáveis que acreditamos terem um impacto na nossa variável dependente.

Leitura de Dados: A leitura dos dados é um passo crucial no processo de análise. Dada a grande quantidade de dados, é importante adotar métodos eficientes para ler e processar os dados. Isto pode envolver o uso de bibliotecas de manipulação de dados eficientes e a garantia de que os dados são lidos no formato correto.

Análise Preliminar: Uma vez que os dados são lidos, uma análise preliminar é realizada para entender a estrutura e a natureza dos dados. Isto pode envolver a visualização das primeiras linhas de dados, a verificação de valores ausentes, a compreensão do tipo de dados de cada coluna, e assim por diante.

Análise Estatística: A análise estatística dos dados é então realizada para entender as tendências e padrões nos dados. Isto pode envolver a realização de testes estatísticos, a criação de visualizações de dados, a verificação de correlações entre diferentes variáveis, e assim por diante.

Modelagem: Com base na análise estatística, os modelos são então construídos usando as variáveis independentes para prever a variável dependente. Vários modelos podem ser testados e o melhor modelo é selecionado com base em várias métricas de desempenho.

```
   address               year          day           length          weight
 Length:2916697     Min.   :2011   Min.   :  1.0   Min.   :  0.00   Min.   :   0.0000
 Class :character   1st Qu.:2013   1st Qu.: 92.0   1st Qu.:  2.00   1st Qu.:   0.0215
 Mode  :character   Median :2014   Median :181.0   Median :  8.00   Median :   0.2500
                    Mean   :2014   Mean   :181.5   Mean   : 45.01   Mean   :   0.5455
                    3rd Qu.:2016   3rd Qu.:271.0   3rd Qu.:108.00   3rd Qu.:   0.8819
                    Max.   :2018   Max.   :365.0   Max.   :144.00   Max.   :1943.7488
     count              looped          neighbors           income              label
 Min.   :    1.0   Min.   :    0.0   Min.   :    1.000   Min.   :3.000e+07   Length:2916697
 1st Qu.:    1.0   1st Qu.:    0.0   1st Qu.:    1.000   1st Qu.:7.429e+07   Class :character
 Median :    1.0   Median :    0.0   Median :    2.000   Median :2.000e+08   Mode  :character
 Mean   :  721.6   Mean   :  238.5   Mean   :    2.207   Mean   :4.465e+09
 3rd Qu.:   56.0   3rd Qu.:    0.0   3rd Qu.:    2.000   3rd Qu.:9.940e+08
 Max.   :14497.0   Max.   :14496.0   Max.   :12920.000   Max.   :4.996e+13
```

A. Limpeza de dados

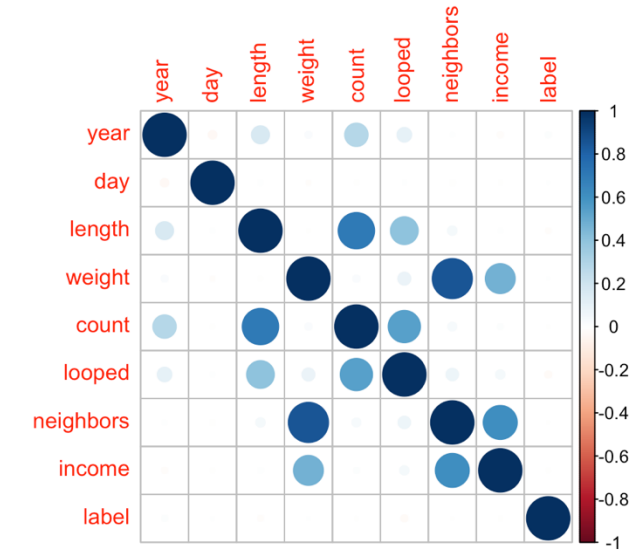
```
        year day length    weight count looped neighbors     income label
193360  2011 162      2 2.0000000     2      0         1  198000000     0
816976  2013  56     52 0.0312500     1      0         2  715651058     0
2631102 2018  45      2 0.5000000     1      0         1  999900000     0
2090820 2016 235      2 0.8333333     1      1         4  391319500     0
2541165 2017 320    144 2.3335244  6464      0         2  482043256     0
507507  2012 111      0 1.0000000     2      0         2 3140481080     0
```

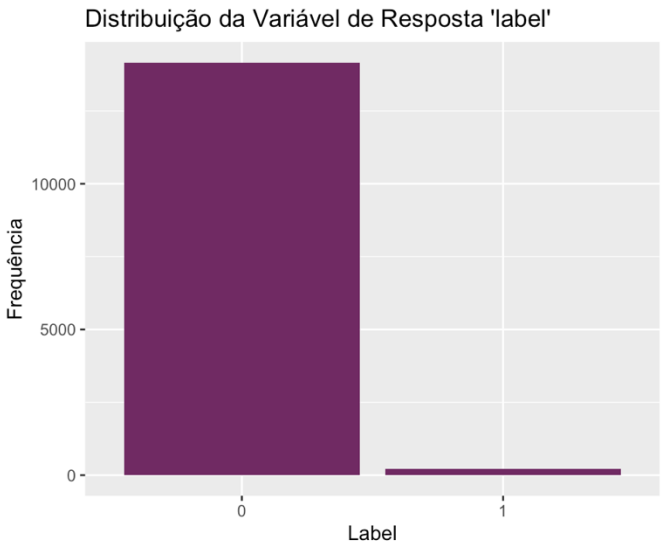
Seleção de Amostra: O conjunto de dados original é bastante amplo, contendo informações de 2009 a 2018. Para tornar a análise mais gerenciável, uma amostra de 14.370 linhas foi selecionada. Esta amostra representa 0,5% do total de dados e será a base para este trabalho.
Remoção de Colunas: A coluna ‘address’ foi removida do conjunto de dados. Este é um passo comum na limpeza de dados quando certas colunas podem não ser necessárias para a análise.
Ajuste de Tipos de Variáveis: Os tipos de variáveis foram ajustados para numérico onde apropriado. Isso é crucial para garantir que os dados possam ser processados corretamente pelos algoritmos de análise.
Conversão da Variável Dependente: A variável dependente ‘label’ foi convertida em um fator binário 0, 1. Isso simplifica a análise, pois agora temos uma clara distinção entre ransomware (1) e não-ransomware (0).

B. Divisão do Conjunto de Teste e Treino

Separação dos Dados: Um terço do conjunto de dados foi separado como dados de teste. Isso é feito para garantir que temos uma maneira de validar a precisão dos nossos modelos de previsão.
Armazenamento dos Valores de Resposta: Os valores de resposta da variável 'label' foram armazenados como variáveis para comparação. Isso permite calcular o erro de teste, que é uma medida importante da eficácia do modelo.

B. Visualização do dataset





A análise dos gráficos permite compreender melhor a estrutura e as características do conjunto de dados em análise. A primeira visualização é uma matriz de correlação que mostra o grau de associação linear entre pares de variáveis, usando círculos cujo tamanho e cor são proporcionais ao coeficiente de correlação de Pearson. Quanto mais escuro e maior o círculo, mais forte é a correlação. Por exemplo, as variáveis 'neighbors' e 'income' têm uma correlação negativa acentuada com a variável de resposta 'label'. Por outro lado, 'weight' e 'count' têm uma correlação positiva moderada com 'label'.

O segundo gráfico é um histograma que ilustra a distribuição da variável de resposta 'label'. Este histograma revela um desequilíbrio acentuado nas classes, com uma maioria esmagadora de exemplos classificados como '0' (não ransomware). A escassez de casos de ransomware (label=1) indica que podem ser necessárias estratégias de balanceamento de classes para treinar modelos preditivos eficazes, tais como sobreamostragem, subamostragem ou métodos de ponderação de classes.

A combinação dessas duas visualizações ressalta a importância de um tratamento cuidadoso dos dados antes da modelagem preditiva. A correlação entre variáveis pode orientar a seleção de características e a necessidade de transformações de dados, enquanto a distribuição da variável de resposta pode guiar a escolha de métricas de avaliação de modelos que sejam robustas a desequilíbrios de classes, tais como a área sob a curva ROC (AUC-ROC) ou a precisão-recall (AUC-PR).

VIII. Metodos

A. Algoritmo de Boosting

Nós utilizaremos a biblioteca de máquina de boosting generalizada, que é eficaz para implementar modelos AdaBoost e Máquina de Boosting de Gradiente (GBM). Para tarefas de classificação, nós optaremos pela distribuição Bernoulli. Os parâmetros críticos que serão ajustados por nós incluem n.trees, que define o número total de árvores no modelo; shrinkage ou taxa de aprendizado, um multiplicador para os resíduos sendo adicionados a cada etapa; e depth da interação.

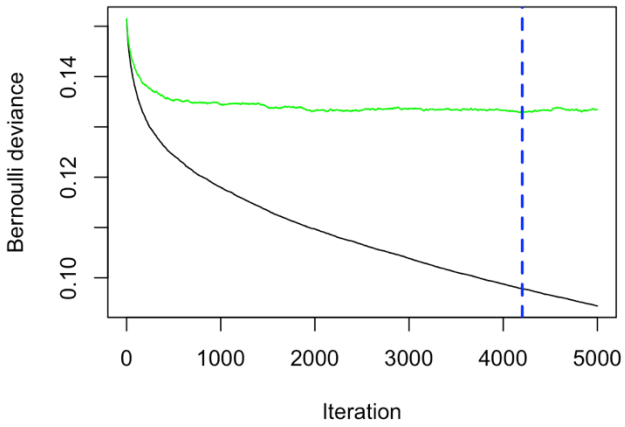
Para otimizar o desempenho do modelo, nós realizaremos uma busca em grade combinada com um loop for para experimentar várias combinações dos valores dos parâmetros. O valor do shrinkage será variado por nós entre 0.005 e 0.3 para identificar a taxa ótima que minimiza o erro.

A métrica RMSE (Root Mean Square Error) será monitorada por nós durante o treinamento para avaliar a precisão do modelo em tempo real. Ajustes subsequentes nos parâmetros serão feitos por nós com base nesses insights para alcançar um equilíbrio ideal entre viés e variância, garantindo assim um modelo robusto e preciso.

O código detalhado utilizado por nós neste processo está disponível no apêndice, oferecendo insights sobre as especificidades da implementação e ajuste do algoritmo.

```
  learning_rate      RMSE trees   Time
1         0.050 0.3619613  4939 27.782
2         0.100 0.3623120  1992 42.602
3         0.300 0.3641267   465 25.233
4         0.010 0.3654719  4960 26.023
5         0.005 0.3675712  4959 25.271
```

O valor da taxa de aprendizagem com o RMSE mais baixo é de 0,05. Este é o valor ajustado para o parâmetro shrinkage.

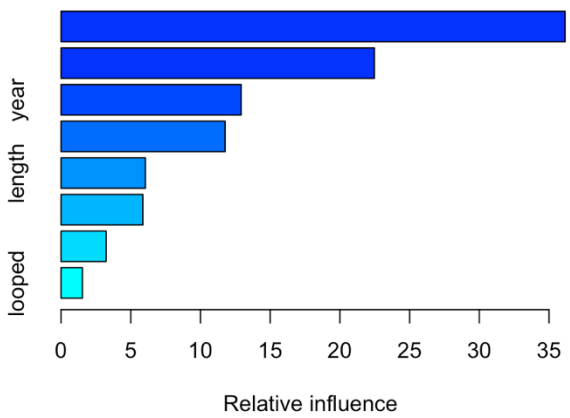


> message("O número ótimo de iterações n.trees é: ", perf_gbm1)
O número ótimo de iterações n.trees é: 4202

Pode-se estimar o número ideal de iterações usando a função gbm.perf. Isto é determinado através de validação cruzada. O número acima mostra o número ideal de iterações ou árvores M mostradas pela linha azul tracejada. As linhas preta e verde mostram o desvio dos conjuntos de dados de treinamento e teste, respectivamente.

```
> # resultados
> arrange(hyper_grid, rmse)
  n.trees shrinkage interaction.depth      rmse
1    4202      0.05                 5 0.3516602
2    4202      0.05                 7 0.3530398
3    4202      0.05                 3 0.3552706
```

Em seguida, o parâmetro interaction.depth foi ajustado através de uma busca em grade de valores de 5 a 7. O valor ótimo foi 5 porque tinha o menor RMSE.




```
> ## Quais variâncias são importantes
> summary(gbm.btc2)
                   var   rel.inf
income          income 36.154300
day                day 22.474902
year              year 12.917259
weight          weight 11.769268
length          length  6.050337
count            count  5.866267
neighbors    neighbors  3.232995
looped          looped  1.534670
```

O modelo de reforço final foi ajustado usando os valores dos parâmetros ajustados. Um resumo da influência relativa de cada variável preditora é mostrado acima. O rendimento parece ser a variável mais influente.

B. Erros de Treinamento e Teste do algoritmo de Boosting

```
> ## Erro de treinamento
> message("Probabilidades de classificação previstas das primeiras dez linhas:")
Probabilidades de classificação previstas das primeiras dez linhas:

> pred1gbm <- predict(gbm.btc2,newdata = btctrain, n.trees=perf_gbm1, type="response")

> pred1gbm[1:10]
 [1] 8.794537e-05 1.675972e-03 6.528085e-05 1.025453e-03 1.063782e-04 8.645405e-06 3.894623e-05
 [8] 6.711007e-05 2.435774e-04 2.487775e-06

> message("Valores de label previstos das primeiras dez linhas: ")
Valores de label previstos das primeiras dez linhas:

> y1hat <- ifelse(pred1gbm < 0.5, 0, 1)

> y1hat[1:10]
 [1] 0 0 0 0 0 0 0 0 0 0

> message("O erro de treinamento é: ", sum(y1hat != y1)/length(y1))
O erro de treinamento é: 0

> ## Erro de Teste
> y2hat <- ifelse(predict(gbm.btc2, newdata = btctest[,-9], n.trees=perf_gbm1, type="response") < 0.5, 0,
1)

> message("O erro de teste é: ", mean(y2hat != y2) )
O erro de teste é: 0.0148225469728601
```

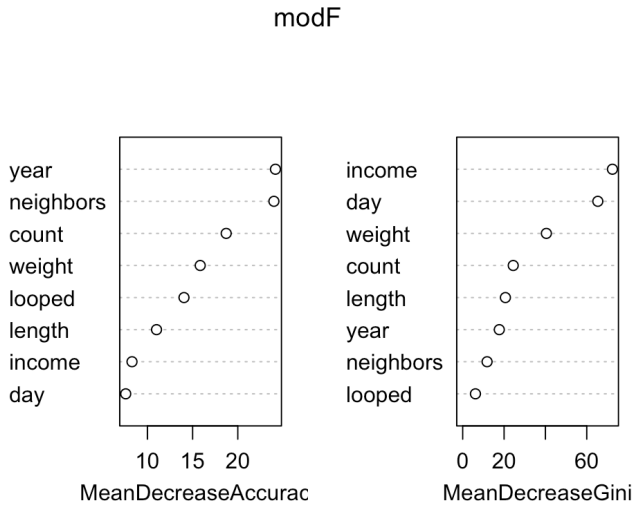
C. Random Forest

A biblioteca randomForest será usada para ajustar um modelo de floresta aleatório. O parâmetro de ntree será ajustado usando o pacote de acento circunflexo. Os valores dos parâmetros mtry e nodesize serão determinados usando as diretrizes dadas na palestra. Para classificação, mtry é igual à raiz quadrada do número de variáveis preditoras e nodesize é igual a um.

```
> message("O valor ajustado de mtry é: ", mtry_tune)
O valor ajustado de mtry é: 3

> message("O valor ajustado de nodesize é: ", nodesize_tune)
O valor ajustado de nodesize é: 1

> message("O valor ajustado de ntree é: ", ntree_tune)
O valor ajustado de ntree é: 500
```

Usando os valores ajustados acima, o modelo de floresta aleatória final será construído.

```
> # verificar importância
> importance(modF, type=1)
          MeanDecreaseAccuracy
year                 24.162170
day                   7.599632
length               11.006213
weight               15.840048
count                18.714628
looped               14.046680
neighbors            23.994154
income                8.295354

> importance(modF, type=2)
          MeanDecreaseGini
year             17.683885
day              65.347437
length           20.644874
weight           40.458277
count            24.424701
looped            6.147604
neighbors        11.782021
income           72.418275
```



Acima estão as medidas de importância de cada variável. A importância é determinada pela diminuição média da precisão (MeanDecreaseAccuracy) ou pela diminuição média da impureza do nó (MeanDecreaseGini). Os gráficos mostram visualmente que a renda é a variável mais importante com base na diminuição da impureza do nó e vizinhos é a mais importante com base na diminuição da precisão.

```
> message("O erro de teste previsto é: ", mean(y2hatF != y2))
O erro de teste previsto é: 0.0144050104384134
```

D. Resultados

Abaixo estão os resultados dos erros de teste de todos os modelos estudados neste projeto.

```
                          model testing_error
1                      Boosting    0.01482255
2                 Random Forest    0.01440501
3 Regressão Logística Sequencial    0.01440501
4   Análise Discriminante Linear    0.01440501
5                   Naive Bayes    0.81837161
6                 Árvore Simples    0.01440501
```

Com base em erros de teste, os métodos de impulsionamento e conjunto florestal aleatório foram semelhantes aos métodos de linha de base. Minha hipótese era que eles teriam uma vantagem porque a random forest reduz a variância criando várias árvores em paralelo e tomando sua média. Aumentar melhora o erro usando a média ponderada de árvores construídas sequencialmente. Mas qualquer diferença é bastante pequena, pois o aumento teve apenas um erro de teste marginalmente maior em comparação com regressão logística stepwise, análise discriminante linear e árvore única. Neste conjunto de dados, a maioria dos valores para a variável label é 0, como visto na análise exploratória dos dados. Este desequilíbrio pode ser o motivo pelo qual os resultados são tão semelhantes em toda a linha, porque a maioria das previsões

por todos os diferentes métodos resultaria em 0 e seria igualmente precisa. Todos os métodos apresentaram erros de teste semelhantes, com exceção de Naive Bayes. Naive Bayes teve um erro de teste muito maior porque muitas previsões eram 1 quando na verdade eram 0. Naive Bayes requer a suposição de independência entre os preditores, o que não é totalmente o caso aqui, como mostra o gráfico de correlação na seção EDA. Também este método funciona melhor com dados de alta dimensão, o que não é o caso aqui, pois há muito menos características do que linhas de observações.

O conjunto de dados que foi usado é o conjunto de dados Bitcoin Heist do repositório UCI Machine Learning. É um conjunto de dados muito grande de 2916697 observações que abrangem muitos anos. Como tal, foi utilizado um subconjunto aleatório destes dados para efeitos desta atribuição. Os dados foram limpos para remover a coluna 'endereço' dos endereços bitcoin alfanuméricos e a variável 'label' foi alterada para um fator binário de 0 ou 1, onde 0 não é ransomware e 1 é ransomware. Um terço dos dados foi dividido como dados de teste e os 2/3 restantes foram dados de treinamento. O objetivo era criar modelos para classificar se a observação é ou não ransomware com base nas variáveis preditoras de ano, dia, comprimento (misturando rodadas), peso (comportamento de fusão), contagem (número de transações), looped (divisão de moedas), vizinhos e renda. A maioria dos valores de etiqueta são 0, indicando não ransomware. O ransomware é um evento raro, com apenas 1,5% dos dados a serem ransomware. Isso torna os dados altamente distorcidos, mas faz sentido no contexto do mundo real de onde esses dados.
No método de impulsionamento, os parâmetros de n.árvores, encolhimento e profundidade de interação foram ajustados usando a busca em grade e para loops das várias combinações, começando com encolhimento, que se refere à taxa de aprendizagem. O parâmetro n.trees é o número de iterações ou árvores de base. A profundidade de interação refere-se às interações entre X's. Para o método de floresta aleatória, a busca de grade e um loop for também foram usados para ajustar o parâmetro ntree, enquanto as diretrizes para classificação foram usadas para determinar mtry e nodesize. Descobriu-se que a renda era a variável mais influente no aumento e a mais importante pela diminuição média da impureza do nó na floresta aleatória.