



UNIVERSIDADE DO MINHO
Departamento de Informática

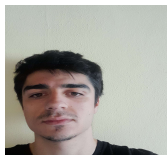
APRENDIZAGEM E DECISÃO INTELIGENTES

Relatório do Trabalho Prático

Grupo 28

Feito por:

Dinis Gonçalves Estrada (A97503)
Emanuel Lopes Monteiro da Silva (A95114)



A97393



A95114

May 13, 2023
Ano Letivo 2022/23

Conteúdo

1	Introdução	2
2	Metodologia e aplicação	2
3	Arquitetura do Workflow Knime	2
4	Tarefa A - Melbourne Housing Market	3
4.1	Domínio e objetivos	3
4.2	Exploração e Tratamento de dados	4
4.2.1	Descrição dos atributos do dataset	4
4.2.2	Exploração e tratamento de dados	5
4.3	Modelos	10
4.3.1	Técnicas de aprendizagem	10
4.3.2	Tunning dos Hiperparâmetros	11
4.3.3	Partitioning	11
4.3.4	Cross Validation	11
4.3.5	Métricas de qualidade	12
4.4	Resultados	12
5	Tarefa B - Obesidade	13
5.1	Domínio e objetivos	13
5.2	Exploração, Visualização e Tratamento de dados	13
5.2.1	Descrição dos atributos do dataset	13
5.2.2	Exploração dos dados	14
5.3	Modelos	16
5.3.1	Tunning dos Hiperparâmetros	17
5.3.2	Decision Tree	17
5.3.3	Gradient Boosted Trees Predictor	17
5.4	Conclusão e análise dos resultados obtidos	18
6	Conclusão	19

1 Introdução

O seguinte projeto encaixa-se na UC de Aprendizagem e Decisão Inteligentes, na qual nos foi proposto realizar uma conceção e otimização de modelos de *Machine Learning* com o uso da plataforma KNIME.

O desafio apresentado visa desenvolver um projeto utilizando modelos de aprendizagem abordados ao longo do semestre e tem por base duas tarefas principais.

Primeiramente foi necessário consultar, analisar e seleccionar um dataset de entre os possíveis de diferentes fontes, tais como *UCI Machine Learning Repository* e *Kaggle*. Como seguinte etapa foi realizada a conceção e otimização de modelos, tanto para o dataset que nos foi atribuído pela equipa docente como para o dataset escolhido por nós, em que se usou o *Kaggle* para tal selecção. Além disso também foram desenvolvidos modelos de visualização e realizada uma análise crítica dos resultados, com o objetivo de compreender completamente o conjunto de dados e suas características.

2 Metodologia e aplicação

A metodologia escolhida pelo grupo é a CRISP-DM. O grupo decidiu seguir esta metodologia pois embora esta metodologia seja voltada para desenvolvimento de projetos em concreto, as etapas de Estudo dos Dados, Preparação dos Dados, Modelação e Avaliação do Modelo, estão dentro do âmbito desta cadeira e serão as etapas abordadas neste relatório.

3 Arquitetura do Workflow Knime

Para a construção do workflow seguimos algumas boas práticas como: o uso de Anotações, ou seja, cada nodo tem associado uma descrição do seu objetivo; o uso de Metanodos, pois em algumas tarefas mais complexas, que exigem múltiplos nodos foram compactados num único nodo (metanodo); o uso de Secções que representam as principais fases do pipeline de machine learning, existem 5 destas secções: Dataset Reader (Roxo), Data Preparation (Verde), Data Visualization (Magenta), Models Learners (Laranja), Models Analysis (Azul).

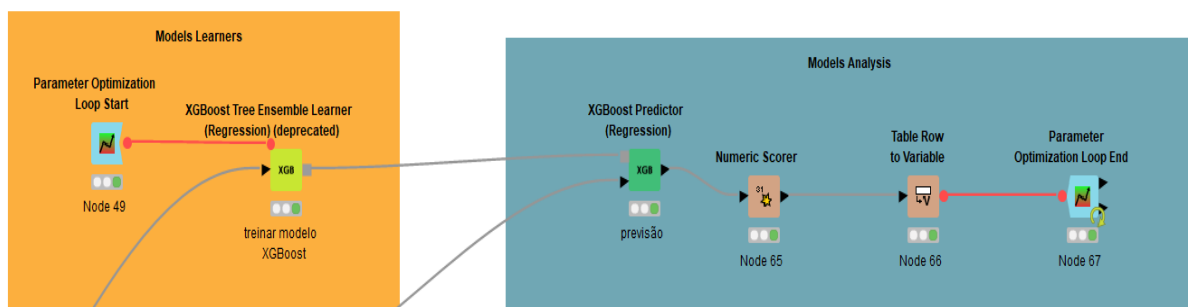


Figure 1: Excerto do workflow relativo às secções models learners e analysis

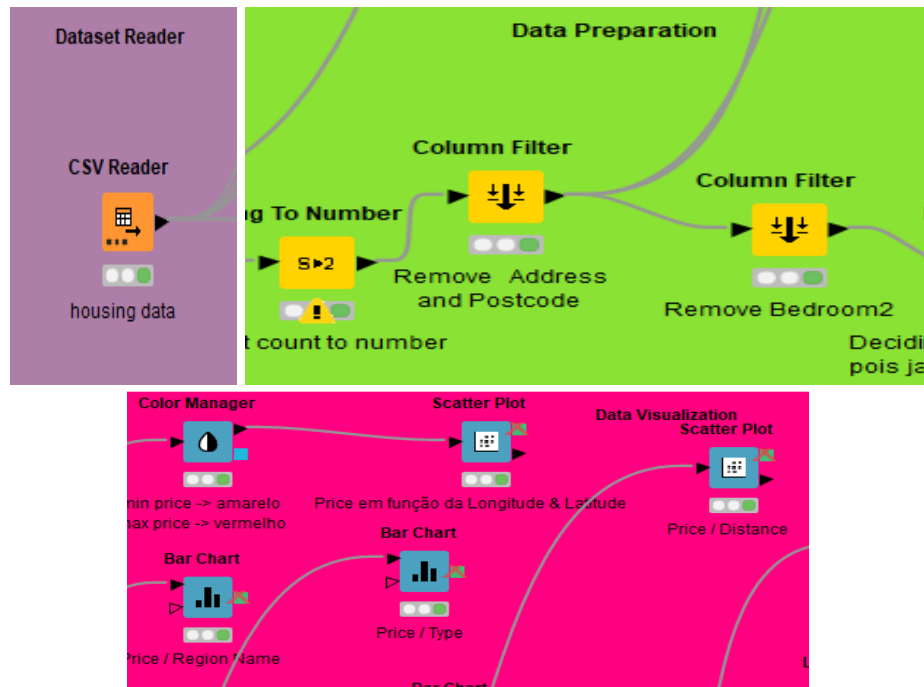


Figure 2: Excerto do workflow relativo às secções readers, preparation e visualization

4 Tarefa A - Melbourne Housing Market

4.1 Domínio e objetivos

O propósito desta tarefa é prever o preço de casas de Melbourne. O dataset usado nesta tarefa foi retirado do Kaggle pelo seguinte link disponibilizado por Tony Pinto.

O dataset está disponível publicamente sob o número de licença CC BY-NC-SA 4.0. Algumas das variáveis incluídas no dataset são:

- Endereço
- Tipo de imóvel
- Bairro
- Método de venda
- Quartos
- Preço
- Tamanho da propriedade... etc.

O objetivo desta tarefa é examinar a influência de vários preditores, ou seja, variáveis do conjunto de dados, nos preços de imóveis em Melbourne, e, ao fazer isso, descobrir os preditores com maior potencial para prever os preços de imóveis. A variável alvo é *Price* de casas em Melbourne, enquanto:

- Quarto
- Banheiro
- Área construída

- Distância
- Tamanho do terreno

são algumas das variáveis consideradas as mais influentes para a predição de preços de imóveis.

4.2 Exploração e Tratamento de dados

4.2.1 Descrição dos atributos do dataset

- *Suburb*: Área suburbana
- *Address*: Endereço da casa
- *Postcode*: Código Postal
- *Rooms*: Número de quartos na casa
- *Price*: Preço em dólares australianos
- *Method*: S - propriedade vendida; SP - propriedade vendida antecipadamente; PI - propriedade não vendida em leilão; PN - vendida antecipadamente e o valor não divulgado; SN - vendida e o valor não divulgado; NB - sem oferta; VB - lance do vendedor; W - retirada antes do leilão; SA - vendida após o leilão; SS - vendida após o leilão e o valor não divulgado. N/A - preço ou lance mais alto não disponível.
- *Type*: br - bedroom(s); h - *house, cottage, villa, semi terrace*; u - *unit, duplex*; t - *townhouse*; dev site - *development site*; o res - *other residential*.
- *SellerG*: Agente imobiliário
- *Date*: Data de venda
- *Distance*: Distância do centro da cidade em quilómetros (CBD - Central Business District)
- *Regionname*: Região (*West, North West, North, North east ...etc*)
- *Propertycount*: Número de propriedades que existem no subúrbio.
- *Bedroom2*: Número de quartos (de uma fonte diferente)
- *Bathroom*: Número de casas de banho
- *Car*: Número de vagas de estacionamento
- *Landsize*: Tamanho do terreno em metros quadrados
- *BuildingArea*: Tamanho do edifício em metros quadrados
- *YearBuilt*: Ano em que a casa foi construída
- *CouncilArea*: Conselho governante para a área

4.2.2 Exploração e tratamento de dados

O conjunto de dados de preços de imóveis em Melbourne consiste em 34.857 linhas (observações) e 21 colunas rotuladas (variáveis). Tendo isto em mente, é necessário encontrar as variáveis, ou seja, preditores, dentro do conjunto de dados fornecido que apresentam alta correlação com a variável alvo. Os preditores devem ser encontrados através de uma análise exploratória dos dados. Primeiro, é feita a distribuição de frequência da variável *Price* sem valores N.A. (não atribuídos) e sem outliers, para obter uma visão geral dos parâmetros estatísticos básicos da variável alvo. Em segundo lugar, as variáveis são atribuídas a uma das três categorias relevantes para o conjunto de dados em questão. As categorias relevantes para o conjunto de dados são:

- Variáveis relacionadas com a localização
- Variáveis relacionadas com o imóvel
- Variáveis relacionadas com o vendedor

Seguindo a abordagem anterior, a coluna *Price* é verificada para remover as linhas que contenham valores N.A., missing values. Foi constatado que existem 7610 valores N.A. na coluna *Price*. Todas as linhas que os contem são removidas. A coluna *Price* também é verificada por outliers, sendo estes removidos. Na figura 1 é apresentada a distribuição de frequência da variável distribuição *Price*.

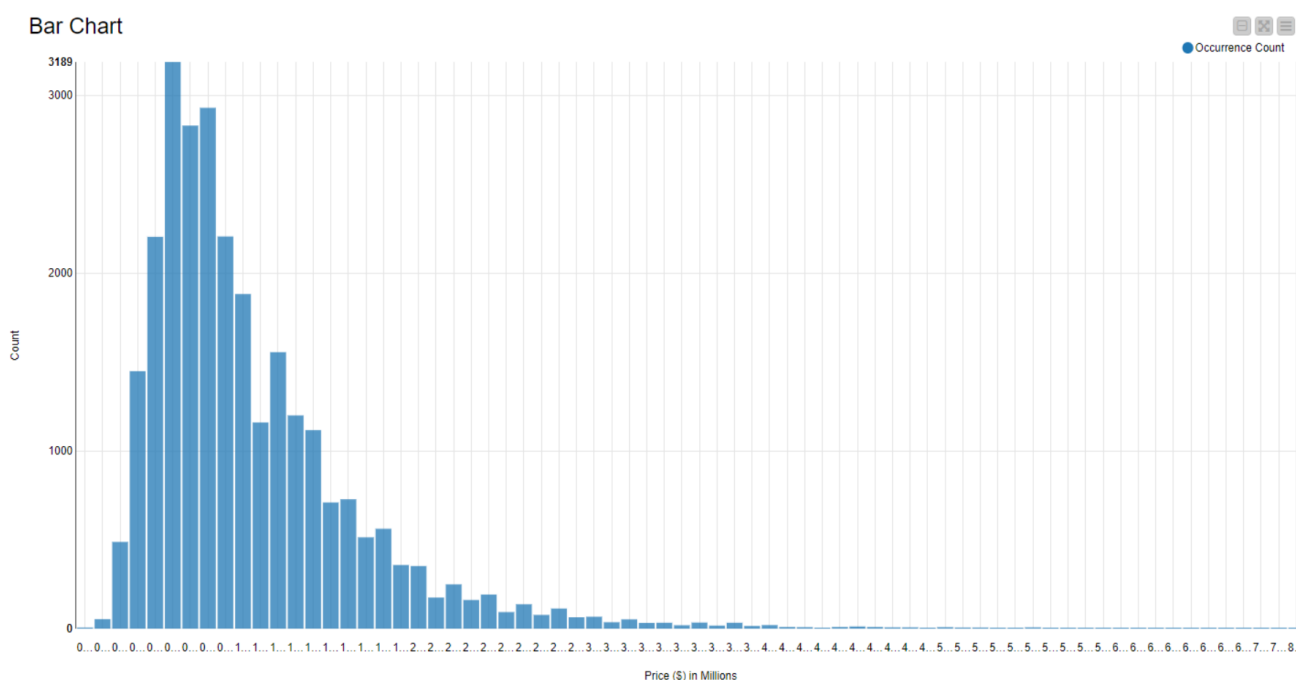


Figure 3: A distribuição de preços de imóveis após a limpeza da coluna *Price* dos missing values

Conforme explicado anteriormente, a fim de fazer uma seleção preliminar das variáveis, estas são distribuídas em três categorias de interesse com base nas informações que cada variável carrega. As categorias com as variáveis relevantes atribuídas são as seguintes:

- Variáveis relacionadas com a localização: *Suburb*, *Address*, *Postcode*, *Council Area*, *Latitude* e *Longitude* (Embora dadas como variáveis separadas, elas podem mostrar a relação entre preço e posição da casa caso forem usadas em conjunto.), *Region Name*, *Property Count*, *Property Type*, *Distance*

- Variáveis relacionadas com o imóvel: *Rooms*, *Bedroom2*, *Bathroom*, *Land Size*, *Building Area*, *Year Built*, *Car*
- Variáveis relacionadas com o vendedor: *SellerG*, *Method*, *Date*

Variáveis relacionadas com a localização

Adress: Como a variável é única para cada casa, ou seja, cada linha do dataset, dificilmente pode trazer algum benefício para o objetivo principal que é prever o valor de imóveis. Por isso esta coluna é removida.

Region Name, *Council Area*, *Suburb* and *Postcode*: Representam as regiões administrativas da cidade de Melbourne, sendo que a *Region* é a mais importante na hierarquia. Depois vem a *CouncilArea* que pode governar mais que um subúrbio. O *Postcode* é basicamente um identificador do *Suburb* logo podemos usar apenas uma destas variáveis, pelo que a coluna *Postcode* foi removida. Podemos concluir então que as 3 variáveis podem ser boas preditoras do preço. Mas por outro lado, é importante estar ciente que podem coexistir imóveis de valor baixo e alto no mesmo *Suburb*, *CouncilArea* e *Region*. Pela análise da variável *Region* é observado que o as regiões que possuem o preço de imóveis mais elevado são *Southern*, *South-Eastern* e *Eastern Metropolitan regions*.

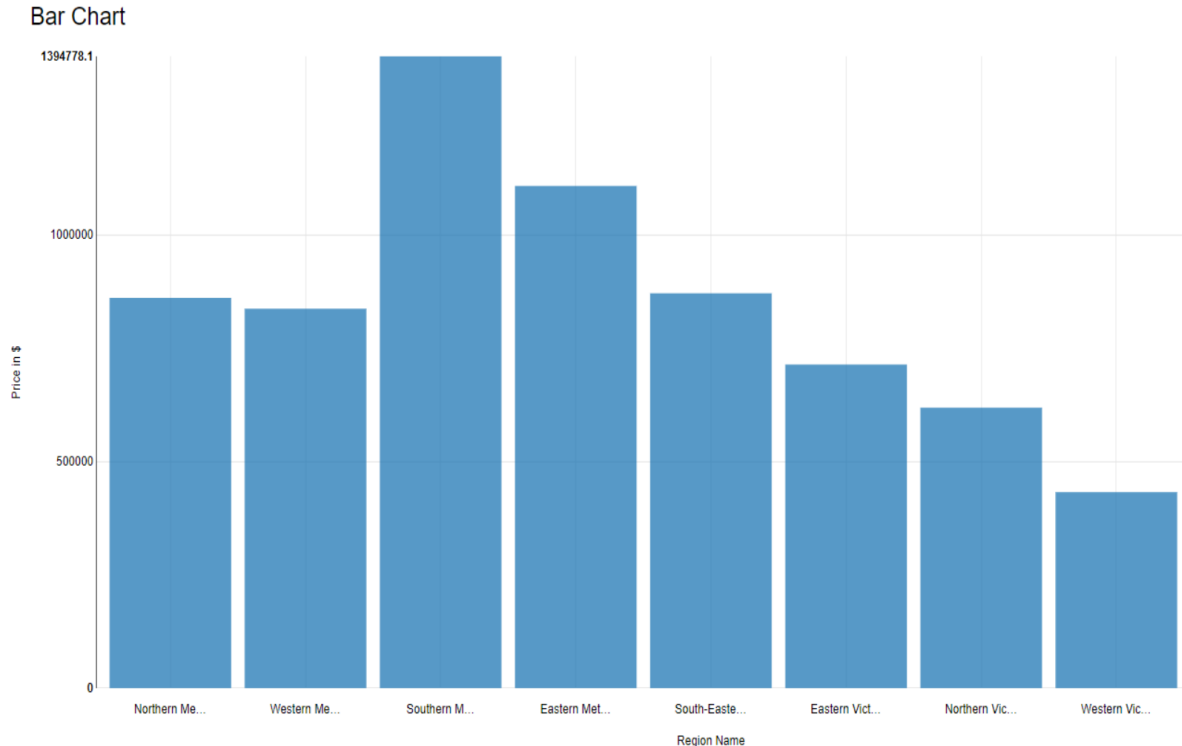


Figure 4: A distribuição dos preços dos imóveis em relação à região

Latitude, *Longitude*: Estas variáveis, por si só, não dizem nada. Como são supostas para mostrar a localização específica de uma propriedade, têm de ser combinadas para mostrar a localização da propriedade. Posteriormente, a relação entre o preço e a localização da propriedade pode ser representada. Esta relação é mostrada na figura 3.

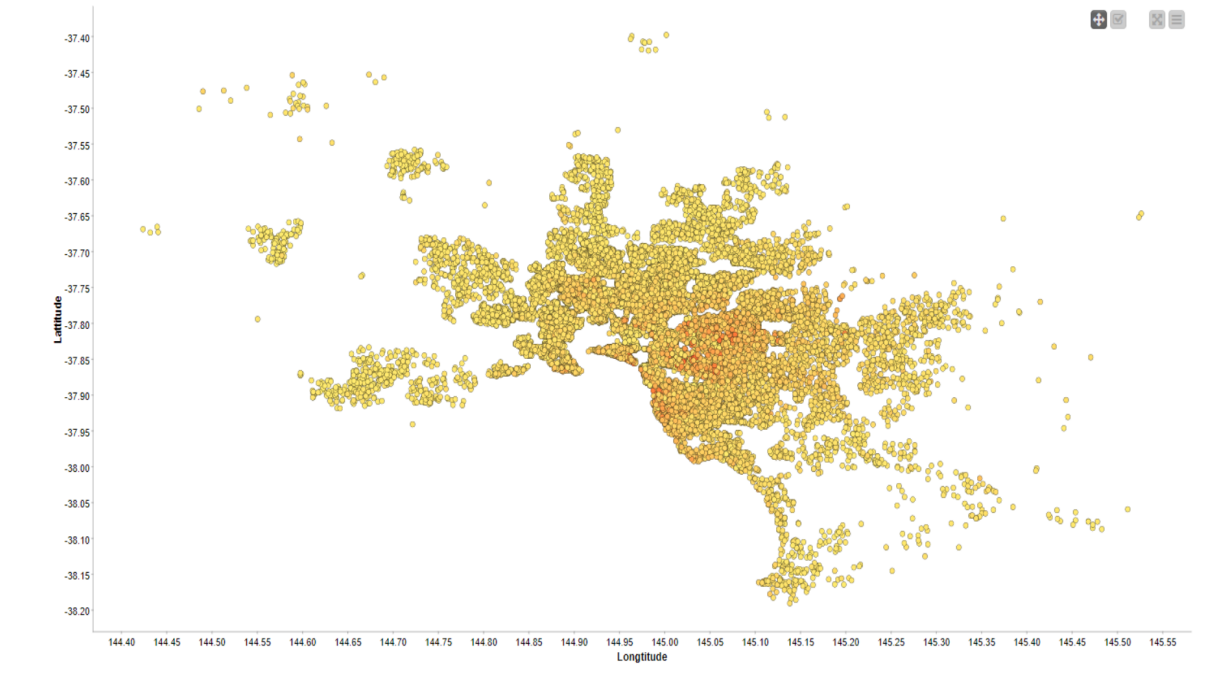


Figure 5: A distribuição dos preços das casas em relação à localização da propriedade na cidade de Melbourne

Na Figura 3, podem-se observar propriedades com preços elevados em torno do Central Business District (CBD), o que significa que, quanto mais longe do CBD, menores são os preços. Este fato é observável não apenas na cidade de Melbourne, mas também em muitas cidades ao redor do mundo. Para apoiar essa afirmação, na Figura 4 é apresentada a relação entre preço e distância do CBD. Como previsto, os preços das casas na maioria dos casos diminuem com o aumento da distância do CBD.

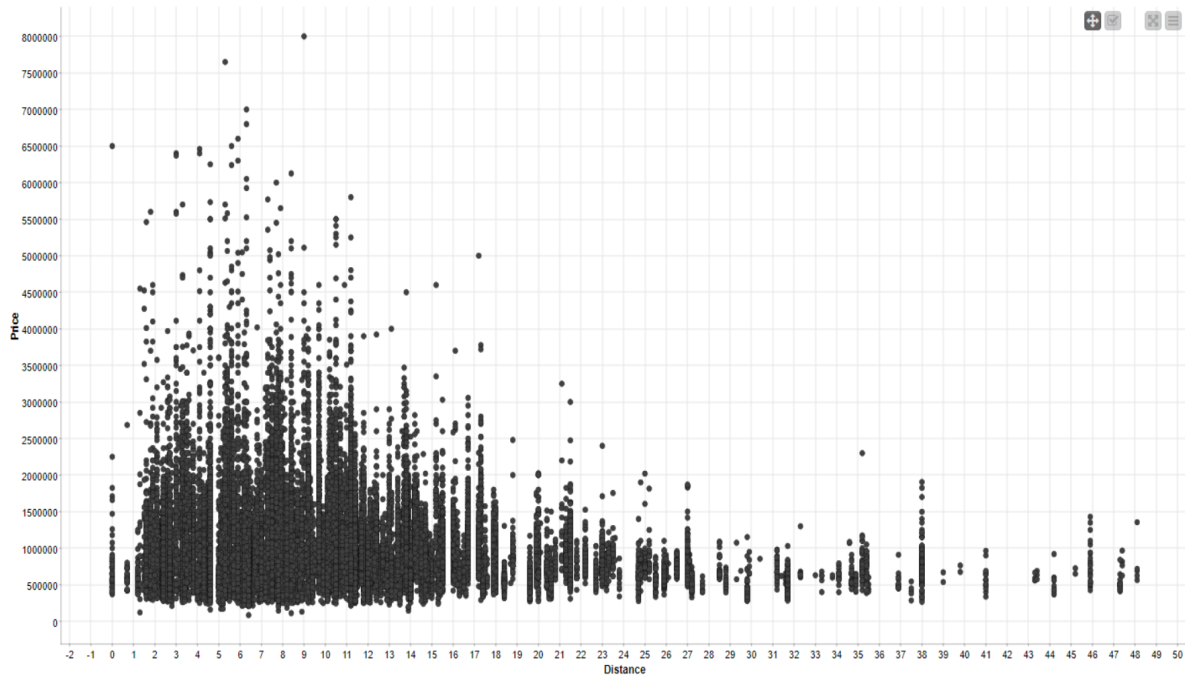


Figure 6: A distribuição dos preços das casas em relação à distância do CBD

Property Count: Esta variável fornece informações sobre a quantidade de propriedades dentro de um subúrbio e região. Além disso, o número total de casas para uma região pode consistir em três tipos diferentes. A relação entre *Type* e *Price* pode ser mostrada na imagem seguinte, pode ser observado que as casas do tipo *u* são mais as baratas.

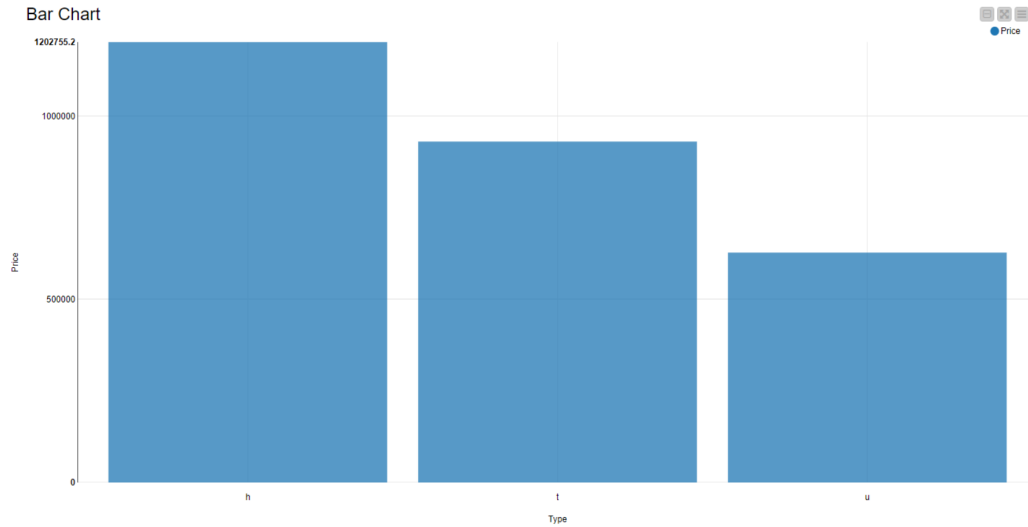


Figure 7: A distribuição dos preços dos imóveis em relação ao tipo

Mesmo que a Figura 5 forneça uma ideia sobre a distribuição de preço e tipo de imóvel na cidade de Melbourne, seria útil verificar em qual região a maior quantidade de propriedades está localizada e fazer uma relação entre o preço da habitação e o tipo de propriedades que prevalecem em uma região. Para esse propósito, a Figura 6 é apresentada.

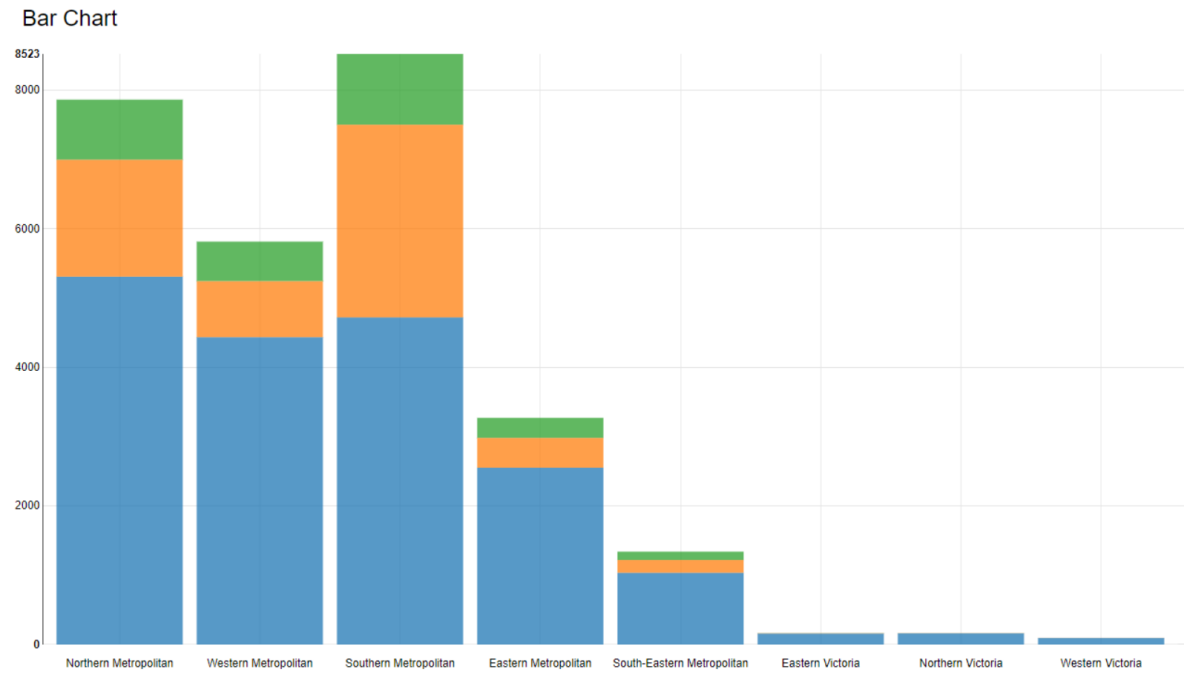


Figure 8: Quantidade de propriedades por região na cidade de Melbourne

A partir da Figura 6, é possível observar que a maioria das casas está localizada na região metropolitana do sul. Como mencionado anteriormente, ao explicar a Figura 2, a região metropolitana do sul é a mais popular com a maior quantidade de propriedades e as maiores flutuações de preços de habitação. As variáveis selecionadas após análise são: *Region*, *ConcilArea*, *Suburb*, *Latitude* e *Longitude*, *Distance*, *Type* e *PropertyCount*.

Variáveis relacionadas com o imóvel

Há 7 variáveis para analisar nesta secção: *Rooms*, *Bedroom2*, *Bathroom*, *Land Size*, *Building Area*, *Year Built*, *Car*.

De acordo com o autor do dataset fornecido, a variável *Bedroom2* fornece informações sobre o número de quartos em cada casa. No entanto, os dados são adquiridos usando várias fontes. A confiabilidade destes dados é altamente questionável. Além disso, a variável *Bedroom2* tem um número maior de missing values em comparação com a variável *Room*. Estes fatos levaram à decisão de eliminar a variável *Bedroom2* para análises futuras.

É importante lembrar que neste ponto cerca de 30% das entradas foram removidas do data set original e qualquer diminuição adicional deve ser evitada. Por isso, os missing values para as variáveis restantes (*Bathroom*, *Car*, *Landsize*, *YearBuilt* e *BuildingArea*) são interpoladas linearmente usando o nodo *Missing Value*. Para verificar se as variáveis restantes afetam o *Price* da mesma maneira (relação linear entre variáveis) usamos o nodo *Linear Correlation*.

Na figura seguinte verificamos que a correlação entre *Room* e *Bedroom2* é 0.96, o que implica linearidade entre estas variáveis, pelo que a colunas são redundantes, provando assim que a remoção da coluna *Bedroom2* é justificada.

Row ID	Rooms	Price	Bedro...	Bathro...	Car	Landsize	Buildin...	YearBuilt	Latitude	Longti...
Rooms	1	0.468	0.959	0.608	0.395	0.034	0.139	-0.002	0.021	0.087
Price	0.468	1	0.434	0.433	0.204	0.033	0.102	-0.335	-0.216	0.199
Bedroom2	0.959	0.434	1	0.605	0.396	0.034	0.137	0.008	0.019	0.091
Bathroom	0.608	0.433	0.605	1	0.306	0.038	0.129	0.186	-0.047	0.104
Car	0.395	0.204	0.396	0.306	1	0.031	0.094	0.125	0.002	0.039
Landsize	0.034	0.033	0.034	0.038	0.031	1	0.375	0.041	0.024	-0.004
BuildingArea	0.139	0.102	0.137	0.129	0.094	0.375	1	0.073	0.026	-0.011
YearBuilt	-0.002	-0.335	0.008	0.186	0.125	0.041	0.073	1	0.095	-0.022
Latitude	0.021	-0.216	0.019	-0.047	0.002	0.024	0.026	0.095	1	-0.347
Longitude	0.087	0.199	0.091	0.104	0.039	-0.004	-0.011	-0.022	-0.347	1

Figure 9: Correlação entre as variáveis relacionadas com o imóvel

Depois de uma análise mais profunda à figura acima deriva-se que a variável *Price* tem as correlações mais fortes com as seguintes variáveis: *Rooms*, *Bathrooms*, *Cars* e *BuildingArea*. *Landsize* e *BuildingArea* também possuem uma correlação com a variável de interesse.

Variáveis relacionadas com o vendedor

As últimas variáveis para analisar são *SellerG*, *Method* e *Date*. Em relação à variável *SellerG*, existem muitos vendedores diferentes e todos eles venderam propriedades de altos e baixos valores. Com isto, esta variável não pode ajudar muito na previsão de preços de imóveis.

A variável *Method* mostra um comportamento idêntico a *SellerG*. Os imóveis baratos e caros foram vendidos usando diferentes métodos, não existindo um método preferido para vender uma propriedade em específico.

Pela análise da seguinte figura também se conclui que a variável *Date* não tem um impacto significativo no *Price* pois não é possível observar nenhuma tendência.

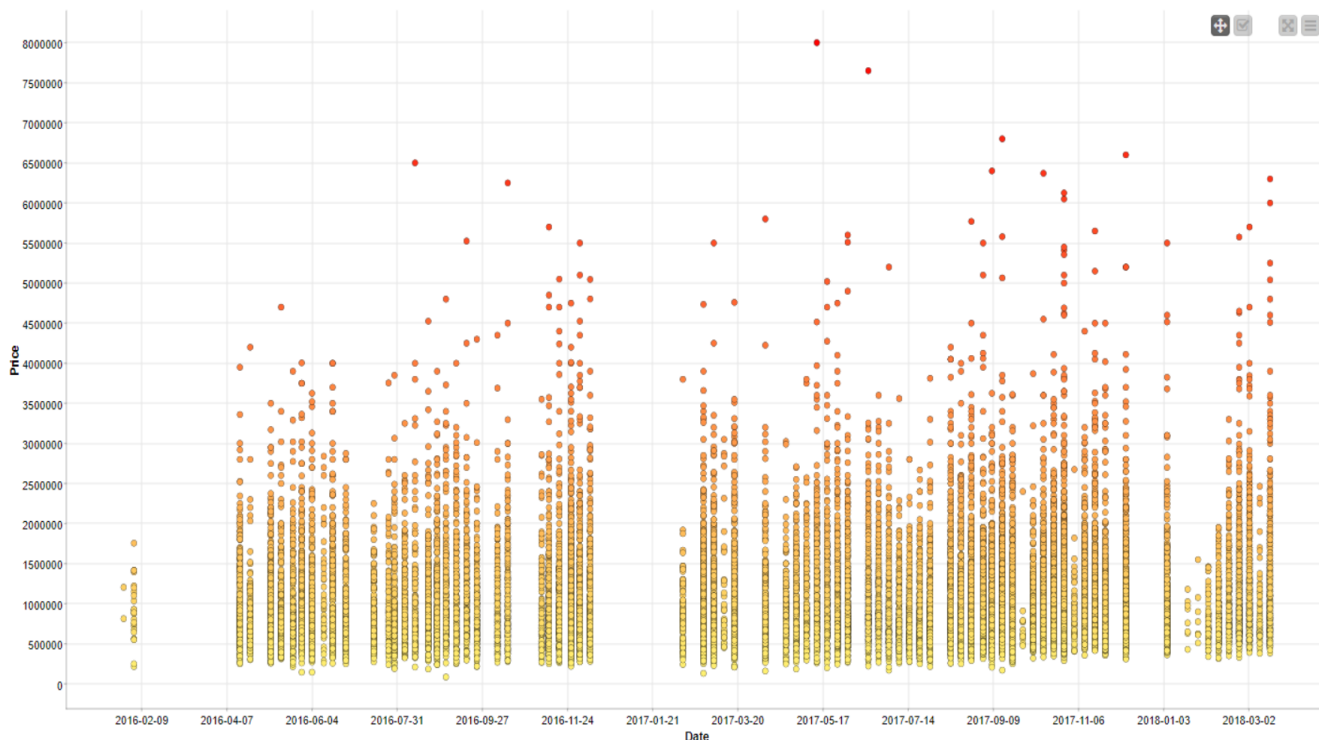


Figure 10: Casas vendidas ao longo do tempo

4.3 Modelos

4.3.1 Técnicas de aprendizagem

Para desenvolver o modelo de machine learning usado para fazer previsões, foi preciso primeiramente identificar as técnicas mais apropriadas para o problema em questão. Para isso, realizou-se uma pesquisa a fim de descobrir as técnicas de regressão mais frequentemente utilizadas na previsão de preços de imóveis. Diversas fontes destacaram o algoritmo XGBoost como a técnica mais popular para este tipo de problema.

O algoritmo Random Forest é conhecido por apresentar bons resultados em diversos problemas de regressão, por isso, também se optou por utilizar esta técnica.

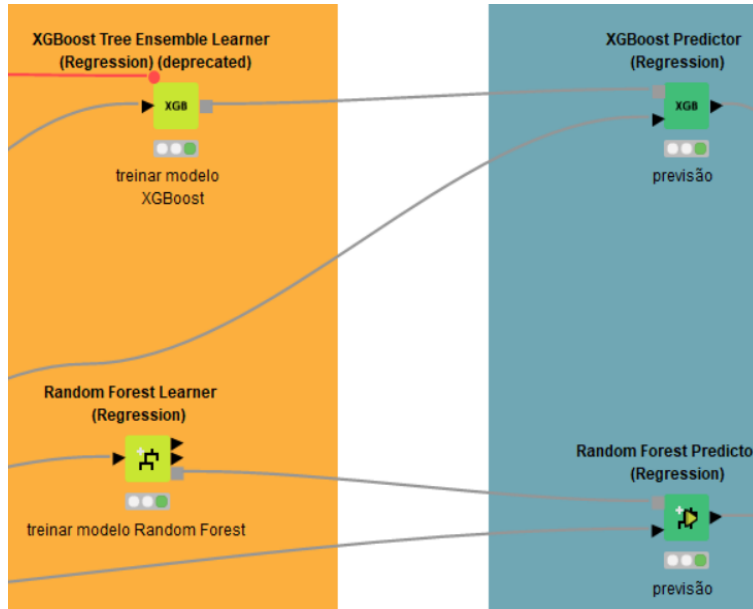


Figure 11: Treino dos modelos e previsões dos casos de teste

A fim de garantir a replicabilidade das experiências, uma random seed de valor 2023 foi utilizada durante o treinamento do modelo (observando-se que a mesma seed foi usada para ambos os datasets).

4.3.2 Tuning dos Hiperparâmetros

O algoritmo XGBoost possui vários parâmetros de treino. De modo a maximizar e ajustar os parâmetros do algoritmo XGBoost, recorreu-se aos nodos *Parameter Optimization Loop Start* e *Parameter Optimization Loop End*. Após se ter analisado os vários parâmetros, apenas se recorreu a dois deles, sendo que eram os que conferiam um maior Objective Value: ETA, Maximum Depth.

4.3.3 Partitioning

Normalmente, na comunidade científica, usa-se uma proporção de 70/30 para dividir o dataset em treinamento e teste. No entanto, essa proporção pode não ser adequada para todos os modelos. Por isso, realizaram-se alguns testes com proporções semelhantes e descobriu-se que a proporção 80/20 gerou as melhores métricas de avaliação. Sendo assim, essa foi a proporção adotada no modelo.

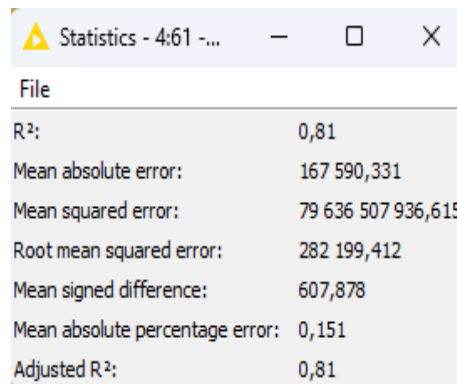
4.3.4 Cross Validation

Durante o processo de construção de um modelo de machine learning, é fundamental prestar atenção ao overfitting. Para que o modelo seja capaz de generalizar bem em dados ainda não vistos, é importante evitar o overfitting. Para atingir esse objetivo, a técnica de cross validation foi utilizada.

Para realizar as diversas partições do dataset foi utilizado o nodo X-Partitioner tendo sido escolhidas 10 validações. Devido ao grande número de linhas do dataset, decidiu-se não utilizar um número elevado de validações, visto que 10 é considerado suficiente para diversos tipos de problemas. Para manter a mesma proporção de valores da classe "price" nas diferentes validações, optou-se pelo uso do Stratified Sampling. Além disso, a random seed de valor 2023 foi utilizada. Para agregar os resultados das previsões para as várias validações, o nodo X-Aggregator foi utilizado.

4.3.5 Métricas de qualidade

A métrica definida como alvo foi a R-Squared. Todo o tratamento foi realizado com vista a maximização deste valor. Os resultados finais de várias métricas de qualidade podem ser vistos na figura seguinte.



R ² :	0,81
Mean absolute error:	167 590,331
Mean squared error:	79 636 507 936,615
Root mean squared error:	282 199,412
Mean signed difference:	607,878
Mean absolute percentage error:	0,151
Adjusted R ² :	0,81

Figure 12: Métricas de qualidade para o modelo XGBoost

As diferentes métricas foram obtidas a partir do nodo *Numeric Scorer*.

4.4 Resultados

O modelo que revelou melhores resultados foi o XGBoost e o valor da métrica definida como objetivo de maximização, o R-Squared, teve um bom resultado 0.81. A métrica MAE também teve um resultado bastante satisfatório de 167600 dólares australianos.

5 Tarefa B - Obesidade

5.1 Domínio e objetivos

O dataset analisado no presente capítulo é referente a um tema proposto pela Equipa Docente. Uma vez que o número de grupo é o 28 o tema que nos foi atribuído é relativo à Obesidade, e tem como base a análise da classificação do tipo de obesidade (sendo este o target).

Em relação a este estudo o principal objetivo é construir um modelo capaz de efetuar uma boa previsão do atributo *NObesidad*, utilizar-se-á a plataforma KNIME, junto com as suas técnicas de preparação, tratamento e visualização como mecanismo para alcançar o objetivo proposto.

5.2 Exploração, Visualização e Tratamento de dados

5.2.1 Descrição dos atributos do dataset

De modo a ter uma perceção do problema em questão, começou-se por fazer a leitura dos atributos do dataset, sendo eles:

- Variáveis Independentes
- 1. rowID: variável contínua do registo de id
- 2. Gender: variável contínua do sexo de uma pessoa
- 3. Age: variável contínua dos anos de uma pessoa
- 4. Date_of_birth: variável contínua da data de nascimento (DD/MM/YYYY)
- 5. Height: variável contínua da altura meters
- 6. Weight: variável contínua do peso em Kgs
- 7. family_history_with_overweight: variável contínua do histórico familiar de obesidade
- 8. FAVC: variável contínua do consumo frequente de comidas altamente calóricas
- 9. FCVC: variável contínua da frequência do consumo de vegetais
- 10. NCP: variável contínua do número de refeições principais - 1, 2, 3 ou 4 refeições
- 11. CAEC: variável contínua do consumo de comida entre refeições
- 12. Smoke: variável binária de se uma pessoa fuma
- 13. CH20: variável contínua do consumo de água diariamente - 1 = menos de um litro, 2 = 1–2 litros, 3 = mais de 2 litros
- 14. SCC: variável binária sobre se monitoriza o consumo de calorias - Yes/No
- 15. FAF: variável contínua da frequência da atividade física - 0 = nenhuma, 1 = 1 a 2 dias, 2 = 2 a 4 dias, 3 = 4 a 5 dias
- 16. TUE: variável contínua do tempo de uso de aparelhos tecnológicos - 0 = 0–2 horas, 1 = 3–5 horas, 2 = mais de 5 horas
- 17. CALC: variável contínua do consumo de álcool
- 18. MTRANS: variável contínua do transportação usada - Public Transportation(transportes públicos), Motorbike(mota), Bike(bicicleta), Automobile and Walking(carro e caminhar)
- Variáveis Dependentes
- 19. NObesidad: variável contínua do nível de Obesidade. **Corresponde ao target** deste estudo de classificação

5.2.2 Exploração dos dados

Feita a leitura e análise da descrição dos atributos, foi necessário ler o dataset *"obesidade.csv"*. Para tal foi usado o nodo CSV Reader.

Numa fase inicial, começou-se por efetuar a Visualização dos Dados, com o objetivo de analisar e compreender o dataset. Para tal, recorreu-se aos nodo *Data Explorer*.

Originalmente as variáveis *TUE*, *CH20*, *FAF*, *NCP*, *AGE*, *Height*, *Weight* encontravam-se em formato string portanto usando o nodo *String to Number*. Verificou-se depois que existiam valores que não faziam sentido, por exemplo, existiam valores para a variável *age* com casas decimais, o mesmo raciocínio pode ser feito para as restantes variáveis, exceto *Height* e *Weight*. Decidiu-se usando o nodo *Math Formula* arredondar esses valores de forma a podermos usar a respetiva informação.

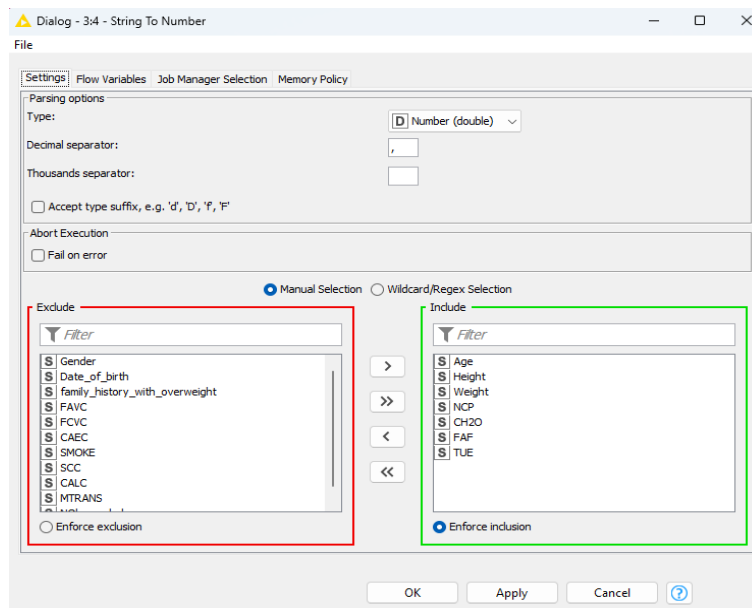


Figure 13: Configurações do nodo *String to Number*

Ao analisar os possíveis valores nominais para as variadas variáveis notou-se vários erros no data set. Por exemplo para a variável *Gender* apresenta valores redundantes, *Male* e *Man*, e também, *Female* e *Woman* portanto usou-se o nodo *Rule Engine*. Para as variáveis *CALC*, *FCVC*, *SMOKE* e *CAEC* foi usado o mesmo nodo para corrigir os erros mais evidentes.

Para a variável *FCVC* substituiu-se os valores nominais, *never*, *sometimes* e *always*, por valores numéricos, 0, 1 e 2, respetivamente, para conseguirmos tratar dos vários missing values por interpolação linear usando o nodo *Missing Value*.

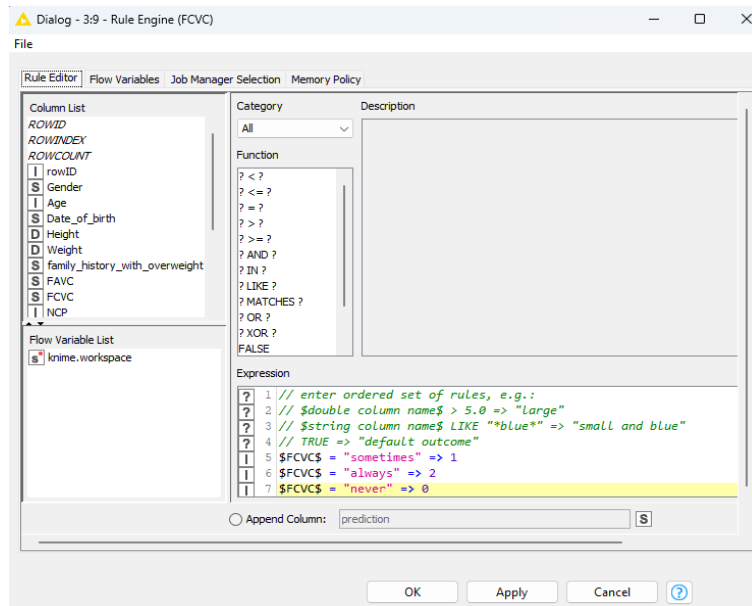


Figure 14: Configurações do nodo *Rule Engine* para tratar da variável *FCVC*

Removeu-se a coluna *rowID* pois não ajuda em nada no nosso problema de prever o tipo de obesidade. Também se extraiu o ano, mês e dia de nascimento usando os nodos *string to Date&Time* e *Extract Date&Time Fields*.

Com o objetivo de obter conhecimento adicional, utilizou-se o nodo *Partitioning* para dividir os dados em conjuntos de teste (para que o modelo possa colocar à prova o seu conhecimento) e treino (para que o modelo adquira conhecimento). Após alguns testes com diferentes proporções de divisão, concluiu-se que a proporção de 80/20 era a melhor opção e, por isso, os dados foram divididos em 80% para treino e 20% para teste. É importante destacar que todo o processo de particionamento foi realizado com a mesma *random seed* de valor 2023, a fim de garantir a possibilidade de replicar os resultados a qualquer momento.

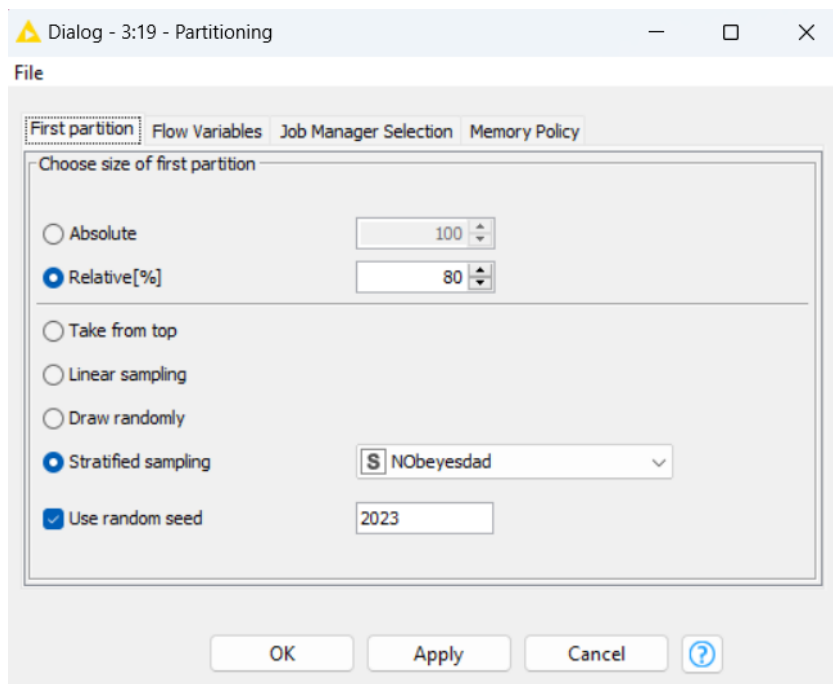


Figure 15: Configurações do nodo *Partitioning*

5.3 Modelos

Antes de construir o modelo de *machine learning*, foi necessário efetuar uma pesquisa inicial acerca de quais as técnicas mais adequadas para o presente problema.

Deste modo, os modelos foram treinados usando os seguintes nodos:

- Decision Tree Learner
- Random Forest Learner
- XGBoost Tree Ensemble Learner
- Gradiente Boosted Trees Learner

Relativamente às previsões, estas foram criadas com os seguintes nodos:

- Decision Tree Predictor
- Random Forest Predictor

- XGBoost Predictor
- Gradiente Boosted Trees Predictor

5.3.1 Tuning dos Hiperparâmetros

de modo a maximizar e ajustar os parâmetros do algoritmo XGBoost, recorreu-se aos nodos **Parameter Optimization, Loop Start** e **Parameter Optimization Loop End**, tendo sido ajustados alguns parâmetros que conferiram melhorias significativas no modelo em questão. Apesar de este tema ser desenvolvido com mais detalhe na Tarefa A, pelo que se explica em seguida os parâmetros que foram alterados para aumentar a performance. Após se ter analisado os vários parâmetros, apenas se recorreu a dois deles, sendo que eram os que conferiam um maior *Objective Value*:

- ETA
- Maximum Depth

5.3.2 Decision Tree

Decision Tree é um tipo de *supervised machine learning* usado para categorizar ou prever o valor de uma variável, aprendendo regras de decisão simples inferidas a partir de dados anteriores (dados de treinamento).

Uma *Decision Tree* é um grafo hierarquizado (árvore) em que:

- Cada **ramo** representa a **seleção entre um conjunto de alternativas**
- Cada **folha** representa uma **decisão**

Dada uma árvore de decisão treinada, o **processo de decisão** desenvolve-se do seguinte modo:

1. Começar do nodo correspondente ao atributo "raiz"
2. Identificar o valor do atributo
3. Seguir pelo ramo correspondente ao valor identificado
4. Alcançar o nodo relativo ao ramo percorrido
5. Voltar a 2. até que o nodo seja uma folha
6. O nodo alcançado indica a decisão para o problema

5.3.3 Gradient Boosted Trees Predictor

O algoritmo *Gradient Boosted Trees*, também conhecido como GBT, deriva do algoritmo XGBoost, que também foi utilizado no outro dataset. Desta forma, Gradient Boosted Trees aprende com o objetivo de classificação, usando árvores de regressão muito rasas e uma forma especial de reforço para consumir um conjunto de árvores.

Ao recorrer ao *Gradient Boosted Trees*, este método constrói uma árvore de cada vez, onde cada árvore ajuda a corrigir os erros que possam ter sido cometidos pela árvore que foi treinada anteriormente.

Para além das razões referidas anteriormente, o algoritmo é também muito eficaz em casos de deteção de anomalias e informação bastante desbalanceada, sendo frequentemente utilizado em situações de teste de ADN, transações de cartões de crédito ou cibersegurança.

5.4 Conclusão e análise dos resultados obtidos

Correct classified: 394	Wrong classified: 28
Accuracy: 93,365%	Error: 6,635%
Cohen's kappa (κ): 0,922%	

Figure 16: Resultado ao usar a Decision Tree

Correct classified: 400	Wrong classified: 23
Accuracy: 94,563%	Error: 5,437%
Cohen's kappa (κ): 0,937%	

Figure 17: Resultado ao usar Random Forest

Correct classified: 408	Wrong classified: 15
Accuracy: 96,454%	Error: 3,546%
Cohen's kappa (κ): 0,959%	

Figure 18: Resultado ao usar XGBoost

Correct classified: 412	Wrong classified: 11
Accuracy: 97,4%	Error: 2,6%
Cohen's kappa (κ): 0,97%	

Figure 19: Resultado ao usar Gradient Boost

6 Conclusão

Ao longo do trabalho realizado, ficou claro que o tratamento adequado dos dados é essencial para obter modelos de previsão precisos. É importante ter em mente que todas as etapas do processo de construção do modelo estão inter-relacionadas e igualmente importantes. Além disso, foi possível aprimorar o conhecimento sobre quais tipos de tratamentos devem ser aplicados para determinados algoritmos de machine learning.

Este trabalho apresentou desafios na identificação das técnicas de tratamento de dados mais efetivas para o modelo, bem como na compreensão de quando evitar certas técnicas de tratamento. Ambos os aspectos são fundamentais para garantir a qualidade do modelo de machine learning e exigiram uma análise cuidadosa das relações entre os dados, o contexto e os algoritmos utilizados.

O trabalho em questão proporcionou a oportunidade de expandir os conhecimentos na área de machine learning, aplicando as técnicas e conhecimentos adquiridos nas aulas teóricas e práticas em problemas reais de regressão ou classificação. A escolha do conjunto de dados *Melbourne Housing Market* para o trabalho de regressão permitiu ao grupo diversificar os seus conhecimentos, impactando positivamente a aprendizagem e prática dos conteúdos ensinados.