



UNIVERSIDADE DO MINHO

DEPARTAMENTO DE INFORMÁTICA

Aprendizagem e Decisão Inteligentes

3^o ano - LEI

Trabalho Prático

Relatório de Desenvolvimento

Grupo 11

Ariana Lousada (a87998)

Rui Armada (a90468)

João Carvalho (a93166)

Tiago Sousa (a67674)

6 de maio de 2022

Resumo

O presente documento descreve sucintamente os objetos de avaliação e de análise ao longo de um projeto de análise inserido na unidade curricular Aprendizagem e Decisão Inteligentes. Este projeto teve como principais objetivos a análise, tratamento e previsão de dados de dois *datasets* distintos. Ao longo dos vários capítulos e secções presentes no relatório são expostas todas as decisões tomadas por parte da equipa de trabalho relativas aos métodos utilizados para o alcance do objetivo do projeto.

Conteúdo

1	Introdução	2
2	Dataset 1 : Valor de habitação nos EUA	3
2.1	Análise do dataset	3
2.2	Tratamento de dados	5
2.3	Previsão da variável alvo	6
2.3.1	Cross-validation	6
2.3.2	Tuning	8
2.3.3	Feature Selection	10
2.4	Modelos desenvolvidos e resultados obtidos	10
2.4.1	Scorer V1	10
2.4.2	Scorer V2	11
2.4.3	Versão final	11
2.4.4	Resultados obtidos	12
3	Dataset 2 : Tráfego de veículos na cidade do Porto	13
3.1	Análise do dataset	13
3.2	Tratamento de dados	15
3.3	Previsão da variável alvo	18
3.3.1	Cross-validation	18
3.3.2	Tuning	19
3.3.3	Feature Selection	20
3.4	Modelos desenvolvidos e resultados obtidos	20
3.4.1	Scorer V1	20
3.4.2	Scorer V2	20
3.4.3	Versão final	21
3.4.4	Resultados obtidos	21
4	Conclusão	22
5	Referências	23
6	Anexos	24

Capítulo 1

Introdução

No âmbito da Unidade Curricular de Aprendizagem e Decisão Inteligentes, foi proposta a realização de um trabalho prático cujo objetivo consistiu no desenvolvimento de um modelo de *Machine Learning*, através da utilização de modelos de aprendizagem que foram abordados ao longo do semestre em ambiente *KNIME*.

Para a construção de uma solução para o presente trabalho prático, foram utilizados dois *datasets*.

O primeiro *dataset* foi fornecido pela equipa docente da unidade curricular e contém dados referentes ao valor de habitação nos EUA. Este *dataset* contém informação sobre várias habitações e como as suas qualidades influenciam o seu preço de compra. O objetivo para este *dataset* consistiu em desenvolver modelos de previsão do preço (de venda) das habitações.

Relativamente ao segundo *dataset*, foi necessário uma consulta e análise de vários *datasets* por parte da equipa de trabalho. Após várias pesquisas, foi escolhido um *dataset* que contém dados referentes ao tráfego de veículos na cidade de Porto durante um período superior a 1 ano. Este *dataset* vai cobrir o período entre 24 de Julho de 2018 até 02 de Outubro de 2019. O objetivo neste *dataset* consiste em elaborar modelos de previsão da intensidade do trânsito na cidade do Porto.

Capítulo 2

Dataset 1 : Valor de habitação nos EUA

2.1 Análise do dataset

Primeiramente, é realizada uma análise cuidada e extensiva do *dataset*. É de notar que o *dataset* se refere a dados relativos a preços de habitações nos EUA. Feito isto, verificou-se que o *dataset* inclui os seguintes atributos.

Atributo	Descrição
Avg. Area Income	média de rendimento dos residentes da cidade onde a casa se localiza
Avg. Area House Age	média da idade das casas na mesma cidade
Avg. Area Number of Rooms	média do número de quartos das casas da mesma cidade
Avg. Area Number of Bedrooms	média do número de quartos de casas da mesma cidade
Area Population	população da cidade onde a casa se localiza
Address	morada da casa
Price	preço de venda da casa

Tabela 2.1: Atributos do *dataset*

O objetivo deste *dataset* é descobrir qual é o atributo que será necessário utilizar para se prever o preço a que as casas são vendidas com a melhor *accuracy* possível. Para este efeito, chegou-se à conclusão que o melhor atributo, para atingir este objetivo, é o atributo **Price**.

Posteriormente foi realizado um estudo exaustivo dos atributos que possuem maior correlação com o atributo *target* do *dataset*, o **Price**. Para este efeito, utilizou-se o **Linear Correlation** para construir a matriz de correlações do *dataset*.

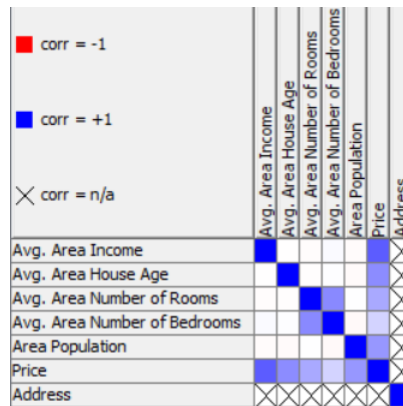


Figura 2.1: Matriz de Correlação do *Dataset1*

Feito isto, foi utilizado o nodo **Line Chart (JFreeChart)** para elaborar gráficos de *plot line* de forma a observar as correlações entre as variáveis.

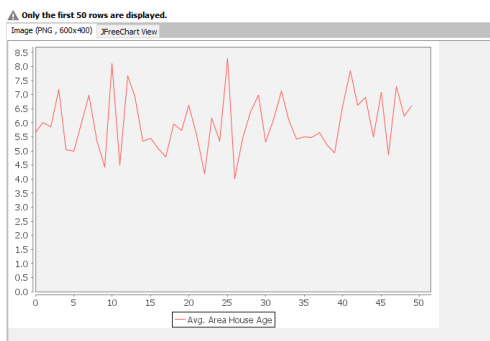


Figura 2.2: Plot Line referente à House Age

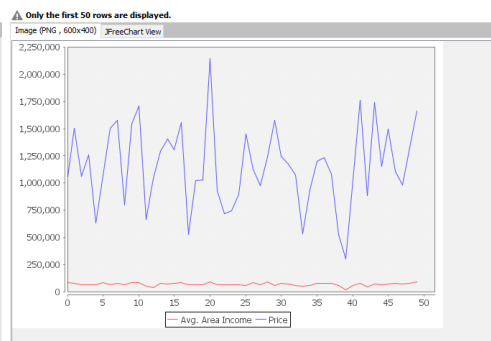


Figura 2.3: Plot Line referente ao Area Income

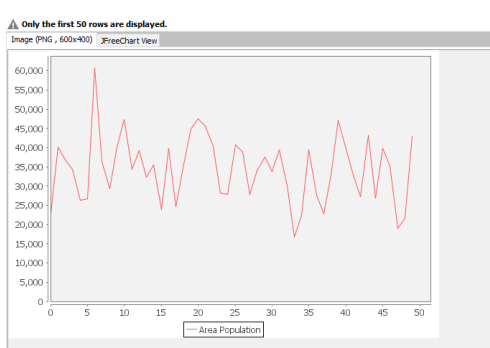


Figura 2.4: Plot Line referente à Area Pop.

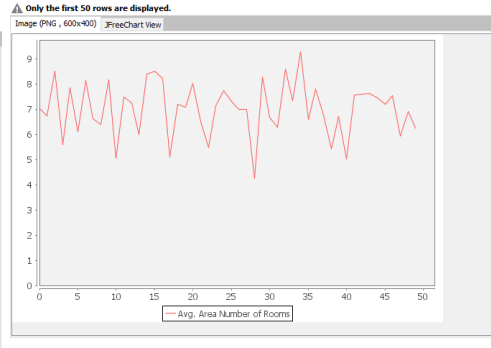


Figura 2.5: Plot Line referente ao N^o of Bedrooms

2.2 Tratamento de dados

Após uma análise ao conjunto de dados começou-se a construção do *workflow* e subsequente tratamento do *dataset* para a previsão do *target*. Primeiramente, foi utilizado o nodo **Missing Values** para resolver todos os *missing values* presentes no *dataset*. No caso de *missing values* do tipo *string*, são transformados para o valor **NULL**, enquanto que no caso de inteiros se alteram para o valor **0.0**.

Seguidamente, são resolvidos os *outliers* do *dataset*. Para este efeito, é utilizado o nodo **Numeric Outliers** de forma a resolver quaisquer valores extremos presentes nos dados do *dataset*.

Para finalizar, são filtradas todas as colunas desnecessárias para a previsão do *target* utilizando o **Column Filter**.

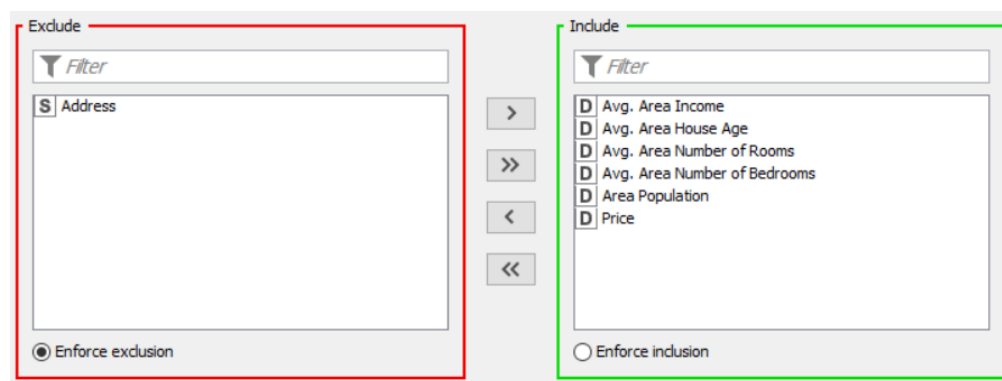


Figura 2.6: Filtragem das colunas do *dataset*

A sequência de passos, referida anteriormente, encontra-se ilustrada na seguinte figura em ambiente KNIME.

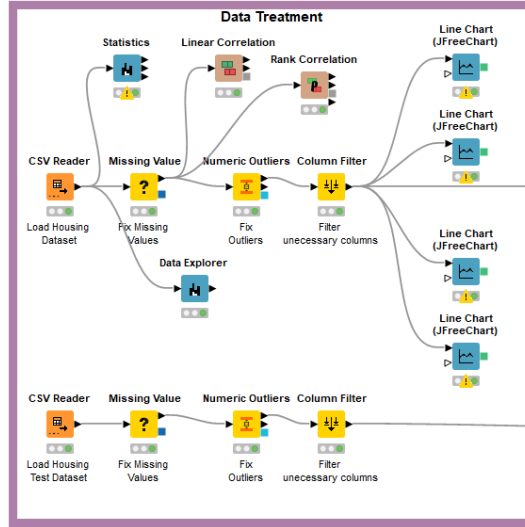


Figura 2.7: Tratamento de dados do *dataset*

2.3 Previsão da variável alvo

Feito o tratamento de dados, foi decidido aplicar uma **Linear Regression** juntamente com o método de **Cross-validation**. Desta forma foi construída uma secção do *workflow* responsável pela otimização das variáveis do modelo da *Linear Regression*.

2.3.1 Cross-validation

Cross-validation[1] é uma técnica de validação de modelos de *Machine Learning* que tem como objetivo construir uma métrica precisa do desempenho do modelo.

Mais precisamente, consiste em dividir o conjunto de dados em ***k-folds***. Em cada execução, $k-1$ *folds* são usados para o treino e 1 *fold* é usado como teste. O processo repete-se até todos os *folds* tenham sido usados como teste. A métrica de erro final é baseada no valor médio de todas as métricas de erro.

Um dos fatores que levou à utilização de *Cross-validation* foi o facto de possibilitar a diminuição de *overfitting*. *Overfitting* consiste na produção de uma análise que corresponde, aproximadamente, a um determinado conjunto de dados fazendo com que exista a possibilidade de falhar ao se ajustar a dados adicionais ou prever valores futuros sem fiabilidade.

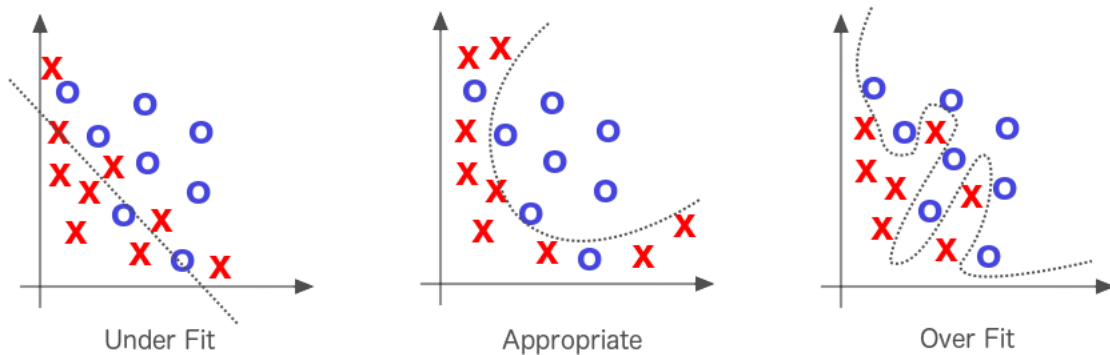


Figura 2.8: Tipos de modelos

Sendo assim, o *overfitting* reflete que o modelo memoriza e modela demasiado o *training set*, originando uma previsão incorreta caso se utilize outro *dataset* no modelo. Assim sendo, para a construção deste modelo optou-se por atribuir um valor de 10 ao *k-fold*.

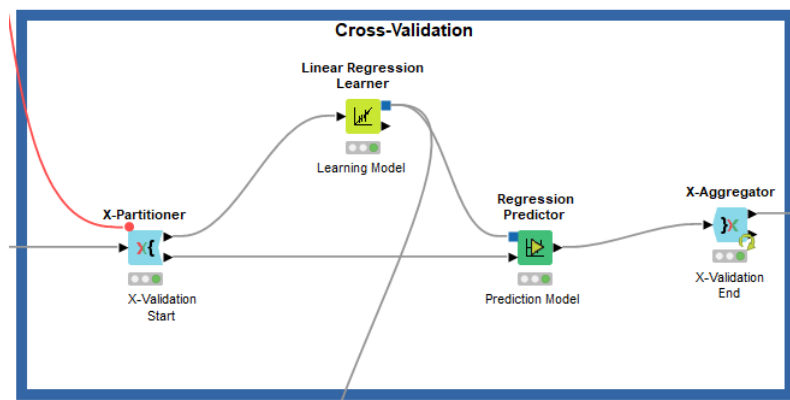


Figura 2.9: *Cross-validation*

Foram utilizados os nodos **X-Partitioner** e **X-Aggregator** para permitir a aplicação desta técnica. Numa explicação mais detalhada, o primeiro nodo faz uma repartição do *dataset* pelo *k-fold* definido como *sampling* aleatório. Já o segundo nodo serve para definir a coluna "alvo" e a coluna da previsão, retornando a tabela das previsões já com a percentagem de erro de cada *fold*.

Standard settings | Flow Variables | Memory Policy

Number of validations: 10

Linear sampling: ☐

Random sampling: ☒

Stratified sampling: ☐

Class column: D Price

Random seed: ☐ 0

Leave-one-out: ☐

Figura 2.10: Definição do nodo *X-Partitioner*

Standard settings | Flow Variables | Memory Policy

Target column: D Price

Prediction column: D HOUSING_PRICE

☐ Add column with fold id

Figura 2.11: Definição do nodo *X-Aggregator*

2.3.2 Tuning

Aqui, foi aplicado um *tuning*[2] de parâmetros nominais e de parâmetros numéricos respetivamente.

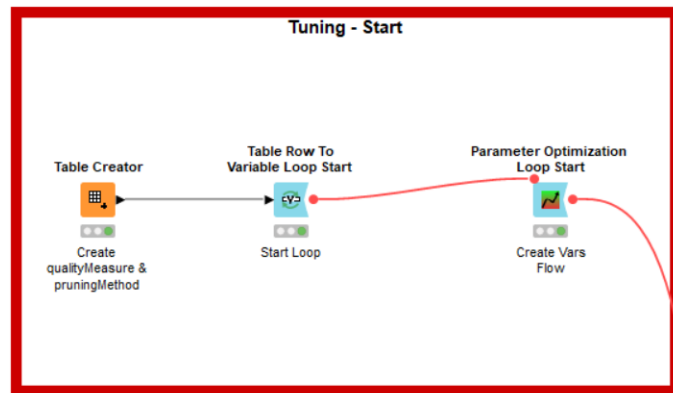


Figura 2.12: *Tuning Start*

Inicialmente foram definidos os parâmetros nominais da tabela a ser criada, através do **Table Creator**, com o intuito de otimizar a variável *Quality Measure*. Para tal foram definidos os parâmetros *Information Gain*, *Information Gain Ratio* e, por último, o *Gini Index*.

Table Creator Settings Flow Variables Memory Policy			
Input line:			
	\$ column1		
Row0	InformationGain		
Row1	InformationG...		
Row2	Gini		
Row3			

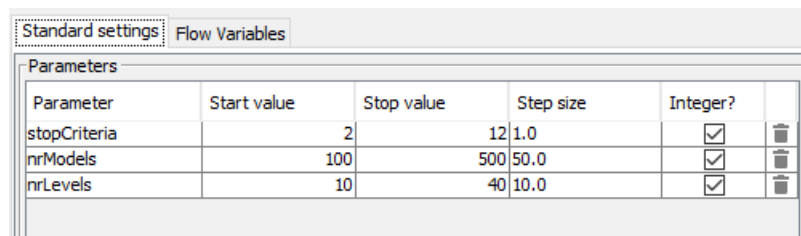
Figura 2.13: Definições do nodo *Table Creator*

Após a introdução dos parâmetros numéricos, foi utilizado o **Parameter Optimization Loop Start** para a criação de três variáveis distintas:

Variável	Descrição
nrModels	corresponde ao número máximo de modelos a usar
nrLevels	corresponde ao número máximo de níveis da árvore de decisão
stopCriteria	corresponde a uma variável que serve de critério de paragem

Tabela 2.2: Variáveis do *Parameter Optimization Loop Start*

Também foi decidido que seria utilizada uma estratégia de **Brute Force** para a previsão da variável alvo.



Parameter	Start value	Stop value	Step size	Integer?
stopCriteria	2	12	1.0	<input checked="" type="checkbox"/>
nrModels	100	500	50.0	<input checked="" type="checkbox"/>
nrLevels	10	40	10.0	<input checked="" type="checkbox"/>

Figura 2.14: Definições do nodo *Parameter Optimization Loop Start*

Já na parte final do *tuning*, de maneira a que seja realizado o cálculo da **accuracy**, optou-se por utilizar o **Parameter Optimization Loop End** onde foi selecionada a função objetivo que se quer maximizar. Neste caso foi selecionada a opção *maximized* para o R^2 . Estes dados são, por fim, passados ao nodo **Variable Loop End** onde foram definidos os parâmetros a otimizar. Ainda foi colocado um **GroupBy** com o objetivo de calcular a média final do valor da função objetivo, ou seja, calcular o valor médio da *accuracy* do *output* gerado. Este *GroupBy* foi apenas utilizado para uma avaliação do *output* por parte da equipa de desenvolvimento.

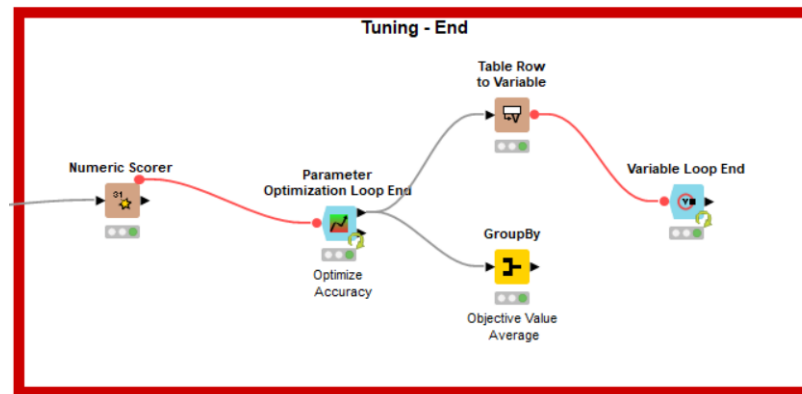


Figura 2.15: *Tuning End*

2.3.3 Feature Selection

De modo a se otimizar as escolhas das *features* a utilizar para gerar os melhores valores de *accuracy*, foi decidido criar um *workflow* que no qual é feita uma seleção das melhores *features* para utilizar na previsão do *target*.

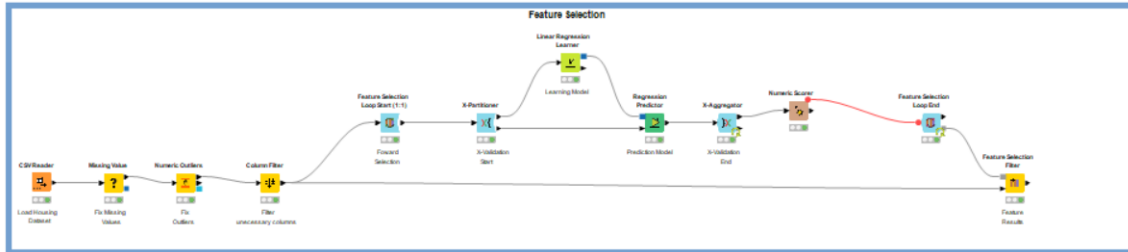


Figura 2.16: *Feature Selection Workflow*

O processo é semelhante ao que já foi descrito. Embora não haja *tuning*, utilizou-se o nodo **Feature Selection Filter** de modo a obter a seleção ótima tal como no seguinte exemplo.

Column Selection Flow Variables Memory Policy		
<input checked="" type="checkbox"/> Include static columns <input type="checkbox"/> Select features manually <input checked="" type="radio"/> Select best score <input type="radio"/> Select features automatically by score threshold Prediction score threshold: <input type="text" value="0.5"/>		
Optimization Criteria: <i>The score is being maximized.</i>		
R ²	Nr. of features	
0.916	4	D Avg. Area Income
0.916	5	D Avg. Area House Age
0.7	4	D Avg. Area Number of Rooms
0.638	3	D Avg. Area Number of Bedrooms
0.493	4	D Area Population
0.409	3	D Price
0.408	1	
0.378	2	

Figura 2.17: Exemplo de uma *Feature Selection*

2.4 Modelos desenvolvidos e resultados obtidos

2.4.1 Scorer V1

Numa primeira versão do modelo desenvolvido, foi introduzido todo o tratamento de dados que foi mencionado anteriormente sendo realizado um **Partitioning** dos dados de treino a serem usados

na aprendizagem da **Random Forest Learner (Regression)**. Após a aprendizagem, a previsão é qualificada pelo **Numeric Scorer**.



Figura 2.18: Primeira versão do modelo de aprendizagem

2.4.2 Scorer V2

Nesta versão do modelo de aprendizagem já são introduzidos os conceitos de *tuning* e *cross-validation* referidos na secção anterior do relatório. Em adição foi usado o **Linear Regression** que permitiu uma melhor análise para escolher que modelo de aprendizagem seria o mais indicado e qual dos modelos nos daria uma melhor *accuracy*. Neste caso o **Linear Regression** mostrou-se ser o melhor dos dois.

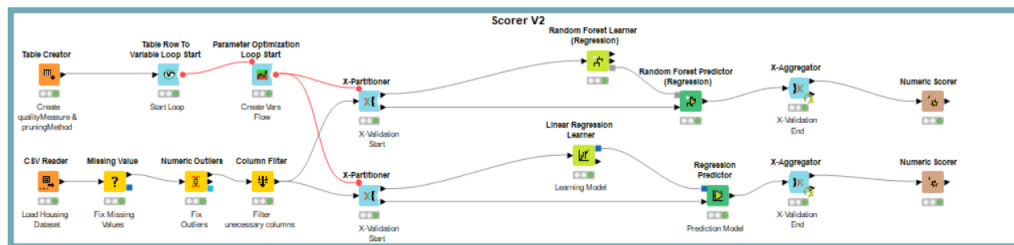


Figura 2.19: Segunda versão do modelo de aprendizagem

2.4.3 Versão final

Referente à versão final do modelo é importante referir que foi introduzido o tratamento de dados para os dados de teste, o que implicou a introdução de um novo **Predictor** para estes dados. Também foram inseridos o *tuning* e o conceito de *Cross-validation* aplicando tudo o que foi referido na secção anterior do relatório. Posteriormente, também foi inserida uma secção do *workflow* onde os dados são formatados num ficheiro CSV, com a filtragem das colunas bem como a introdução da coluna *RowID*.

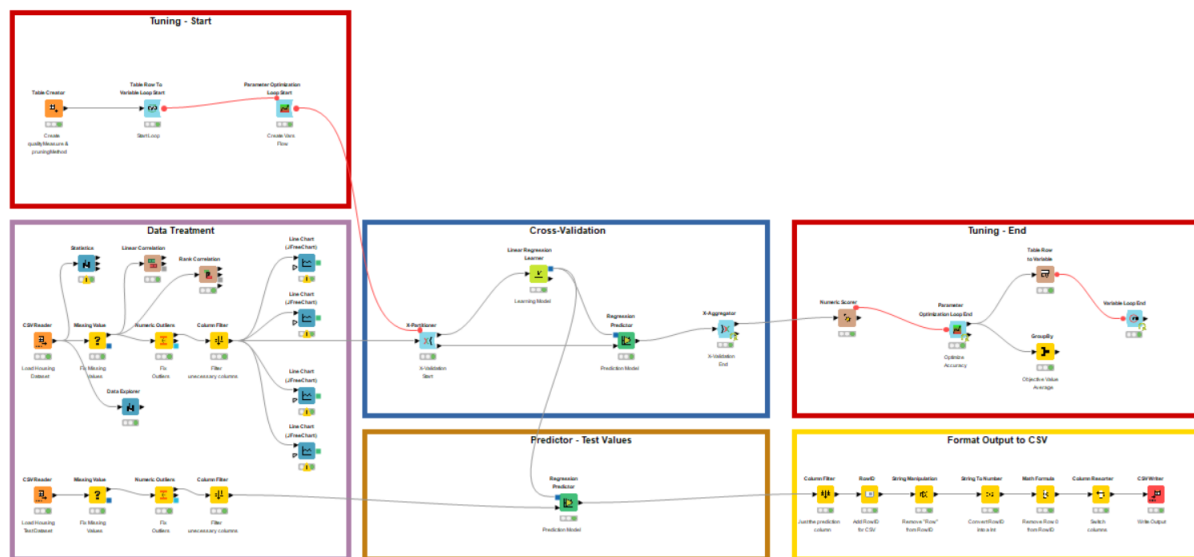


Figura 2.20: Versão final do modelo de aprendizagem

2.4.4 Resultados obtidos

No que toca aos resultados obtidos no primeiro modelo desenvolvido, estes encontram-se na gama dos 88% de *accuracy* na previsão da variável alvo.

Numa segunda fase do modelo, foram comparadas as performances do **Random Forest Learner (Regression)** e do **Linear Regression** de forma a determinar qual dos nodos iria fornecer melhores resultados, sendo que o **Linear Regression** produziu um valor de *accuracy* mais elevado.

Feitas todas as otimizações ao modelo de aprendizagem, é seguro afirmar que os resultados obtidos podem ser considerados satisfatórios, uma vez que o valor de R^2 se encontra com um valor de 91.6% fornecendo uma melhor *accuracy* na previsão do preço de habitações nos EUA.

Row ID	I stopCri...	I nrModels	I nrLevels	D Objective value
Row0	6	450	20	0.916

Figura 2.21: Resultados do modelo de aprendizagem

Capítulo 3

Dataset 2 : Tráfego de veículos na cidade do Porto

3.1 Análise do dataset

Como no *dataset* anterior, foi realizada uma análise extensiva do conjunto de dados. É importante referir que este *dataset* irá conter dados cruciais referentes ao trânsito na cidade do Porto durante um determinado período de tempo. Depois desta análise, verificou-se que o *dataset* inclui os seguintes atributos:

Atributo	Descrição
city_name	nome da cidade em questão
record_date	<i>timestamp</i> do registo
average_speed_diff	diferença de velocidade
average_free_flow_speed	média da velocidade máxima que o carro pode atingir
average_time_diff	média da diferença do tempo que demora a percorrer ruas
average_free_flow_time	média do tempo que demora a percorrer ruas quando não há trânsito
luminosity	nível de luminosidade que se verifica na cidade
average_temperature	média da temperatura, para o <i>record_date</i> , na cidade
average_atmosp_pressure	média da pressão atmosférica no <i>record_date</i>
average_humidity	média da humidade no <i>record_date</i>
average_wind_speed	média da velocidade do vento no <i>record_date</i>
average_cloudiness	média da percentagem de nuvens no <i>record_date</i>
average_precipitation	valor médio de precipitação no <i>record_date</i>
average_rain	avaliação quantitativa da precipitação no <i>record_date</i>

Tabela 3.1: Atributos do *dataset*

O objetivo deste *dataset* é descobrir qual é o atributo que será necessário utilizar para se conseguir prever a intensidade do trânsito num dado momento com a maior *accuracy* possível. Como tal, chegou-se à conclusão que o melhor atributo, para atingir este objetivo, é o ***average_speed_diff***.

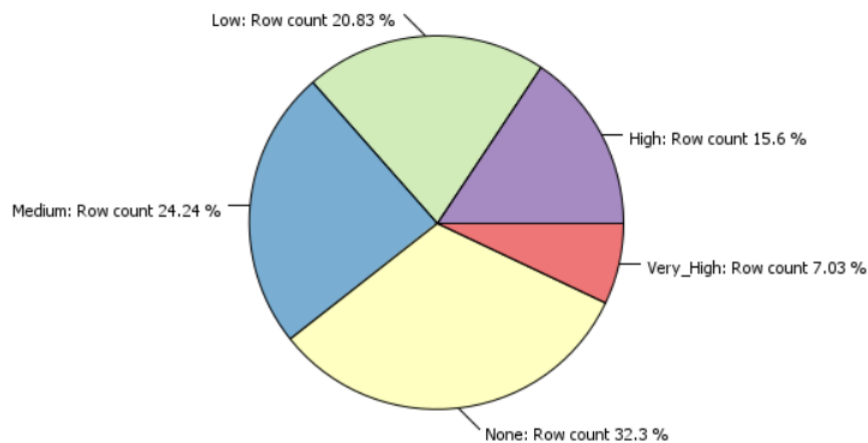


Figura 3.1: Distribuição do atributo *average_speed_diff*

Como se pode observar no gráfico anterior, o atributo é composto pelos parâmetros **None**, **Low**, **Medium**, **High** e **Very_High** que definem a intensidade do trânsito. Estes parâmetros constituem a escala que será utilizada nas previsões.

Através do gráfico pode-se elaborar uma ideia do trânsito que existe ao longo do dia. Por exemplo, a maior percentagem corresponde a **None**, cerca de 32%. Com isto, pode-se concluir que existem bastantes períodos do dia em que não existe qualquer tipo de trânsito. Já no caso do **Very_High**, com percentagem de 7%, conclui-se que é referente a períodos do dia como no início da manhã e o final de tarde nos quais existe uma intensidade de trânsito mais elevada.

Os restantes valores levam a concluir que o trânsito ao longo do dia tem um tráfego mediano, visto que existem percentagens altas tanto para o **Low** como para o **Medium**.

Após a escolha do atributo mais ideal para prever a intensidade do trânsito, foi feito um estudo exaustivo de quais atributos possuem maior correlação com o atributo *target*, o *average_Speed_Diff*. Dito isto, foi utilizado o nodo **Linear Correlation** que permitiu a visualização da matriz de correlações do *dataset*.

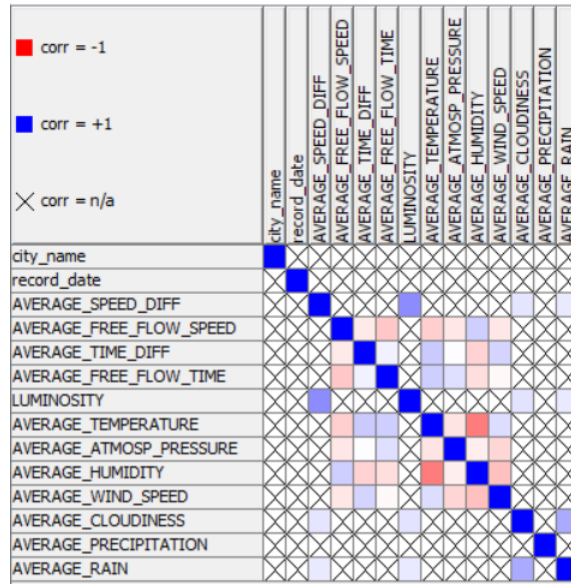


Figura 3.2: Matriz de Correlação do *Dataset2*

Pode-se observar pelo gráfico anterior, que variáveis como o *city_name* não possuem qualquer correlação com outros atributos do *dataset* tornando-se, assim, irrelevante para a previsão da intensidade do trânsito.

3.2 Tratamento de dados

Como referido anteriormente, para a construção deste modelo de *Machine Learning*, foi necessário realizar uma análise crítica ao conjunto de dados fornecidos pelo *dataset*, determinando quais seriam os dados a utilizar, quais teriam de ser alterados e quais não seriam utilizados.

Para isto, foi desenvolvida uma secção do *workflow* onde se poderia visualizar todas as estatísticas referentes ao *dataset* em questão. Foi usado o **Color Manager** juntamente com o **Pie Chart (local)** de forma a facilitar a análise da variável *target* e elaborar o gráfico referido na secção anterior.

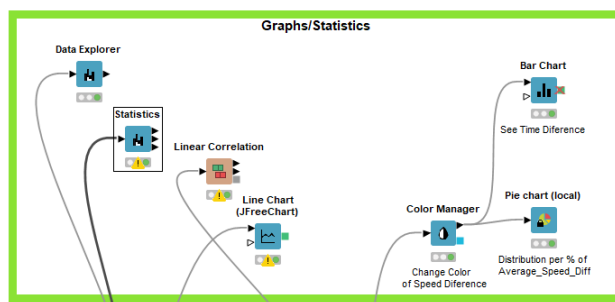


Figura 3.3: Gráficos e Estatísticas

Foi ainda utilizado um **Bar Chart** para construir uma estatística do *Average_Speed_Diff* pelos seus atributos.

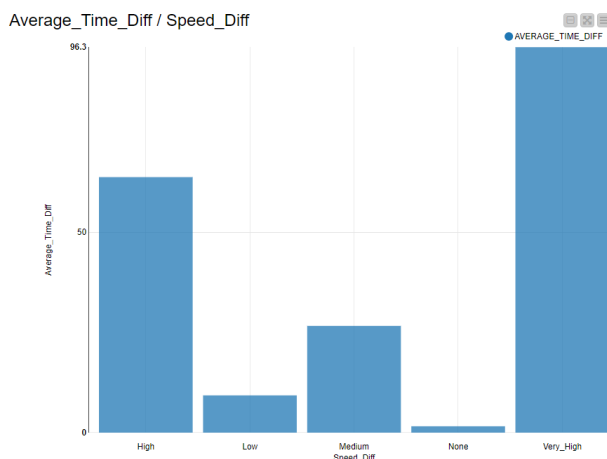


Figura 3.4: Distribuição do *Average_Speed_Diff*

Feita esta análise, começou-se a fase de tratamento do *dataset*. Primeiramente, procedeu-se a transformar a *string* "*record_date*" no tipo **DateTime** para, posteriormente, ser retirado o dia da semana, o mês, o ano e a hora do dia do *DateTime*. Para este efeito, foi utilizado o **Extract DateTime Fields** para dividir as componentes da data em colunas distintas.

Seguidamente, foi utilizado o **String Manipulation** para transformar os valores da coluna **Luminosity**, que são *strings*, em valores inteiros e alterou-se o tipo da coluna utilizando o nodo **String to Number**.

String	Inteiro
Dark	0
Light	1
Low_Light	2

Tabela 3.2: Transformação dos valores da coluna *Luminosity*

Também foi utilizado o nodo **Missing Values** para alterar os *missing values* nas tabelas "**Average.Cloudiness**" e "**Average.Rain**" para **NULL**. Em Adição, foi realizado um tratamento dos *outliers* presentes no *dataset*. Para este efeito foi usado o nodo **Numeric Outliers** para resolver o problema. Feito isto, através do **Math Formula**, é realizado o cálculo da **distância** percorrida que é utilizada para calcular a **velocidade média** de cada veículo. De seguida, é calculada a **velocidade individual** do veículo, ou seja, a velocidade a que este realmente circulou. Por fim, a **diferença entre estas velocidades** é calculada através da subtração da velocidade individual e da velocidade média do trajeto. Para além disto, utilizou-se o **Column Filter** para remover todas as colunas consideradas desnecessárias para a previsão da intensidade do trânsito.

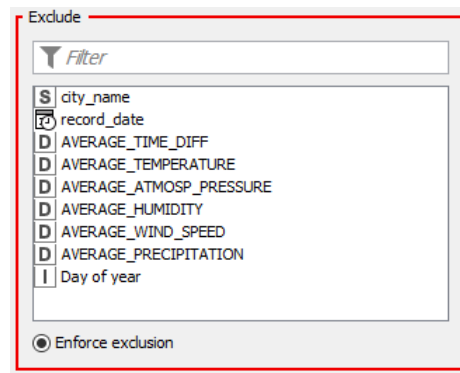


Figura 3.5: Filtragem das colunas do *dataset*

Esta sequência de passos pode ser ilustrada na seguinte figura em ambiente KNIME.

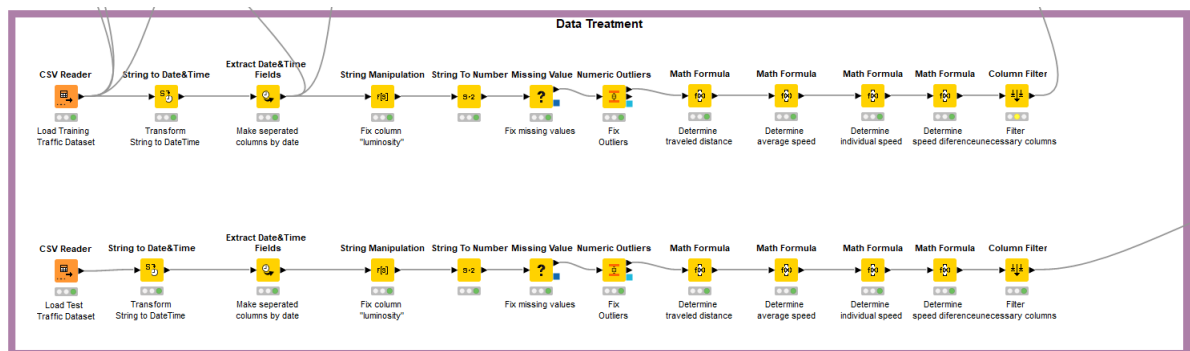


Figura 3.6: Tratamento de dados do *dataset*

3.3 Previsão da variável alvo

Feito o tratamento de dados, foi decidido aplicar uma **Decision Tree** juntamente com o método de **Cross-validation**. Desta forma foram construídas duas secções do *workflow*: uma secção para otimizar as variáveis do modelo da *Decision Tree* tais como a *quality measure*, *pruning*, *minimum number of records per node*, *number of records to store for view* e o *number of threads*. A segunda secção será para realizar uma *selecção das features do dataset*.

3.3.1 Cross-validation

Como mencionado no *dataset* sobre o valor de habitação nos EUA, decidiu-se utilizar a técnica de *Cross-validation* para validar o modelo de *Machine Learning* desenvolvido e diminuir qualquer tipo de *overfitting* que possa ter ocorrido durante a previsão da variável alvo. Dito isto, também foi atribuído o valor de 10 ao *k-fold*.

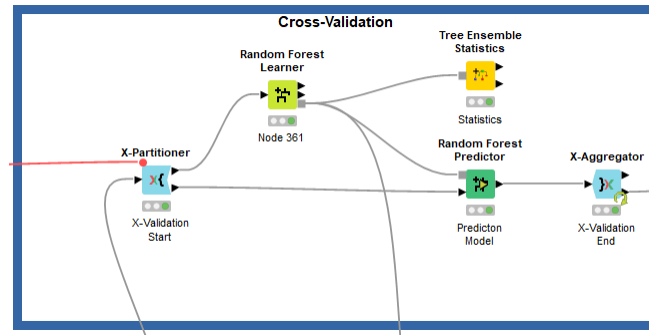


Figura 3.7: *Cross-validation*

Seguidamente, foram utilizados os nodos **X-Partitioner** e **X-Aggregator** da mesma maneira que foram utilizados no *dataset* anterior. O valor de validações continua igual ao valor do *k-fold*, ou seja, $k=10$ e posteriormente é gerada uma nova coluna com as previsões da variável alvo.

Figura 3.8: Definição do nodo *X-Partitioner*

Figura 3.9: Definição do nodo *X-Aggregator*

3.3.2 Tuning

Relativamente ao *tuning*[2], este foi dividido numa parte final e parte inicial. Na fase inicial foi realizado o *tuning* de parâmetros nominais e de parâmetros numéricos, respetivamente.

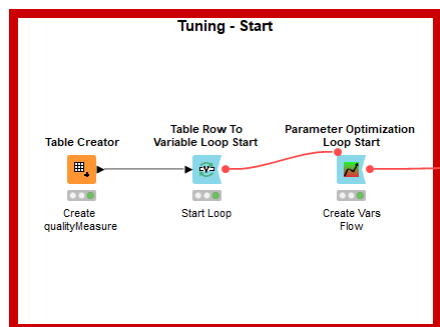


Figura 3.10: Parte inicial do *tuning*

Nesta fase do *tuning*, foram utilizados os mesmos métodos que foram aplicados no *dataset* anterior como a utilização do **Table Creator** para otimizar a **Quality Measure**, criar as variáveis necessárias para a *Cross-validation* através do nodo **Parameter Optimization Loop Start** e o uso da estratégia de **Brute Force** para ajudar a previsão do *target*.

Já na parte final do *tuning*, de maneira a que seja realizado o cálculo da *accuracy*, optou-se por se utilizar o nodo **GroupBy**, que através das percentagens fornecidas pelo *X-Aggregator*, calcula a média dos erros. Esta média é usada posteriormente no **Math Formula** que efetuar o cálculo da *accuracy* segundo a fórmula: $accuracy = 100 - mean_error$

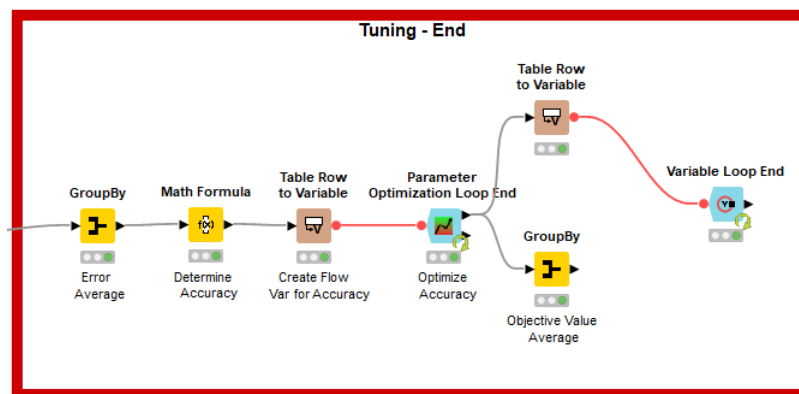


Figura 3.11: Parte final do *tuning*

Posteriormente, foram utilizados os mesmo métodos utilizados no *tuning* final do *dataset* anterior (secção 2.3.2) onde é utilizado o **Parameter Optimization Loop End** de forma a maximizar a função objetivo, que neste caso será a *accuracy*, e passar a mesma para o **Variable Loop End** onde estão definidos os parâmetros a otimizar.

3.3.3 Feature Selection

Como já descrito no *dataset* anterior (secção 2.3.3), foi construído um *workflow* que permite uma seleção óptima de *features* do *dataset*. Aqui é adotado o mesmo método de utilização da técnica de *Cross-validation* juntamente com o nodo **Feature Selection Filter** para apresentar as melhores escolhas no que toca às *features* do *dataset*.¹

3.4 Modelos desenvolvidos e resultados obtidos

3.4.1 Scorer V1

Numa primeira versão do modelo, apenas foram tratados os dados relativos à data, recolhendo o dia, mês, ano e a hora, realizando um **Partitioning** dos dados de treino para serem usados na aprendizagem da **Decision Tree**. Após a aprendizagem, a previsão é qualificada pelo nodo **Scorer**.

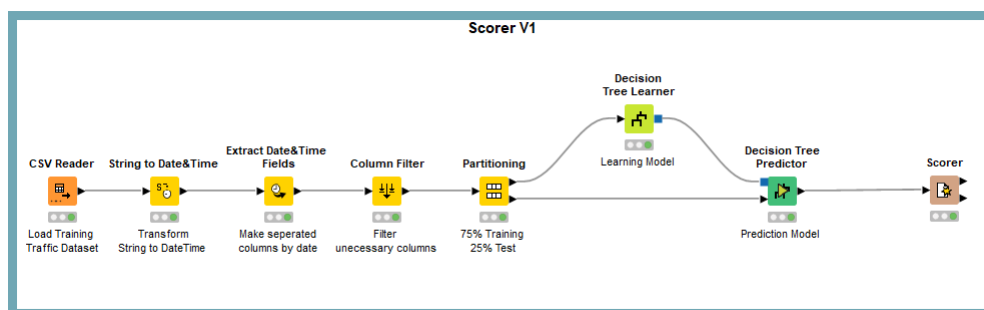


Figura 3.12: Primeira versão do modelo de aprendizagem

3.4.2 Scorer V2

Através desta versão do modelo de aprendizagem, é possível verificar que existe um aperfeiçoamento do *workflow* referido anteriormente, uma vez que existe a inserção do **tuning** inicial e final bem como a inserção do conceito **Cross-validation**, sendo ambos idênticos ao que foi mencionado nos capítulos anteriores. Optou-se por manter o nodo *Decision Tree Learner* e abdicou-se do *Scorer* para a introdução do cálculo final da *accuracy* como no modelo final.

¹A figura ilustrativa deste *workflow* encontra-se na secção de anexos do presente relatório. Figura 6.1

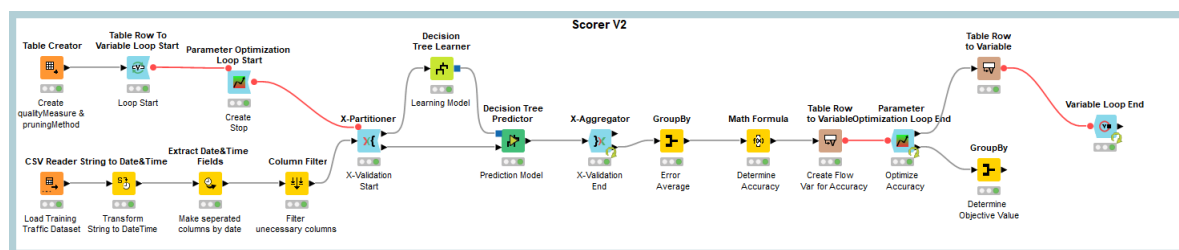


Figura 3.13: Segunda versão do modelo de aprendizagem

3.4.3 Versão final

Relativamente à versão final do modelo de aprendizagem é importante referir que foi introduzido o tratamento de dados para os dados de teste, que implicou uma introdução de um novo nodo **Predictor** para estes novos dados. Também foi introduzida a formatação dos dados de *output* para um ficheiro CSV, com a filtragem das colunas desnecessárias e com a introdução da coluna *RowID*.

Também é importante referir que este *workflow* pode ser dividido em duas partes: a primeira parte, onde o modelo utiliza o **Decision Tree Learner**, obtendo alguns resultados, e uma segunda parte, onde foi introduzido o conceito de **Random Forest**, permitindo obter uma melhoria significativa nos resultados. De uma forma muito simplista, uma *Random Forest* é um conjunto de *Decision Trees*, daí o aumento significativo da *accuracy*.²

3.4.4 Resultados obtidos

Relativamente aos resultados obtidos, numa primeira fase muito simplista do modelo, foi obtido um valor da *accuracy* igual a 59.836% o que foi considerado um valor não desejável para a previsão da variável alvo.

Na segunda versão do modelo de aprendizagem, começam a ser utilizados conceitos como *tuning* e *cross-validation*, que acabam por melhorar a performance do modelo atingindo uma *accuracy* de 62.581%. Isto é, quando é utilizado o *Gini index* sem o uso de qualquer tipo de *pruning*.

Após todas as otimizações do modelo, é possível verificar que o melhor resultado obtém-se quando se utiliza o **InformationGainRatio**. Sendo assim, os resultados são considerados satisfatórios, visto que todos se encontram na gama de valores acima dos 80%, o que permite uma previsão mais precisa da intensidade do trânsito na cidade do Porto.

Row ID	I stopCri...	I nrModels	I nrLevels	D Objective value	S RowID	S qualityMeasure	I currentIteration	I maxIterations
Row0	11	450	40	81.107	Best parameters	InformationGain	0	3
Row1	6	400	20	81.122	Best parameters	InformationGainRatio	1	3
Row2	6	100	30	80.93	Best parameters	Gini	2	3

Figura 3.14: Resultados do modelo de aprendizagem

²A figura ilustrativa deste *workflow* encontra-se na secção de anexos do presente relatório. Figura 6.2

Capítulo 4

Conclusão

Com a elaboração deste trabalho prático foi possível ter uma perspectiva perto de um contexto real de como os *datasets* têm de ser tratados de modo a produzirem bons resultados. Também se observou como esse tratamento se consiste num dos passos mais importantes na área de *Machine Learning*.

Foi também possível consolidar várias formas de utilização de modelos, assim como a importância da escolha do deste para um determinado *dataset*. Esta escolha juntamente com o tratamento correto do *dataset* pode melhorar consideravelmente a exatidão(*accuracy*) dos dados previstos.

Em suma, este trabalho prático foi crucial para consolidar e aplicar os vários tópicos e conceitos abordados e adquiridos na unidade curricular Aprendizagem e Decisão Inteligentes. Através de uma análise geral da solução implementada, podemos dizer que a equipa de trabalho foi capaz de atingir todos os objetivos inicialmente estabelecidos.

Capítulo 5

Referências

- [1] Cross-Validation : <https://www.geeksforgeeks.org/cross-validation-machine-learning/>
- [2] Model Tuning : <https://www.datarobot.com/wiki/tuning/>
- [3] Building your first ML model : <https://www.analyticsvidhya.com/blog/2017/08/knime-machine-learning/>

Capítulo 6

Anexos

24

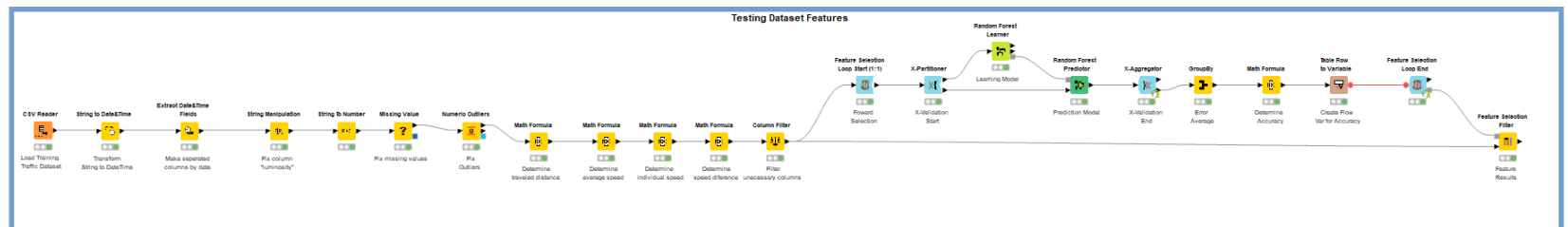


Figura 6.1: *Feature Selection workflow*

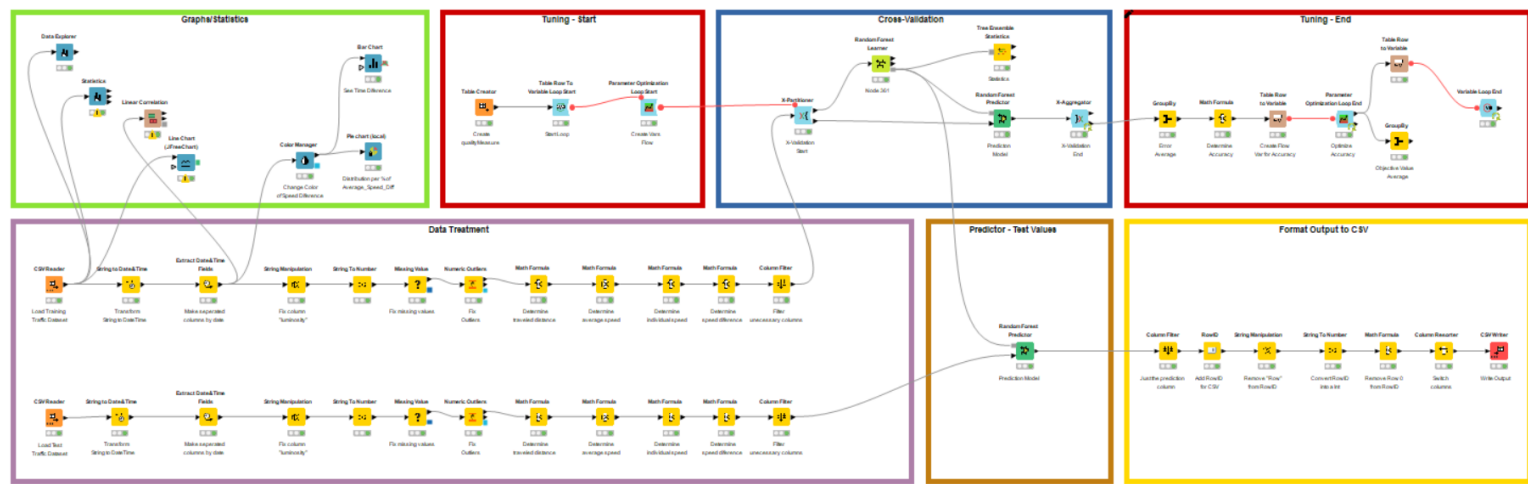


Figura 6.2: Versão final do modelo de aprendizagem