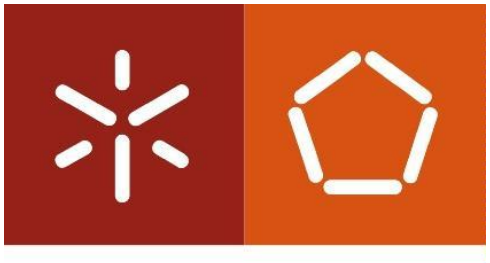

Aprendizagem e Decisão Inteligentes

Relatório Trabalho Prático
Grupo 26

LEI - 3º Ano - 2º Semestre
Ano Letivo 2024/2025


Tiago Matos Guedes
A97369


André Vaz
A93221


Tiago Carneiro
A93207


Luís Ferreira
A98286

Braga,
4 de junho de 2025

Conteúdo

Lista de Figuras

1 Introdução

Este relatório descreve o trabalho realizado no âmbito da unidade curricular de Aprendizagem e Decisão Inteligentes, focando-se em desenvolvimento de modelos de machine learning. O objetivo deste projeto é a análise, tratamento e modelação de dados utilizando a ferramente KNIME.
O projeto está dividido em duas tarefas principais:

- Tarefa Dataset Grupo : Escolha, exploração e posterior modelação de um dataset livremente selecionado a partir das fontes fornecidas pela equipa docente.
- Tarefa Dataset Atribuído : Análise e desenvolvimento de modelos de machine learning com um dataset disponibilizado pela equipa docente.

A realização deste trabalho permite consolidar os conceitos abordados nas aulas, como a compreensão de algoritmos de machine learning e técnicas de extração de conhecimento a partir de datasets ou conjuntos de dados.

O relatório documenta todas as etapas da realização do nosso projeto, desde a seleção e preparação do dataset até á implementação e avaliação dos modelos projetados, assim como os resultados reproduzidos pelos mesmos.

2 Tarefa Dataset Grupo

2.1 Domínio e Objetivo

2.1.1 Dataset escolhido

O dataset escolhido pelo grupo contém variáveis relevantes para a análise de métricas associadas à Fórmula 1, permitindo explorar diversos aspetos das corridas. As variáveis podem ser agrupadas nas seguintes categorias:

- **Informações da Corrida** : temporada (Season), número da ronda (Round), nome do circuito (Circuit), nome da corrida (Race Name), data (Date), horário da corrida (Time_of_race) e localização geográfica (Location e Country).
- **Informações do Piloto e Construtora** : nome do piloto (Driver), construtora (Constructor) e abreviação do nome do piloto (Abbreviation).
- **Desempenho na Corrida** : número total de voltas completadas (Laps), posição final (Position), número total de pit stops (TotalPitStops), variação do tempo por volta (Lap Time Variation), tentativas de volta mais rápida (Fast Lap Attempts), alterações de posição durante a corrida (Position Changes) e pontuação de agressividade do piloto (Driver Aggression Score).
- **Dados Ambientais e da Pista** : temperatura do ar (Air_Temp_C), temperatura da pista (Track_Temp_C), umidade relativa do ar (Humidity_%), e velocidade do vento em km/h (Wind_Speed_KMH).
- **Informações de Pit Stop** : tempo médio de pit stop (AvgPitStopTime), número total de pit stops (Total Pit Stops), agressividade no uso dos pneus (Tire Usage Aggression).
- **Dados sobre o Composto de Pneus e Estratégia de Corrida** : número do stint, ou seja, em que set de pneus de encontra desde o inicio da corrida (Stint), tipo de composto utilizado em cada stint (Tire Compound), e número de stints durante a corrida (Stint - repetido para indicar mais de um registo por piloto).

2.1.2 Objetivo

Com base neste dataset, propomos a análise e previsão de duas métricas principais relacionadas com o desempenho nas corridas:

- Posição final dos pilotos.
- Índice de agressividade dos pilotos.

2.2 Metodologia

Para analisar estes dados, seguimos a metodologia **SEMMA** (Sample, Explore, Modify, Model, Assess). Assim sendo:

- **Sample**: Obter o dataset de corridas de Fórmula 1 e lê-lo no KNIME.
- **Explore**: Analisar o dataset, identificando padrões de desempenho, missing values, outliers e inconsistências nos dados.
- **Modify**: Tratar de missing values, outliers e inconsistências nos dados e preparar as variáveis (ex.: posição, número de pit stops, temperatura da pista, agressividade do piloto) para análise e modelagem preditiva.

- **Model**: Foram usados diferentes modelos de regressão para prever o **posicionamento final do piloto** e o seu **índice de agressividade**, com base em variáveis como clima, estratégia de pit stop, tipo de composto usado e características da corrida.
- **Assess**: Avaliar o modelo, através de nodos disponíveis no *KNIME*, bem como o motivo dos resultados.

Este processo permitirá uma abordagem mais detalhada e estruturada na análise de desempenho em corridas de Fórmula 1, garantindo a extração de conhecimento relevante para a tomada de decisão em relação à variável-alvo, como a previsão da posição final dos pilotos ou o impacto de certas estratégias na mesma.

2.3 Descrição, Exploração e Tratamento

2.3.1 Análise inicial do dataset

Numa fase inicial, foi realizada uma exploração visual dos dados recorrendo a nodos como **Data Explorer**, **Statistics**, **Linear Correlation**, **Rank Correlation** e vários tipos de gráficos (Box Plots, Scatter Plots, Histograms, Bar Chart) aplicados aos diferentes atributos. Esta análise permitiu obter uma perceção do estado do dataset e compreender a relação entre as diferentes features.

A primeira observação relevante foi a presença de *missing values* em diversas colunas, nomeadamente: `Pit_Lap`, `Race Name`, `Date`, `Time_of_race`, `Location`, `Country`, `Air_Temp_C`, `Track_Temp_C`, `Humidity_%`, `Wind_Speed_KMH`, `Pit_Time`, `AvgPitStopTime`, `Lap Time Variation`, `Fast Lap Attempts`, `Driver Aggression Score`, `Stint`, `Tire Compound`, `Stint Length`, `Tire Usage Aggression`.

Verificou-se também que alguns atributos não se encontravam com o tipo de dados adequado, como por exemplo:

- **Números inteiros representados como decimais**: `Season`, `Round`, `Laps`, `Position`, `TotalPitStops`, `Total Pit Stops`, `Stint`, `Stint Length`, `Pit_Lap`
- **Datas representadas como strings**: `Date`, `Time_of_race`

Adicionalmente, foram detetados alguns outliers que merecem destaque, tais como:

- **AvgPitStopTime** > 800: casos em que o piloto permaneceu nas boxes devido a avaria do veículo.
- **Fast Lap Attempts** > 759: situações em que o piloto não completou a volta, normalmente pelo mesmo motivo.
- **Driver Aggression Score** > 130: valores anormais causados pela ausência de conclusão da corrida.

Por fim, observou-se que a coluna `Pit_Time` apresenta tanto valores numéricos como a string `"Final Stint"`, indicando que o piloto não voltou às boxes. Este atributo requer, portanto, um tratamento adicional para garantir a consistência dos dados.

2.3.2 Tratamento de dados

Foram realizados dois processos distintos de tratamento de dados: um destinado aos modelos de previsão da **posição final dos condutores**, e outro focado nos modelos de previsão do **índice de agressividade dos pilotos**. Ambos os processos partilham uma etapa comum de pré-processamento.

O tratamento iniciou-se com a análise dos **missing values**. Para as variáveis consideradas essenciais para a previsão, e que apresentavam uma quantidade reduzida de valores em falta, optou-se por remover as respetivas linhas. As variáveis afetadas foram: `AvgPitStopTime`, `Lap Time Variation` e `Driver Aggression Score`.

Para os restantes valores em falta, foram utilizadas as **médias** para variáveis numéricas (inteiros e decimais) e a **moda** para variáveis categóricas (strings).

Adicionalmente, foi incluído um passo no pipeline responsável por eliminar todas as **linhas duplicadas**, garantindo a integridade dos dados.

Todo este processo de tratamento pode ser visualizado na Figura 1

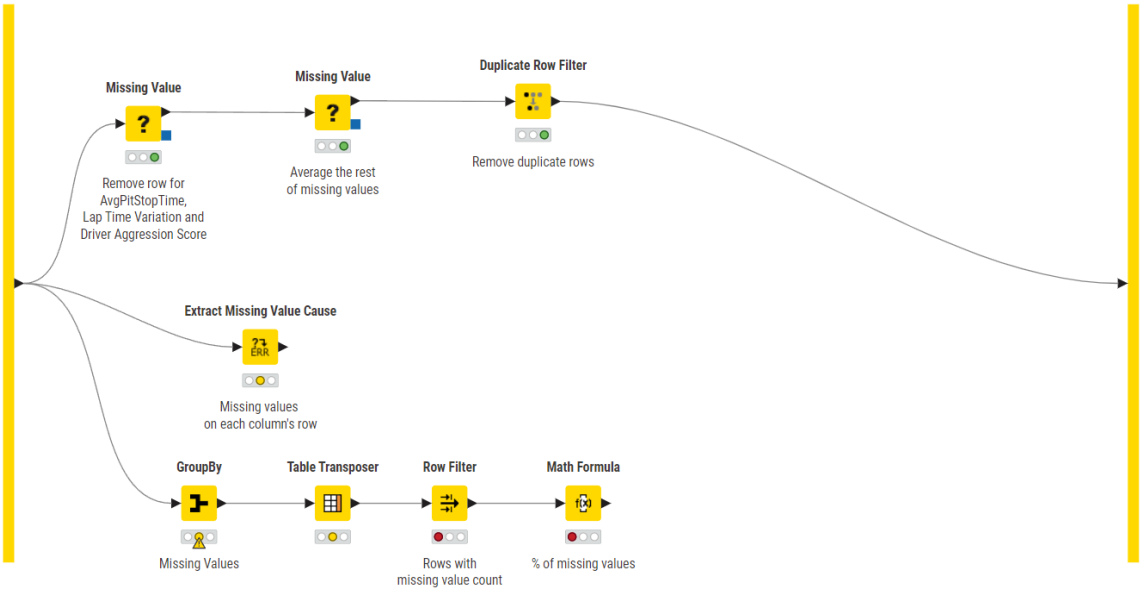

Figura 1: Tratamento de missing values

Em seguida, procedeu-se à **correção dos tipos de dados** de diversas variáveis que se encontravam com tipos inadequados para a sua natureza. As seguintes conversões foram realizadas:

- **Conversão de valores decimais para inteiros** nos atributos: `Season`, `Round`, `Laps`, `Position`, `TotalPitStops`, `Total Pit Stops`, `Stint`, `Stint Length`, `Pit_Lap`
- **Conversão de strings para objetos de data** nos atributos: `Date`, `Time_of_race`

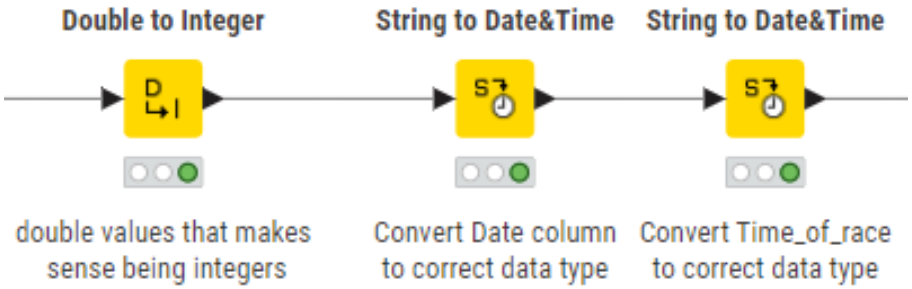

Figura 2: Conversão do tipo de dados

Posteriormente, procedeu-se ao **tratamento de outliers**, com o objetivo de eliminar valores extremos que poderiam comprometer os modelos. Foram aplicadas as seguintes regras específicas de remoção, baseadas em limites considerados irrealistas:

- **AvgPitStopTime** > 800: valores removidos por excederem o tempo plausível para uma paragem nas boxes.
- **Fast Lap Attempts** > 759: valores removidos por ultrapassarem largamente o número habitual de tentativas de volta mais rápida.
- **Driver Aggression Score** > 130: valores removidos por representarem um nível de agressividade fora do intervalo esperado.

Este tratamento foi realizado inteiramente num nodo **Row Filter** que trata de remover estes valores.

Decidiu-se também remover a coluna com o **nome completo do piloto**, mantendo apenas a **abreviação**, por se adequar melhor ao contexto da Fórmula 1, onde é comum identificar os pilotos através das suas siglas. Esta simplificação contribui para a redução de redundância e facilita o tratamento dos dados.

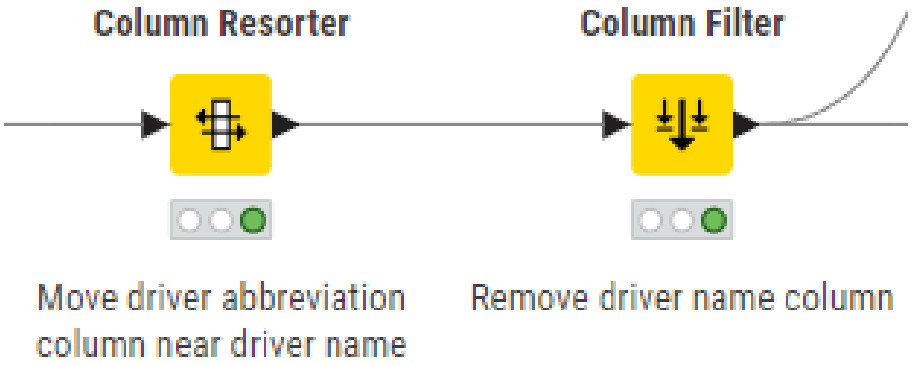

Figura 3: Remoção da coluna com o nome completo do piloto

Adicionalmente, foi realizado o **tratamento da coluna** `Pit_Time`, que apresentava valores inconsistentes devido à mistura de strings com valores numéricos decimais. Para resolver esta situação, adotaram-se as seguintes medidas:

- Os registos com o valor `"Final Stint"` foram convertidos para **0**, assumindo que o piloto não realizou mais nenhuma paragem nas boxes, não existindo assim tempo associado.
- Posteriormente, a coluna foi convertida do tipo `string` para `decimal`, uniformizando os dados.
- Finalmente, foi feita a remoção de **outliers**, eliminando os registos cujo valor de `Pit_Time` fosse superior a **90**, por se tratarem de tempos irrealistas no contexto de uma paragem em corrida.

O processo pode ser observado na Figura 4.

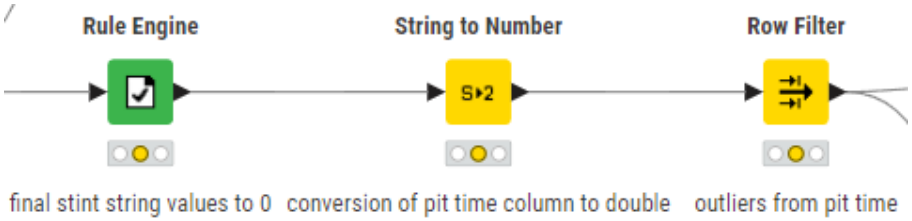

Figura 4: Tratamento do Pit_Time

Após o tratamento dos dados, foram calculadas as **correlações entre as diferentes variáveis**, permitindo identificar relações relevantes que podem influenciar a performance dos modelos de machine learning.

O resultado desta análise pode ser observado na Figura 5.

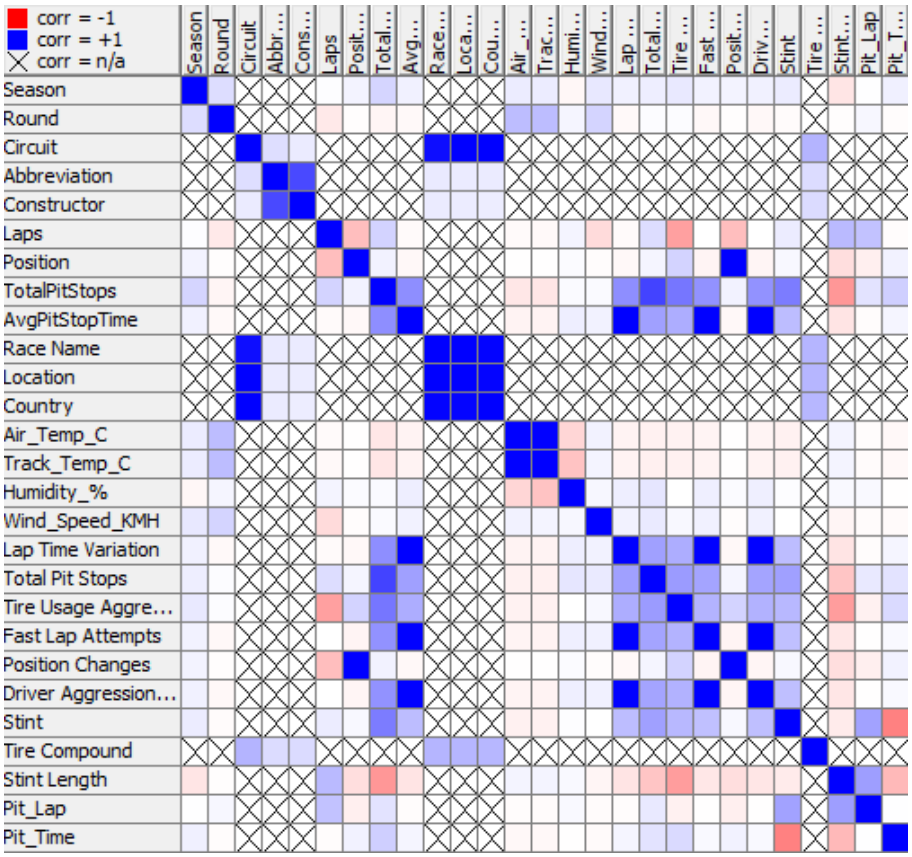

Figura 5: Correlação entre as diferentes features

Com base na análise de correlação e na relevância das variáveis para os objetivos definidos, foram selecionadas as colunas mais significativas para cada modelo. A filtragem foi feita de forma a otimizar o desempenho dos modelos de previsão, resultando nas seguintes seleções:

- **Modelo de previsão da posição final do piloto**:
  - `Season`
  - `Abbreviation`
  - `Constructor`
  - `Laps`
  - `Position`

  - TotalPitStops
  - AvgPitStopTime
  - Date
  - Lap Time Variation
  - Tire Usage Aggression
  - Fast Lap Attempts
  - Driver Aggression Score
  - Stint Length
  - Pit_Lap
  - Pit_Time

- **Modelo de previsão do índice de agressividade do piloto**:
  - Season
  - Round
  - Circuit
  - Laps
  - Position
  - TotalPitStops
  - Race Name
  - Date
  - Time_of_race
  - Location
  - Country
  - Total Pit Stops
  - Tire Usage Aggression
  - Position Changes
  - Driver Aggression Score
  - Stint
  - Stint Length
  - Pit_Time

Após o tratamento dos dados, foram geradas as matrizes de correlação específicas para cada um dos modelos.

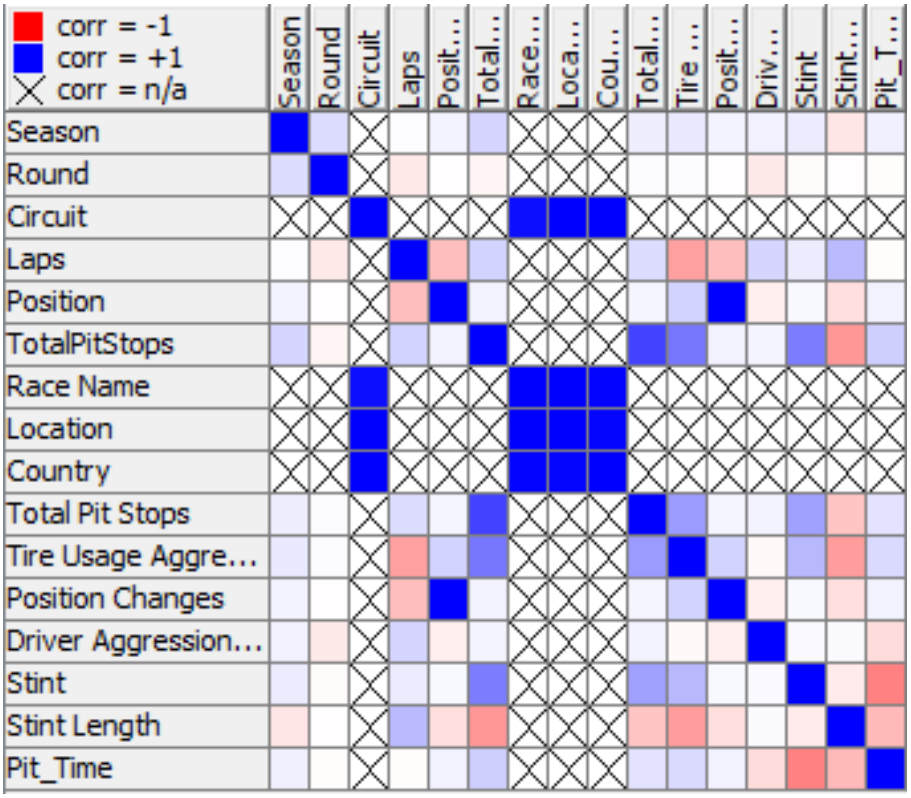

Figura 6: Correlação para os modelos de driver aggression

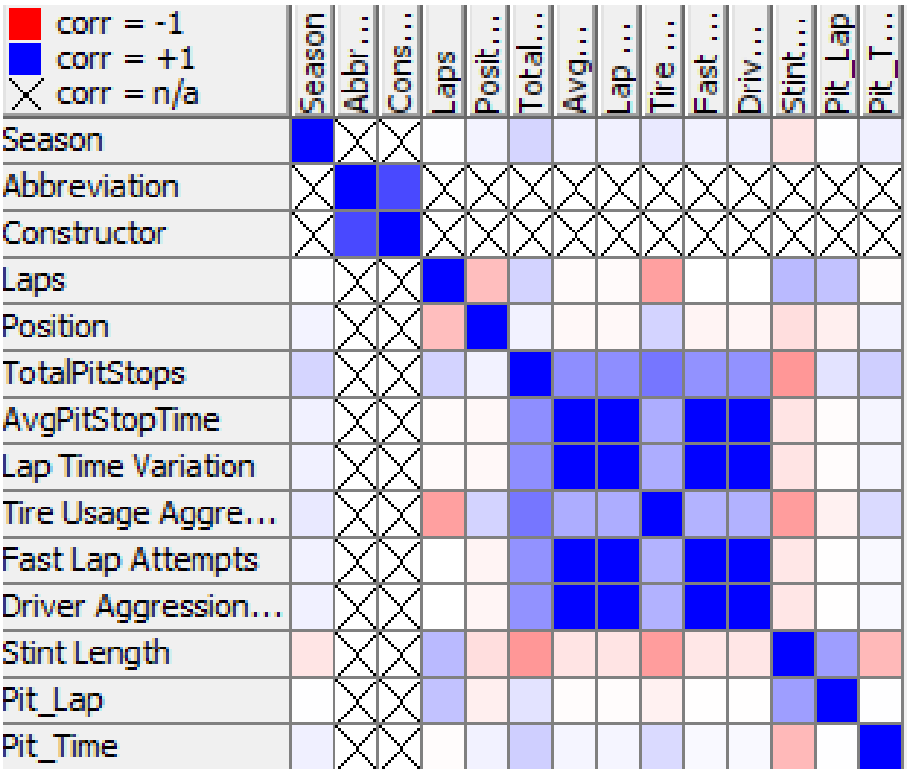

Figura 7: Correlação para os modelos de position

2.4 Modelos

Foram testados diversos modelos de forma a explorar diferentes abordagens de previsão. No total, foram utilizados cinco algoritmos distintos: Linear Regression, Random Forest (Regression), Gradient Boosted Trees (Regression), Tree Ensemble (Regression) e Simple Regression Tree.

Para cada um destes modelos, foram aplicadas duas estratégias de partição dos dados: 80/20 e 70/30, correspondentes à divisão entre conjuntos de treino e teste. Adicionalmente, os modelos Random Forest (Regression) e Gradient Boosted Trees (Regression) também foram avaliados com a técnica de validação cruzada (cross-validation), proporcionando uma análise mais robusta da sua performance.

2.5 Resultados obtidos

Resultados da posição do piloto			
Partitioning	**Modelo**	R^2	**MSE**
80/20	Linear Regression	1	0
80/20	Random Forest	0.845	4.725
80/20	Random Forest (3x trees)	0.792	6.317
80/20	Gradient Boosted	0.752	7.547
80/20	Gradient Boosted (3x models)	0.859	4.295
80/20	Tree Ensemble	0.841	4.834
80/20	Simple Regression Tree	0.844	4.744
70/30	Linear Regression	1	0
70/30	Random Forest	0.804	5.877
70/30	Random Forest (3x trees)	0.764	7.063
70/30	Gradient Boosted	0.732	8.019
70/30	Gradient Boosted (3x models)	0.829	5.11
70/30	Tree Ensemble	0.814	5.578
70/30	Simple Regression Tree	0.777	6.67
Cross Validation	Gradient Boosted	0.732	8.129
Cross Validation	Gradient Boosted (3x models)	0.848	4.616
Cross Validation	Random Forest	0.857	4.342
Cross Validation	Random Forest (3x trees)	0.86	4.234

Tabela 1: Resultados da posição do piloto

Resultados do índice de agressiviadade do piloto			
Partitioning	**Modelo**	R^2	**MSE**
80/20	Linear Regression	0.588	0.083
80/20	Random Forest	0.801	0.04
80/20	Random Forest (3x trees)	0.796	0.041
80/20	Gradient Boosted	0.772	0.046
80/20	Gradient Boosted (3x models)	0.796	0.041
80/20	Tree Ensemble	0.805	0.039
80/20	Simple Regression Tree	0.612	0.078
70/30	Linear Regression	0.573	0.084
70/30	Random Forest	0.78	0.043
70/30	Random Forest (3x trees)	0.774	0.045
70/30	Gradient Boosted	0.748	0.05
70/30	Gradient Boosted (3x models)	0.776	0.044
70/30	Tree Ensemble	0.779	0.044
70/30	Simple Regression Tree	0.656	0.068
Cross Validation	Gradient Boosted	0.781	0.045
Cross Validation	Gradient Boosted (3x models)	0.808	0.039
Cross Validation	Random Forest	0.819	0.037
Cross Validation	Random Forest (3x trees)	0.825	0.036
Cross Validation	Tree Ensemble	0.816	0.038

Tabela 2: Resultados do índice de agressiviadade do piloto

Analizando os resultados obtidos dos modelos de aprendizagem temos a seguinte análise sobre os melhores modelos:

O modelo de aprendizagem *Random Forest* baseia-se na técnica de *bagging*, isto é, na amostragem aleatória com reposição, treinando cada árvore com subconjuntos distintos dos dados de entrada. Esta abordagem permite reduzir de forma natural o *overfitting* e melhora significativamente a capacidade de generalização do modelo. Para além disso, o *Random Forest* caracteriza-se por uma estrutura relativamente simples, com um número reduzido de parâmetros a configurar, funcionando de forma eficiente mesmo sem ajustes complexos dos seus híperparâmetros.

Por outro lado, o modelo *Gradient Boosted* também se destaca como uma abordagem poderosa, frequentemente alcançando desempenhos elevados. No entanto, é igualmente mais suscetível ao *overfitting*, exigindo uma afinação cuidadosa dos seus principais híperparâmetros, como a taxa de aprendizagem (*learning rate*), a profundidade das árvores e o número de iterações. A sensibilidade a estas configurações requerem uma abordagem mais rigorosa no processo de treino, o que comprova o facto de não ser melhor que o *Random Forest* para este caso.

No contexto de validação cruzada (*cross-validation*), verifica-se uma distinção importante entre os dois modelos. O *Gradient Boosted* constrói as árvores de forma sequencial, ajustando cada novo modelo com base nos erros anteriores, o que pode conduzir a uma elevada variabilidade entre os diferentes *folds*. Em contrapartida, o *Random Forest*, ao introduzir aleatoriedade tanto na seleção de amostras como nas características utilizadas para cada árvore, tende a apresentar um desempenho mais estável e consistente entre os *folds* gerados, por exemplo, através do nó *X-Partitioner* no *KNIME*.

Adicionalmente, os resultados obtidos indicam que o modelo *Tree Ensemble* apresenta também um desempenho robusto quando sujeito a validação cruzada. Este comportamento explica-se pelo facto deste modelo ser igualmente baseado em *bagging*, tal como o *Random Forest*, recorrendo à construção de múltiplas árvores de decisão treinadas sobre subconjuntos diferentes dos dados e integrando elementos de aleatoriedade no processo. Esta estrutura contribui para uma boa capacidade de generalização.

No entanto, apesar da sua semelhança estrutural com o *Tree Ensemble*, o *Random Forest* evidencia-se como o modelo com melhor desempenho em contexto de validação cruzada. Esta superioridade poderá justificar-se pela maior flexibilidade na configuração dos seus hiperparâmetros, permitindo um nível de otimização mais elevado. O *Random Forest* possibilita, por exemplo, o ajuste do número de árvores, da profundidade máxima, do número mínimo de registos por nó terminal, entre outros parâmetros. Em contrapartida, o modelo *Tree Ensemble* é frequentemente utilizado com configurações mais genéricas, podendo por isso não estar tão bem adaptado a determinadas características específicas do conjunto de dados em análise.

3 Tarefa Dataset Atribuído

3.1 Domínio e Objetivo

O dataset atribuído inclui variáveis relevantes para a análise da viabilidade de um empréstimo, tais como:

- **Características do Cliente** : idade, género, rendimento anual e experiência profissional.
- **Informações acerca do empréstimo** : valor solicitado, propósito para o empréstimo, taxa de juro, e relação entre o valor do empréstimo e o rendimento do cliente.
- **Histórico Financeiro Pessoal** : tempo de histórico de crédito, existência de incumprimentos, impostos e contagem de créditos.
- **Estado de Empréstimo** : Indica se o empréstimo foi **aprovado** ou **recusado**.

O objetivo será, apartir do dataset atribuído, desenvolver um modelo de machine learning capaz de prever a decisão da variável alvo do conjunto de dados, a aprovação ou rejeição de um pedido de empréstimo mediante as características atribuídas.

Os principais objetivos deste estudo são:

- **Exploração e preparação dos dados** : Analisar o dataset, identificar padrões, tratar missing values e outliers e preparar os dados para serem modelados.
- **Desenvolvimento de modelos de previsão** : Construção do modelo para a previsão se um empréstimo será **aprovado**, **reprovado** ou ficará **pendente**. Além disso, testar modelos de regressão para previsão de dados tais como montantes de empréstimo baseado no rendimento do cliente e taxas de juro.
- **Avaliação e otimização dos modelos desenvolvidos** : Comparar diferentes algoritmos e aplicar técnicas de otimização para melhorar o desempenho dos modelos.
- **Identificação de padrões** : Identificar quais as variáveis que têm maior impacto na decisão final do empréstimo, e como as características influenciam o resultado.
- **Aplicação na ferramenta KNIME** : Implementar os modelos finais e analisar os resultados. Utilizando o KNIME será possível através da interface visualizar os dados resultantes.

3.2 Metodologia

Para obter os objetivos citados acima será necessário ter em conta esta metodologia apresentada abaixo:

- **Exploração e preparação dos dados**: Análise do dataset da autorização de um crédito, avaliando a distribuição e a consistência das variáveis socioeconómicas dos indivíduos (como `person_age`, `person_income`, `person_education`, `person_emp_exp`, etc.). Tratamento de missing values, outliers e inconsistências, bem como transformação de variáveis categóricas (ex.: `person_gender`, `loan_intent`, `person_home_ownership`) em formatos adequados para modelos.
- **Desenvolvimento de modelos de previsão** : Construção de modelos preditivos para a variável-alvo `loan_status`, com base em atributos como a intenção do empréstimo (`loan_intent`), histórico de crédito (`credit_score`, `previous_loan_defaults_on_file`), e fatores financeiros (`loan_amnt`, `loan_int_rate`, `loan_percent_income`, `tax`). Utilização de algoritmos supervisionados para prever a aprovação ou rejeição dos empréstimos.

- **Avaliação e otimização dos modelos desenvolvidos** : Comparação entre diferentes algoritmos de classificação, como Random Forest, XGBoost, e Regressão Logística. Aplicação de cross validation e ajuste de hiperparâmetros (Grid Search, Random Search) visando maximizar métricas como accuracy, F1-score e AUC-ROC.

- **Identificação de padrões** : Investigação das variáveis mais relevantes para a concessão de crédito e o risco de rejeição, utilizando técnicas de importância de atributos. Exploração de como fatores como o histórico de crédito, nível de escolaridade e experiência profissional impactam a decisão de empréstimo.

- **Aplicação na ferramenta KNIME** : Implementação de todo o processo analítico na plataforma KNIME, desde o carregamento e pré-processamento do dataset até a modelagem e análise de resultados. Utilização de workflows visuais para facilitar a replicação dos experimentos e a interpretação dos modelos preditivos.

Este processo fará com que seja possível uma abordagem mais detalhada e estruturada na análise ao risco do crédito, garantido a extração somente de conhecimento útil para a tomada de decisão da variável-alvo.

3.3 Descrição, Exploração e Tratamento

3.3.1 Análise inicial do dataset

Numa fase inicial, foi realizada uma exploração visual dos dados recorrendo a nodos como **Data Explorer**, **Statistics**, **Rank Correlation** e vários tipos de gráficos (Box Plots, Scatter Plots, Histograms, Bar Chart) aplicados aos diferentes atributos. Esta análise permitiu obter uma perceção do estado do dataset e compreender a relação entre as diferentes features.

A primeira observação relevante foi a presença de *missing values* em diversas colunas, em ambos os dataset, tanto no de treino como de aprendizagem, nomeadamente: *person_age*, *person_income*, *person_emp_exp*, *loan_amnt*, *loan_int_rate*, *loan_percent_income*, *cb_person_cred_hist_length* e *credit_score*.

Verificou-se também que alguns atributos não se encontravam com o tipo de dados adequado, como por exemplo:

- **Datas representadas como strings**: `Date`

Para tal, decidiu-se separar a data como string para novas *features* ano, mês e dia, verificou-se que existiam valores para a *feature* dia que equivaliam a 30 e 31 para a o mês de fevereiro, o que representa um valor inválido. Após o tratamento das respetivas linhas em que ocorria o sucedido, decidiu-se manter apenas a *feature* ano que possuia correlação com a idade do cliente.

Foi verificado também que algumas colunas continham vários valores que representavam o mesmo conteúdo, como por exemplo:

- **gender**: Foi detetado que *M* e *Men* significava o mesmo que *male* e *F* e *Women* o mesmo que *female*.

- **person_education**: Foi detetado que *Dr.*, *Doctor*, *Doctrate*, *Doctoral* e *PhD* significava o mesmo que *Doctorate*, que *Bachelors*, *Bachlor* e *BSc* significava o mesmo que *Bachelor*, que *Assoc. Degree*, *Associates* e *Assoc* significa o mesmo que *Associate*, que *Highschool*, *H-School* e *HS* significava o mesmo que *High School* e por fim que *MSc*, *Masters* e *Mstr* significava o mesmo que *Master*.

- **ownership**: Foi detetado que *RNT* significava o mesmo que *RENT*, que *MORTG* e *MRTG* significava o mesmo que *MORTGAGE*, que *OWNN* e *OWNERSHIP* significava o mesmo que *OWN* e por fim que *OTHR* e *OTER* significava o mesmo que *OTHER*.
- **intent**: Foi detetado que *EDUCTION* e *EDU* significava o mesmo que *EDUCATION*, que *MDICAL* e *MED* significava o mesmo que *MEDICAL*, que *VENTUREE* significava o mesmo que *VENTURE*, que *PERSONL* e *PERSON* significava o mesmo que *PERSONAL*, que *DEBTCONS* significava o mesmo que *DEBITCONSOLIDATION* e por fim que *HOMEIMP* e *HOME-IMPROVE* significava o mesmo que *HOMEIMPROVEMENT*.

Adicionalmente, foram detetados alguns outliers que merecem destaque, tais como:

- **person_income**: foram filtrados os rendimentos superiores a 1.440.915 e foi aplicada uma função logarítmica a valores maiores que 1 milhao (log(person_income + 1)) para o tratamento de outliers nesta coluna
- **loan_percent_income**: foi aplicada uma função logarítmica para valores superiores a 1.1 (log(loan_percent_income +1)), para o tratamento de outliers nesta coluna

Para o tratamento destes outliers foi utilizado o nodo *Rule-based Row Filter* para filtrar os rendimentos acima do valor estipulado e o nodo *Java Snippet* para aplicar a função logarítmica $\log(feature + 1)$).

A estratégia apresentada oferece uma melhor precisão dos modelos de aprendizagem, face às seguintes estratégias anteriormente implementadas, a remoção de outliers a partir de um certo valor ou até mesmo a passagem de outliers para missing values e posteriormente substituídos para a média, mediana, etc, não apresentavam valores que favoreciam assim tanto os modelos de aprendizagem.

3.3.2 Tratamento de dados

Foi realizado um processo de tratamento de dados destinado ao modelos de previsão do **estado do empréstimo**.

O tratamento iniciou-se com a análise dos **missing values**.Foi utilizado o nodo Missing Value para tratar as colunas que foram referidas em cima. Foram utilizadas **médias** para variáveis decimais e nada para variáveis inteiras e também para strings, pois não existiam missing values tanto para strings como para variáveis inteiras.

O tratamento de missing values pode ser observado na Figura 8

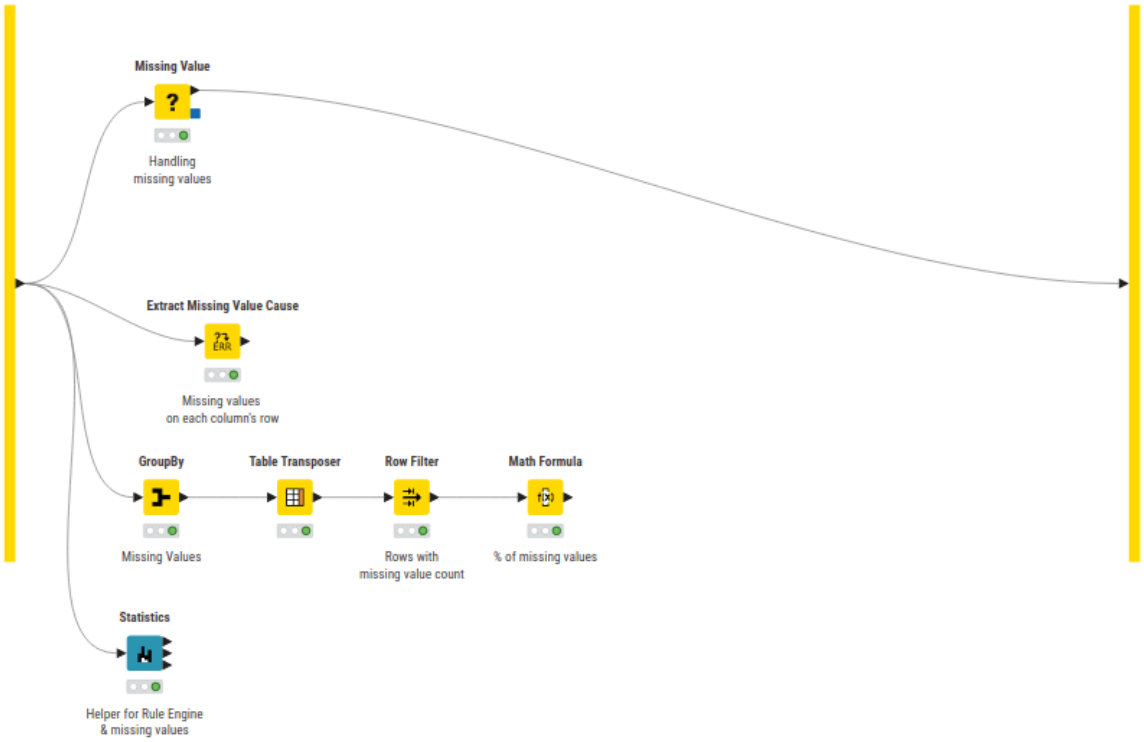

Figura 8: Tratamento de missing values

Adicionalmente, foi incluído um passo no pipeline responsável por tratar de todas as variáveis que tinham diferentes designações mas significado idêntico referidas na secção acima. Foi efetuado o encoding para que todas resultassem num só significado respetivo a cada categoria.

Todo este processo de tratamento pode ser visualizado na Figura 9

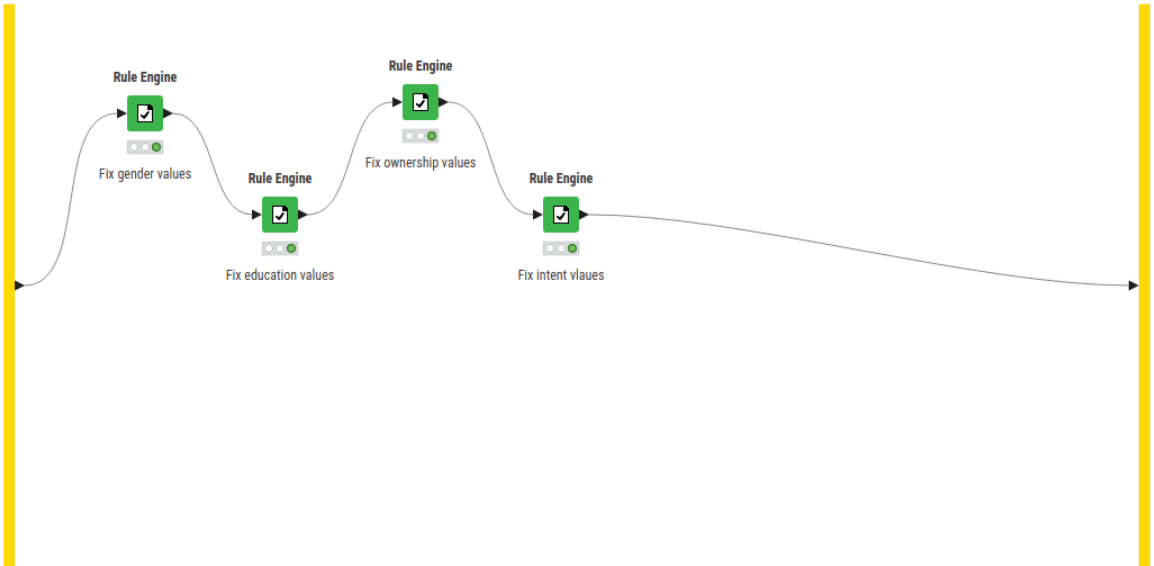

Figura 9: Tratamento de variáveis idênticas

Exemplo de encoding efetuado para estes valores idênticos pode ser visualizado na Figura 10

```
Expression
 9 $person_education$ = "Bachelors" => "Bachelor"
10 $person_education$ = "Bachlor" => "Bachelor"
11 $person_education$ = "BSc" => "Bachelor"
12 // Handle values that represent Associate
13 $person_education$ = "Assoc. Degree" => "Associate"
14 $person_education$ = "Associates" => "Associate"
15 $person_education$ = "Assoc" => "Associate"
```

Figura 10: Exemplo de encoding efetuado

Além disso, foi também efetuado o tratamento da variável *date*, que era recebida como uma *string*. Considerámos pertinente realizar esse tratamento não apenas devido ao seu tipo, mas também por causa da existência do mês de fevereiro, que poderia tornar os dados inconsistentes, uma vez que possui apenas 28 dias.

Todo este processo de tratamento pode ser visualizado na Figura 11

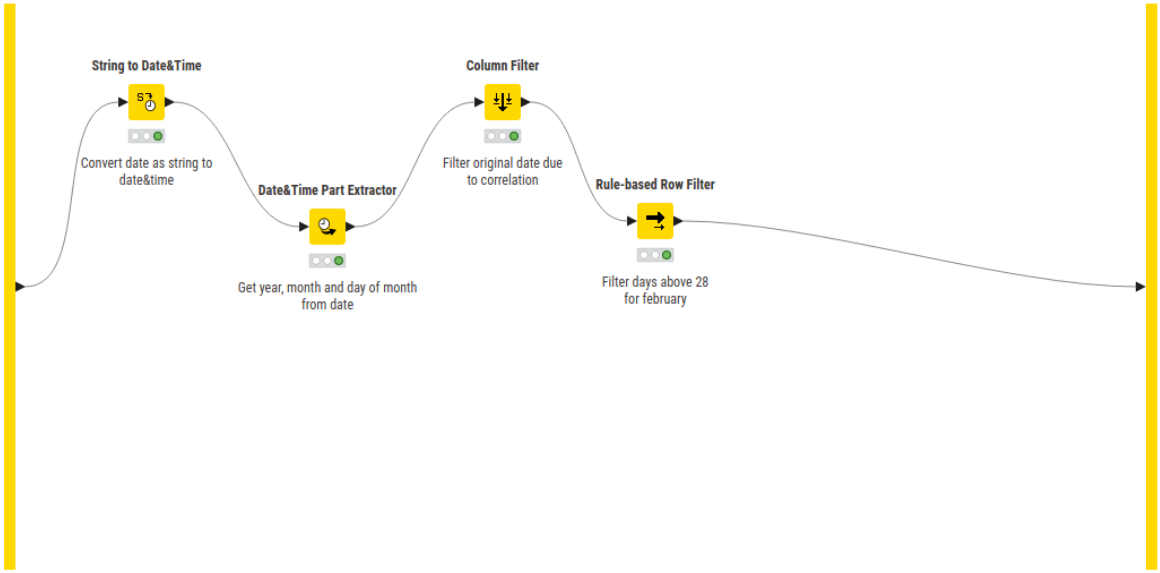

Figura 11: Tratamento efetuado para a variável *date*

Também foi efetuado o tratamento de outliers referido em cima, através da aplicação de uma função logarítmica a valores superiores a 1 milhão na variável *person_income* e em valores superiores a 1.1 na variável *loan_percent_income*. Este tratamento foi efetuado para reduzir o impacto de valores extremos e tornar a distribuição mais próxima de uma normal nessas respetivas variáveis. Foram adicionados box plots no meio do processo para verificar o resultado de cada passo.

Todo este processo de tratamento pode ser visualizado na Figura 12

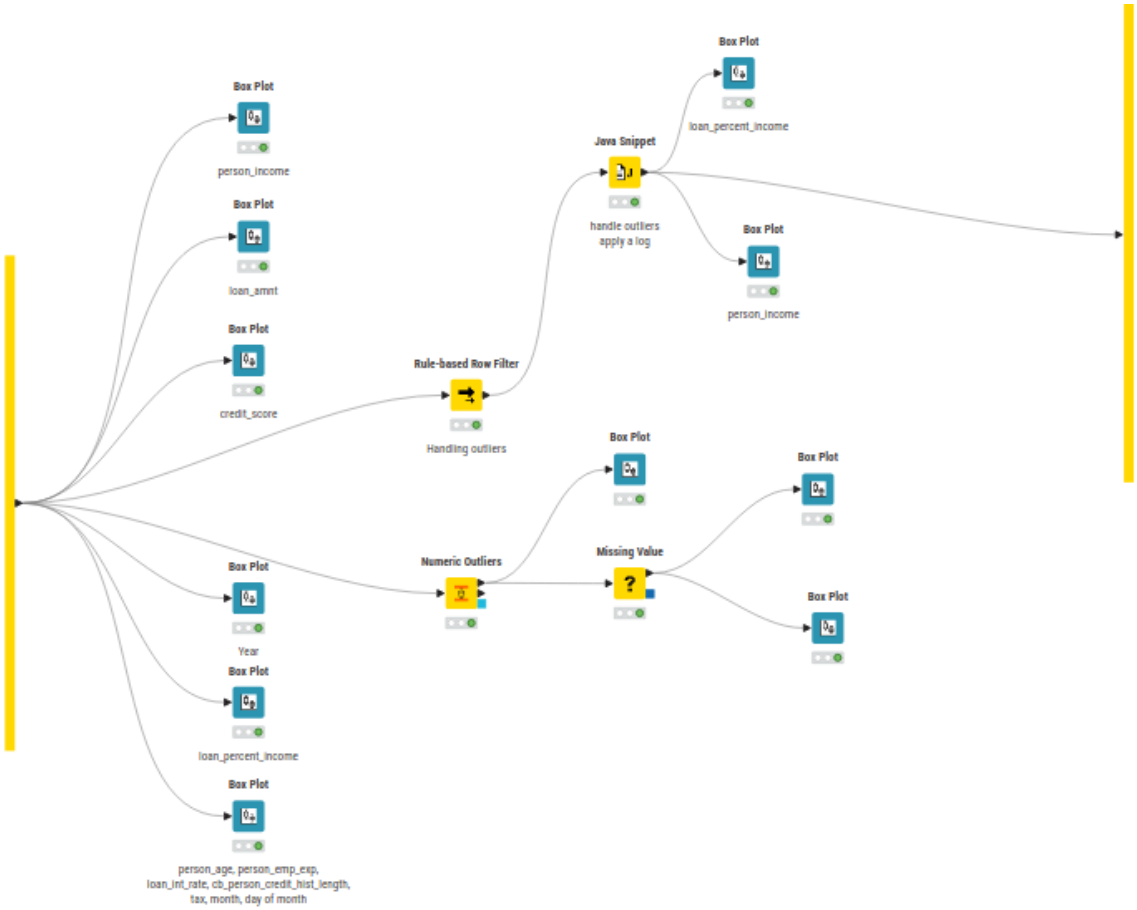

Figura 12: Tratamento efetuado para outliers

Por fim, efetuámos também a remoção de algumas colunas que verificámos ter pouca correlação com o modelo, nomeadamente: *tax*, *person_name*, *person_gender*, *month*, *day_of_month* e *loan_intent*.

Verificámos ainda que algumas variáveis, como a *person_education*, apresentam efetivamente pouca correlação. No entanto, como após vários testes os resultados se mantinham iguais com ou sem essa variável, optámos por mantê-la no modelo.

Foram adicionados alguns nós de *Rank Correlation* durante o processo, para observar como variava a correlação à medida que as colunas eram removidas.

Todo este processo de tratamento pode ser visualizado na Figura 13

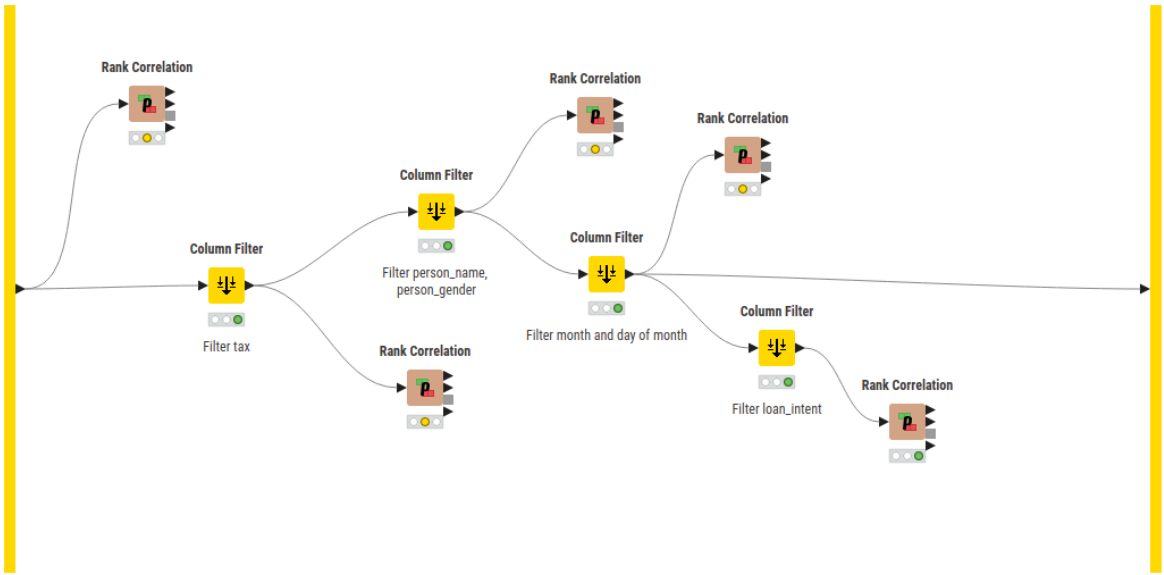

Figura 13: Tratamento efetuado para variáveis com pouca correlação

3.4 Modelos

Para prever o estado de um empréstimo, foram testados diversos algoritmos de classificação baseados em árvores de decisão, nomeadamente: `Decision Tree`, `Random Forest`, `Gradient Boosted Trees` e `Tree Ensemble`. A avaliação do desempenho dos modelos foi realizada utilizando a *partitioning* fornecida de treino/teste (90/10), bem como *cross-validation*, com o objetivo de garantir maior robustez na análise dos resultados.

Adicionalmente, foram feitos testes com diferentes variações dos modelos, como o aumento do número de árvores no caso do `Random Forest` e a combinação de múltiplos modelos no caso do `Gradient Boosted Trees`, de forma a explorar possíveis melhorias no desempenho.

3.5 Resultados obtidos

A Tabela 3 apresenta os resultados obtidos por cada modelo, avaliados com base na métrica de `accuracy` (precisão) e no `Cohen's kappa`.

Os modelos baseados em `Gradient Boosted Trees` apresentaram os melhores desempenhos em termos de `accuracy` e `kappa`, com destaque para a versão com múltiplos modelos (`3x models`), que atingiu a mesma precisão que a versão mais simples, tendo um valor de κ ligeiramente maior (0.329%). Em contraste, o modelo mais simples, a `Decision Tree`, teve o desempenho mais fraco, com uma precisão de apenas 50.524

A utilização de validação cruzada resultou em ligeira redução da performance dos modelos, o que é esperado, dado o maior rigor na avaliação. Ainda assim, os resultados mantiveram-se consistentes, o que indica que os modelos têm um comportamento estável.

Resultados do estado do empréstimo			
Partitioning	**Modelo**	**Accuracy**	κ
90/10	Random Forest	61.2%	0.319%
90/10	Random Forest (3x trees)	61.556%	0.321%
90/10	Decision Tree	50.524%	0.19%
90/10	Gradient Boosted	62.044%	0.324%
90/10	Gradient Boosted (3x models)	62.044%	0.329%
90/10	Tree Ensemble	61.089%	0.317%
Cross Validation	Gradient Boosted	60.862%	0.31%
Cross Validation	Random Forest	59.729%	0.301%

Tabela 3: Resultados do estado do empréstimo

4 Sugestões e Recomendações

Com base na análise crítica dos resultados obtidos e na avaliação dos modelos desenvolvidos, é possível identificar várias oportunidades de melhoria e aprofundamento do trabalho realizado. Abaixo são apresentadas algumas sugestões e recomendações com vista à obtenção de desempenhos mais robustos e generalizáveis:

- **Aprofundamento dos modelos com melhor desempenho:** Os modelos que apresentaram os melhores resultados deverão ser alvo de uma exploração mais aprofundada. Em particular, recomenda-se a realização de experiências adicionais com versões expandidas destes modelos, nomeadamente através do aumento significativo do número de *trees* (por exemplo, 10 vezes mais), no caso de modelos baseados em *random forest* ou *gradient boosting*. Esta abordagem poderá permitir uma maior capacidade de generalização e um refinamento adicional na aprendizagem de padrões complexos dos dados.

- **Exploração alargada do espaço de atributos (features):** Uma outra linha de evolução passa pela consideração de diferentes combinações de atributos, bem como pela criação de novas *features* a partir das existentes, através de técnicas de engenharia de atributos. Esta estratégia poderá revelar relações e padrões nos dados que não foram inicialmente evidentes, contribuindo assim para um desempenho preditivo superior.

- **Avaliação da estabilidade dos modelos:** Sugere-se ainda a realização de análises de sensibilidade e de estabilidade dos modelos perante variações nos dados de treino, nomeadamente através de técnicas como a validação cruzada estratificada e o uso de conjuntos de validação adicionais. Este tipo de análise permitirá aferir a robustez dos modelos desenvolvidos face a flutuações naturais nos dados.

- **Integração de técnicas de seleção de atributos:** Poderá ser pertinente integrar métodos automáticos de seleção de atributos, como *Recursive Feature Elimination* (RFE) ou análise de importância de atributos, com o objetivo de reduzir a dimensionalidade e eliminar informação redundante ou ruidosa, o que poderá melhorar tanto a eficiência como a interpretabilidade dos modelos.

- **Experimentação com outros algoritmos de aprendizagem automática:** Para trabalhos futuros, seria interessante experimentar outros algoritmos de `machine learning`, como o `SVM` ou o `k-NN`, com o objetivo de comparar o seu desempenho face aos modelos implementados neste estudo.

- **Análise visual dos resultados:** A inclusão de uma componente visual mais aprofundada na análise dos resultados poderá facilitar a identificação de padrões e tendências nos dados, contribuindo para uma interpretação mais clara e compreensiva do comportamento dos modelos preditivos.

Estas recomendações visam não apenas otimizar os modelos desenvolvidos, mas também fomentar uma abordagem mais sistemática e exploratória no tratamento de problemas semelhantes no futuro.

5 Conclusões

Este trabalho prático teve como principal objetivo a exploração e análise de dois `datasets`, um orientado para tarefas de classificação e outro para regressão.

Numa fase inicial, enfrentámos alguns desafios na seleção das `features` mais relevantes para os modelos. No entanto, através da realização de diversos experimentos e testes, foi possível obter resultados satisfatórios e identificar os elementos mais influentes para cada tipo de tarefa.

Em síntese, esta experiência permitiu-nos aprofundar o conhecimento sobre as várias formas de abordar um `dataset` e reforçou a importância do `machine learning` como ferramenta para extrair valor e obter insights a partir dos dados.