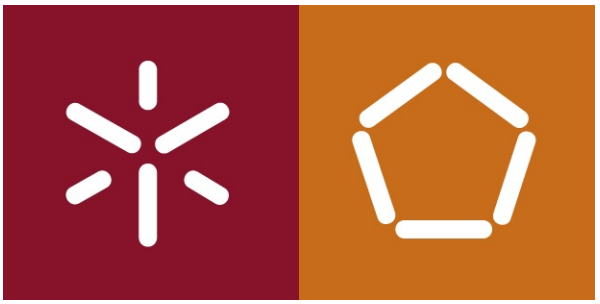


Universidade do Minho

Sistemas Baseados em Similaridade

Conceção e implementação de modelos de Machine Learning usando Árvores de Decisão

Autores:	*Números Aluno:*
Bruno Nascimento	A67647
João Palmeira	A73864
Rafael Silva	A74264

24 de Novembro de 2019

Conteúdo

Lista de Figuras

1 Introdução

Este trabalho prático foi desenvolvido no âmbito da UC de **Sistemas Baseados em Similaridade** e tem como principal desenvolver e implementar **modelos *Machine Learning*** usando **Árvores de Decisão** através da plataforma ***KNIME***.

Para tal foram desenvolvidos dois modelos de *Machine Learning* de maneira a responder a dois *datasets*: um para a modelação de **tráfego na cidade do Porto** e outro sobre ***Aumento da Temperatura Média Global***. De referir ainda que o modelo desenvolvido para o primeiro *dataset*, referente ao tráfego, foi utilizado para uma competição na plataforma ***Kaggle***.

Neste relatório são apresentadas todas as etapas da realização deste modelos, desde a análise inicial dos dados de cada *dataset*, o processamento, a avaliação das *features*, o tunning, a validade até aos testes de cada modelo.

De modo a concluir este trabalho iremos também realizar uma análise crítica dos resultados obtidos para cada modelo.

2 *DataSet* Intensidade do Trânsito

2.1 Análise dos dados

Inicialmente verifica-se que o *dataset* contém dados relativos ao tráfego na cidade do Porto num determinado período temporal, mais concretamente entre 24 de Julho de 2018 e 20 de Setembro de 2019. Este *dataset* inclui os seguintes atributos:

- **city_name** - nome da cidade em causa;
- **record_date** - o timestamp associado ao registo;
- **average_speed_diff** - a diferença de velocidade corresponde à diferença entre (1.) a velocidade máxima que os carros podem atingir em cenários sem trânsito e (2.) a velocidade que realmente se verifica. Quanto mais alto o valor, maior é a diferença entre o que se está a andar no momento e o que se deveria estar a andar sem trânsito, i.e., valores altos deste atributo implicam que se está a andar mais devagar;
- **average_free_flow_speed** - o valor médio da velocidade máxima que os carros podem atingir em cenários sem trânsito;
- **average_time_diff** - o valor médio da diferença do tempo que se demora a percorrer um determinado conjunto de ruas. Quanto mais alto o valor maior é a diferença entre o tempo que demora para se percorrer as ruas e o que se deveria demorar sem trânsito, i.e., valores altos implicam que se está a demorar mais tempo a atravessar o conjunto de ruas;
- **average_free_flow_time** - o valor médio do tempo que demora a percorrer um determinado conjunto de ruas quando não há trânsito;
- **luminosity** - o nível de luminosidade que se verificava na cidade do Porto;
- **average_temperature** - o valor médio da temperatura para o record_date na cidade do Porto;
- **average_atmosp_pressure** - o valor médio da pressão atmosférica para o record_date;
- **average_humidity** - o valor médio da humidade para o record_date;
- **average_wind_speed** - o valor médio da velocidade do vento para o record_date;
- **average_cloudiness** - o valor médio da percentagem de nuvens para o record_date;
- **average_precipitation** - o valor médio de precipitação para o record_date;
- **average_rain** - avaliação qualitativa da precipitação para o record_date.

O objetivo da análise deste *dataset* é perceber qual o atributo que será necessário utilizar para o modelo ser capaz de prever a intensidade do transito num dado momento com a melhor *accuracy* possível. Como tal importa destacar que esse atributo é o ***"average_speed_diff"***.

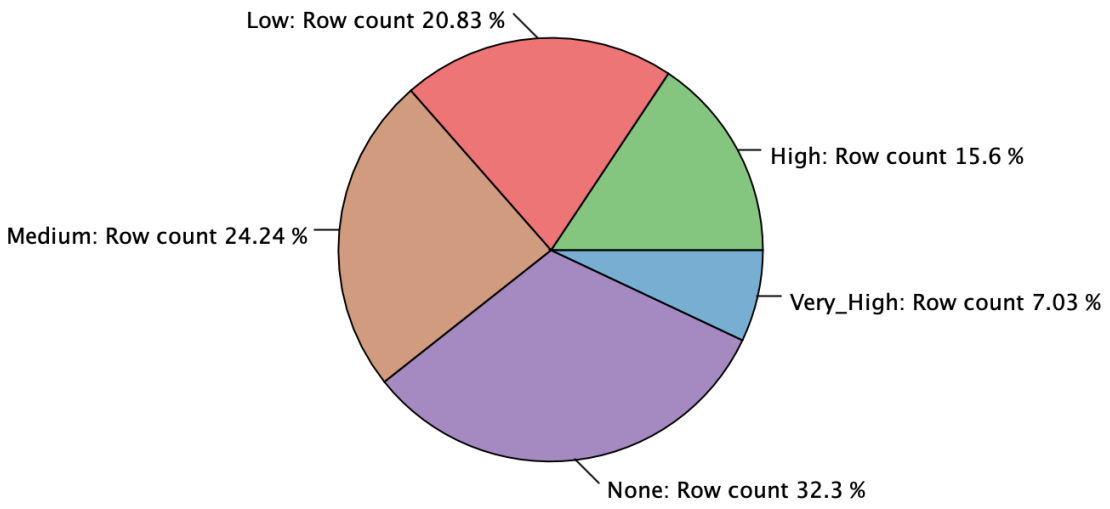


Figura 1: Distribuição do atributo **"average_speed_diff"**

Como se pode observar pela figura anterior, o atributo é composto pelos parâmetros ***"Low"***, ***"Medium"***, ***"High"***, ***"Very_High"*** e ***"None"*** que definem a intensidade do trânsito. Estes parâmetros constituem uma escala que será utilizada para realizar as previsões.

Através do gráfica é possível elaborar uma ideia do trânsito que existe ao longo do dia, isto é, através do nosso quotidiano verificamos que, como se pode observar no gráfico. a maior percentagem corresponde ao *"None"*, cerca de 32%, logo podemos concluir que ao longo do dia existem bastantes períodos onde não existe qualquer tipo de trânsito nomeadamente de madrugada. Já o caso do *"Very_High"* devido à sua percentagem ser menor (7%), conclui-se que é referente a períodos como o início da manhã e o final da tarde, alturas em que as pessoas entram/saem dos seus empregos ou levam/buscam os seus filhos à escola.

Os valores de *"Low"*, *"Medium"*, *"High"* são respectivamente 21%, 24% e 16%, fazendo com que se conclua que o transito ao longo do dia tem um tráfego mediano, visto que existem percentagens altas tanto para o *"Low"* como para o *"Medium"*.

2.2 Tratamento dos dados

Numa fase inicial da contrução deste modelo de *Machine Learning*, foi necessário realizar um tratamento de dados, isto é, realizou-se uma análise crítica do conjunto de dados fornecidos nos *datasets* e determinou-se quais seriam os dados a serem utilizados, quais teriam que ser alterados e quais não seriam utilizados. De certa forma foram selecionados os atributos imprescindíveis para a criação do modelo.

Inicialmente começou-se por transformar a string ***"record_date"*** de cada dados para o tipo ***Date&Time*** para, posteriormente, retirar o **dia** da semana, o **mês**, o **ano** e a **hora** do dia desse *DateTime*. De seguida através de um ***Column Filter*** removeu-se as colunas que achamos desnecessárias, como podemos ver na figura seguinte.

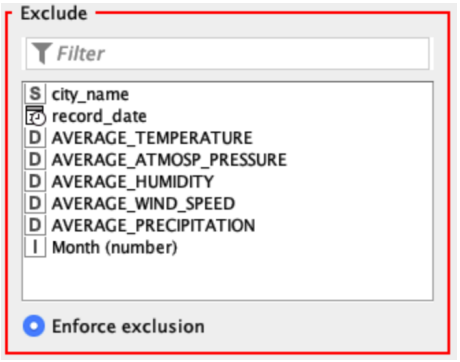


Figura 2: Filtragem das colunas

Optou-se por utilizar um ***String Manipulation*** de modo a transformar os valores da coluna ***LUMINOSITY*** que são *strings* em valores inteiros e alterou-se o tipo da coluna através do ***String to Number***:

- **DARK = 0**
- **LIGHT = 1**
- **LOW_LIGHT = 2**

É usado o ***Missing Value*** para alterar os *missing values* das tabelas *AVERAGE_CLOUDINESS* e *AVERAGE_RAIN* para ***NULL***. Posteriormente, através dos quatro ***Math Formulas***, é elaborado o cálculo da **distância** percorrida para cada *row* que é utilizada para fazer calcular a **velocidade média** para cada *row*. De seguida, é calculado a **velocidade individual** de cada *row*, ou seja, a velocidade a que realmente o carro circulou e, por fim, a **diferença entre estas velocidades** calculada através da subtração da velocidade individual à velocidade média do trajeto. Por último, utilizou-se um ***Column Filter*** com o intuito de remover colunas desnecessárias.

Esta sequência de passos é comprovada pela próxima figura.

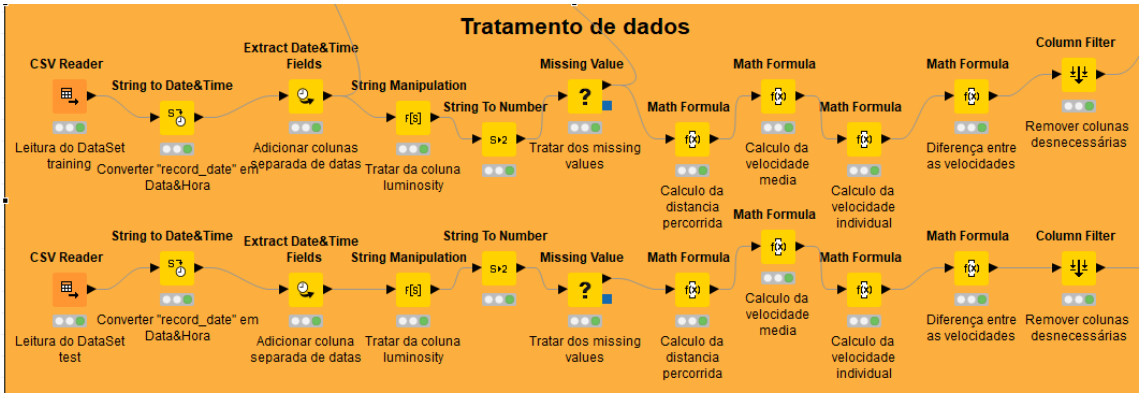


Figura 3: Tratamento de dados no *Workflow*

Foi ainda aplicado um ***Color Manager*** e um ***Pie chart (local)*** para elaborar o gráfico referido na secção anterior.

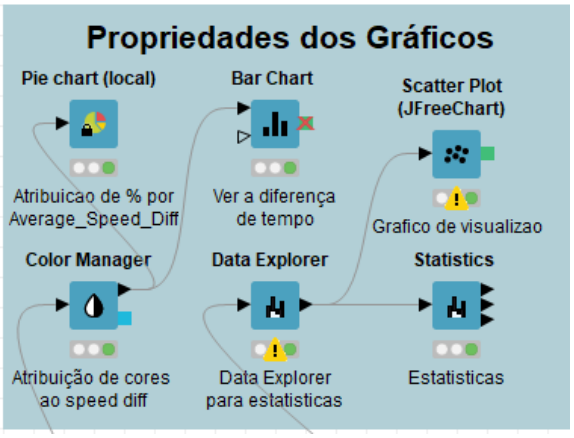


Figura 4: Propriedades do gráficos

De modo a obter mais dados estatísticos decidimos construir mais dois gráficos através de um ***Bar Chart*** e de um ***Scartter Plot***, que constituem uma estatística do *AVERAGE_SPEED_DIFF* pelos seus atributos e uma estatística da distribuição dos valores da velocidade pelo tempo, respetivamente.

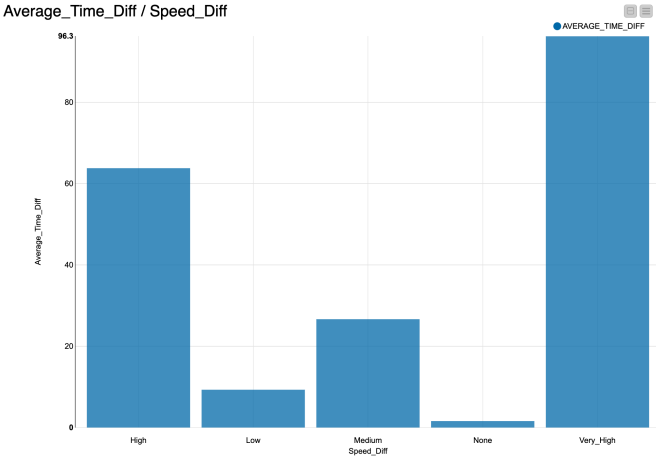


Figura 5: Distribuição do *AVERAGE_SPEED_DIFF*

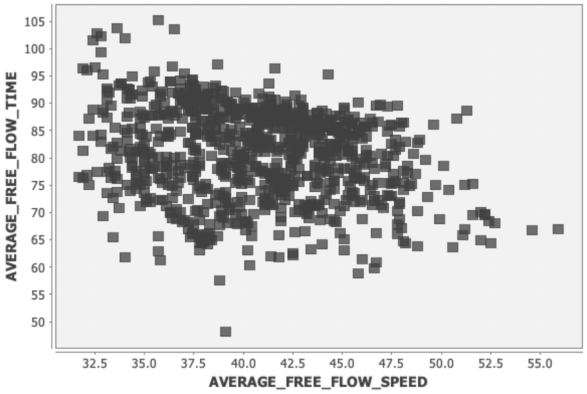


Figura 6: *AVERAGE_FREE_FLOW_TIME*/*AVERAGE_FREE_FLOW_SPEED*

2.3 Métodos aplicados

Após o tratamento de dados, decidiu-se aplicar uma ***Decision Tree*** através do método de ***Cross-validation***. Desta forma foram criados dois *workflows*: um *workflow* 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***; um segundo *workflow* para realizar uma ***feature selection***.

2.4 *Cross-validation*

`Cross-validation` é uma técnica de validação de modelos de *Machine Learning* que tem como objetivo ter uma métrica precisa do desempenho do modelo na prática.

Essencialmente, consiste em dividir o conjunto de dados em ***k-folds***. Em cada execução do modelo, k-1 dobras são usadas para treino e 1 dobra (a restante) é usado como teste. Continua-se a repetir o processo até que todas as dobras tenham sido usadas para teste. A métrica de erro final é baseada no valor médio de todas as métricas de erro.

Foi decidido colocar em prática o `Cross-validation` devido a uma das suas vantagens: **diminuição do *overfitting***. O *overfitting* consiste na produção de uma análise que corresponde aproximadamente a um determinado conjunto de dados e, portanto, pode falhar ao se ajustar aos dados adicionais ou prever observações futuras sem fiabilidade.

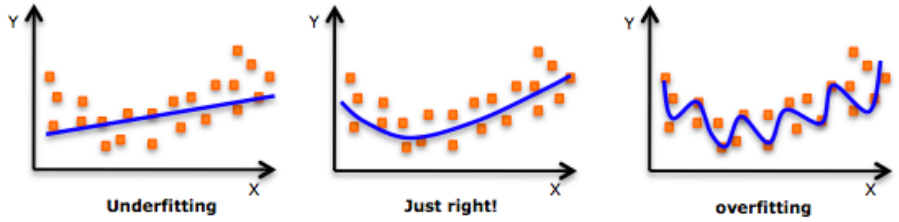


Figura 7: Tipos de modelos

Queremos com isto dizer que o *overfitting* reflete que o modelo memoriza e modela demasiado o *training set*, originando uma previsão errada caso se utilize outro *dataset* no modelo.

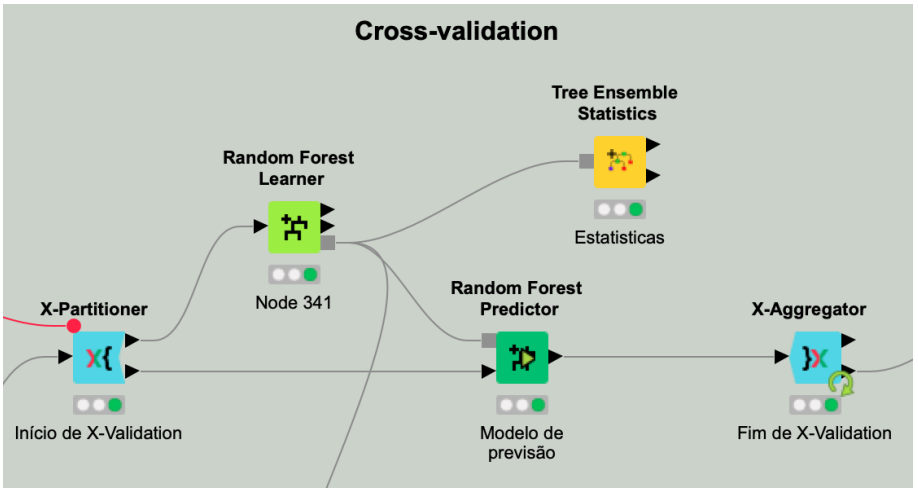


Figura 8: *Cross-validation* no *Workflow*

Assim sendo para a construção deste modelo optou-se por atribuir uma valor de 10 ao *k-fold* (k=10).

Figura 9: *Cross-validation* com *k-fold*=10

Utilizou-se um ***"X-Partitioner"*** e um ***"X-Aggregator"*** para que fosse possível aplicar esta técnica. O primeiro faz a repartição do *dataset* pelo *k-fold* definido, ou seja, por k=10 (10 partes iguais) com *sampling* aleatório. Já o segundo serve para definir a coluna "alvo"e a coluna da previsão, retornado a tabela das previsões já com a percentagem de erro de cada *fold*.

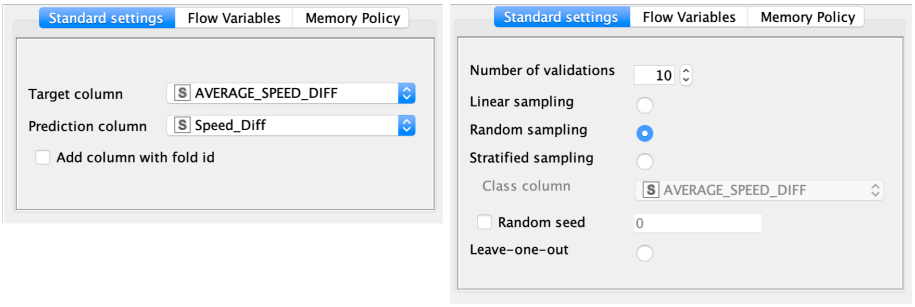


Figura 10: Definições dos nodos "*X-Partitioner*"e "*X-Aggregator*"

2.5 *Tuning*

Já relativamente ao *tuning*, este foi dividido na parte inicial e na parte final. Na fase inicial foi realizado o *tuning* de parâmetros nominais e de parâmetros numéricos, respetivamente.

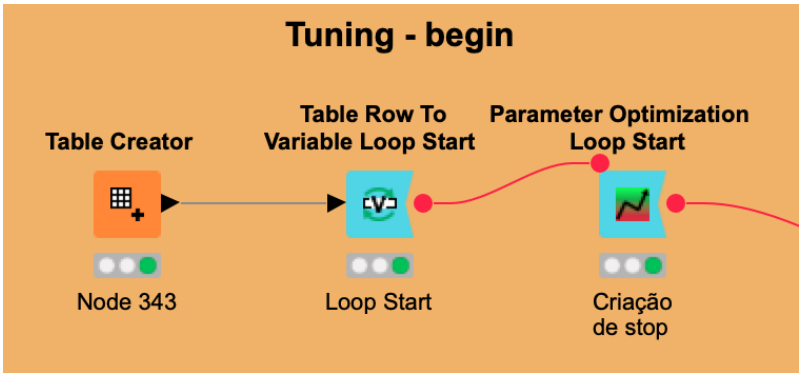


Figura 11: Parte inicial do *tuning*

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

	qualityMeasure
Row0	InformationGain
Row1	InformationGainRatio
Row2	Gini

Figura 12: Definições "*Table Creator*"

Após a introdução dos parâmetros nominais, aplicou-se os parâmetros numéricos. Através do ***Parameter Optimization Loop Start*** criou-se três variáveis diferentes: ***nrModels*** que corresponde ao número máximo de modelos a usar, ***nrLevels*** que corresponde ao número máximos de níveis de cada árvore de decisão e o ***stopCriteria*** que corresponde a uma variável que serve de critério de paragem. A estratégia utilizada foi a ***Brute Force***.

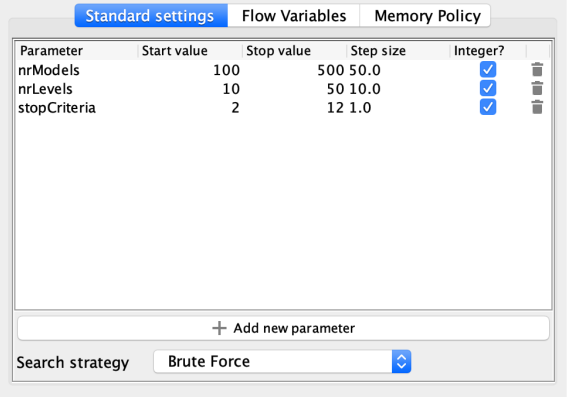


Figura 13: Definições "*Parameter Optimization Loop Start*"

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

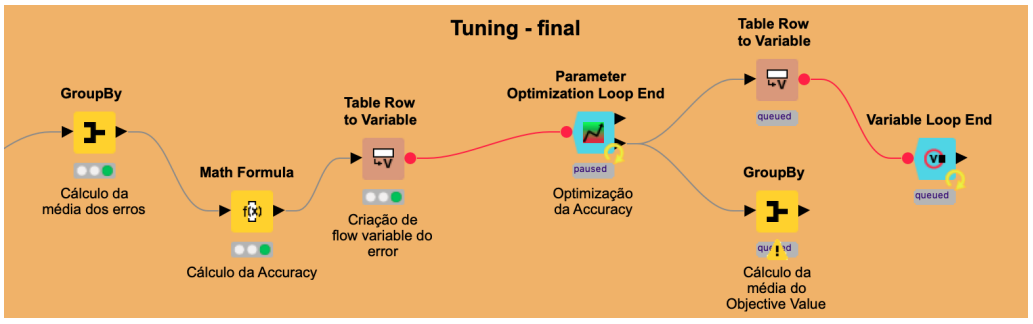


Figura 14: Parte final do *tuning*

Seguidamente no ***Parameter Optimization Loop End*** selecionou-se a função objetivo que se queria maximizar, isto é, neste caso selecionou-se a opção *maximized* para a *accuracy*. Estes dados são, por fim, passados ao ***Variable Loop End*** onde foram definidos os parâmetros a otimizar, como se pode observar na figura 13. Colocou-se ainda um ***GroupBy*** com o intuito 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 do grupo de trabalho.

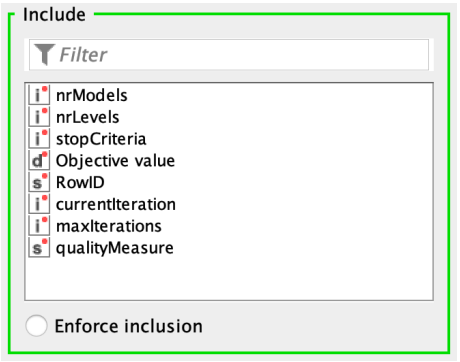


Figura 15: Definições "*Variable Loop End*"

2.6 *Feature Selection*

De modo a otimizar as escolhas das *features* a utilizar para gerar valores de *accuracy* desejáveis, decidiu-se criar um *workflow* que realiza-se uma seleção ótima das *features* a utilizar na nossa previsão.

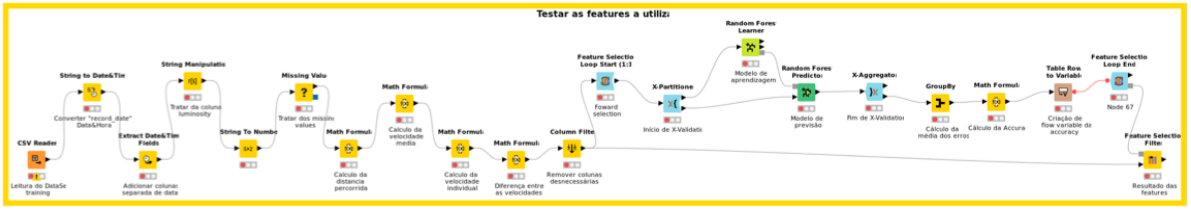
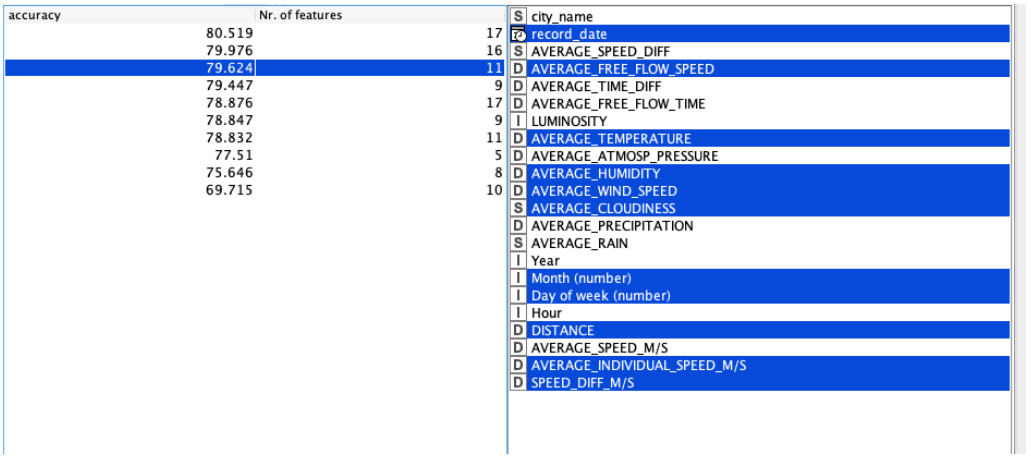


Figura 16: *Feature Selection Workflow*

Como se pode observar através da Figura 16 o processo é semelhante ao que já foi descrito, embora não haja *tuning*, utilizou-se um ***Feature Selection Filter*** de modo a obter a seleção ótima tal como no exemplo da Figura 17.



accuracy	Nr. of features
80.519	17
79.976	16
79.624	11
79.447	9
78.876	17
78.847	9
78.832	11
77.51	5
75.646	8
69.715	10

S	city_name
D	record_date
S	AVERAGE_SPEED_DIFF
D	AVERAGE_FREE_FLOW_SPEED
D	AVERAGE_TIME_DIFF
D	AVERAGE_FREE_FLOW_TIME
I	LUMINOSITY
D	AVERAGE_TEMPERATURE
D	AVERAGE_ATMOSP_PRESSURE
D	AVERAGE_HUMIDITY
D	AVERAGE_WIND_SPEED
S	AVERAGE_CLOUDINESS
D	AVERAGE_PRECIPITATION
S	AVERAGE_RAIN
I	Year
I	Month (number)
I	Day of week (number)
I	Hour
D	DISTANCE
D	AVERAGE_SPEED_M/S
D	AVERAGE_INDIVIDUAL_SPEED_M/S
D	SPEED_DIFF_M/S

Figura 17: Exemplo de uma *feature selection*

2.7 Modelos desenvolvidos e resultados obtidos

2.7.1 1ª Versão

Nesta primeira versão, apenas tratamos dos dados relativos à data, recolhendo o dia, o mês, o ano e a hora, e realizando um ***Partitioning*** dos dados de treino para serem usado na aprendizagem de uma ***Decision Tree Learner***. Após a aprendizagem, a previsão é qualificada através de um ***Scorer***.

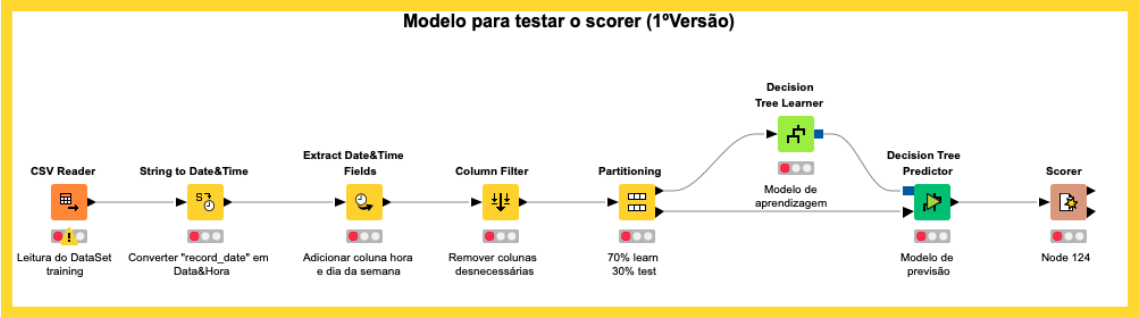


Figura 18: Primeiro Modelo

2.7.2 2ª Versão

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

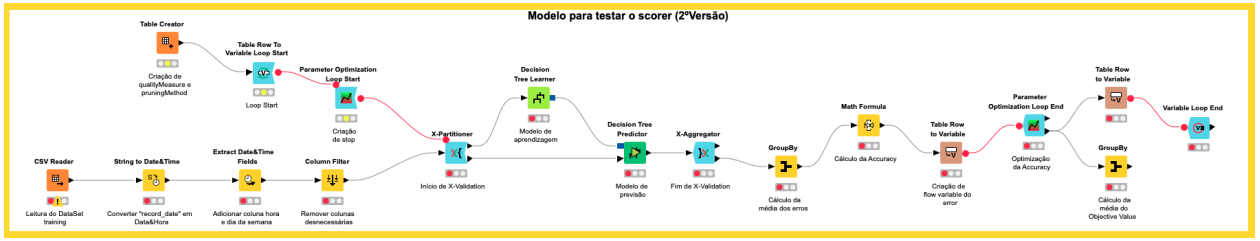


Figura 19: Segundo Modelo

2.7.3 3ª Versão

Relativamente a esta terceira versão, bastante semelhante ao *workflow* final, é importante referir que foi introduzido o tratamento de dados para os dados de teste, que implicou a introdução de um novo ***Predictor*** para estes novos dados, e, também foi introduzido, a formatação dos dados de *output* para CSV, com a filtragem das colunas bem como a introdução da coluna *RowId*.

Ainda a respeito desta versão, ela pode ser dividida em dois estágios: primeiro estágio onde o modelo de *workflow* utiliza uma ***Decision Tree Learner***, obtendo resultados interessantes, e um segundo estágio onde foi introduzido o conceito de ***Random Forest***, permitindo obter uma melhoria significativa nos resultados. Basicamente, de uma forma simplista, uma *Random Forest* é um conjunto de *Decision Trees*, daí o aumento significativo da *accuracy*.

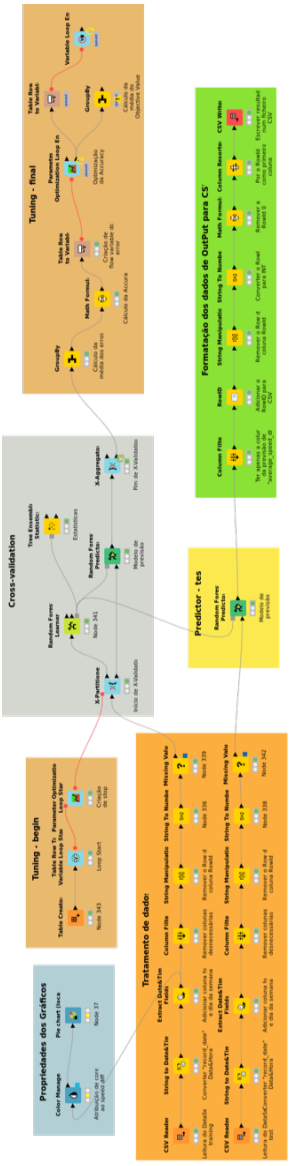


Figura 20: Terceiro Modelo

2.7.4 Versão Final

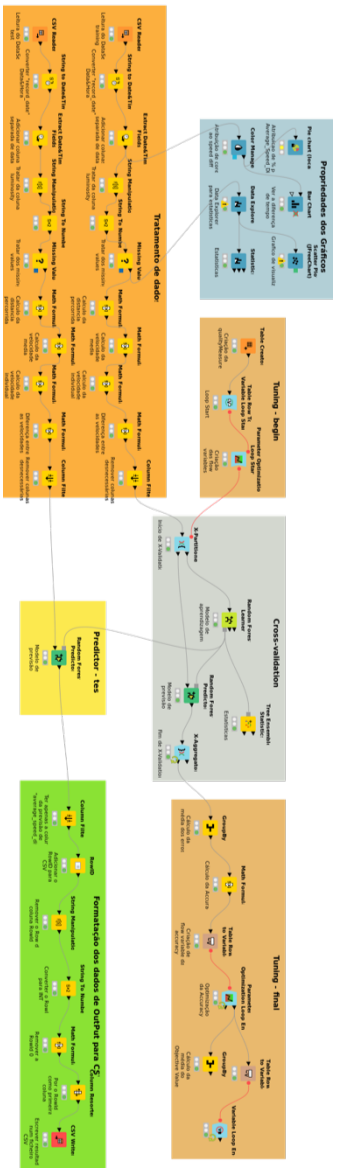


Figura 21: *Workflow* final

Por último, a versão final, em relação à versão anterior, contou com a introdução de novos dados tal como foi mencionado nos capítulos anteriores e que dizem respeito a: ***distância do trajeto, velocidade média do trajeto, velocidade individual do carro no trajeto, diferença entre a velocidade média e velocidade individual.***

Com a introdução destes novos dados, acha-se que existe uma maior exatidão nos resultados obtidos referentes à *accuracy*.

2.8 Submissão no *Kaggle*

Quanto à submissão no *Kaggle*, o nosso grupo foi o que realizou mais submissões, fruto também dos vários modelos que desenvolveu, das várias implementações que decidiu realizar. Como tal a escolha da submissão final foi difícil e, como tal, o grupo decidiu utilizar a última que foi realizada, apesar de não ter sido a que nos tenha fornecido valores mais altos tanto no *Kaggle* como no KNIME, o grupo estava confiante da fiabilidade dos resultados deste mesmo *output*.

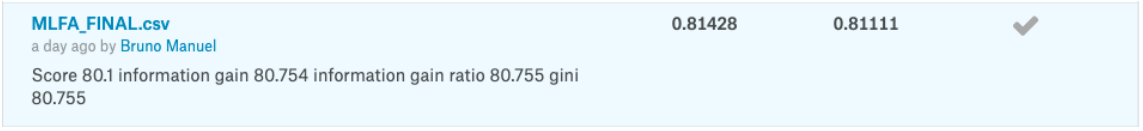


Figura 22: Submissão escolhida

Relativamente à melhor submissão é de destacar, que apesar do valor fornecido pelo KNIME do *output* dessa submissão ser razoável, os valores obtidos no *Kaggle* público não foram satisfatórios (78.66%) embora no *leaderboard* privado tenham sido bastante satisfatórios.

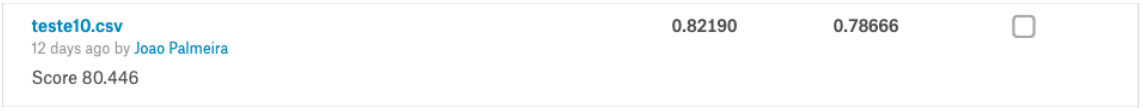


Figura 23: Melhor submissão

3 *DataSet* da Temperatura Global

3.1 Contextualização

Neste capítulo é será explicado a segunda parte do projeto, além do *dataset* fornecido pelos docentes, que depois é submetido perante uma competição no *Kaggle*, com o uso do programa KNIME. Agora temos liberdade de escolha perante o *dataset* que queremos analisar, nesta segunda parte não temos que submeter as previsões numa plataforma como o *Kaggle*, basta medirmos localmente a capacidade de aprendizagem do modelo que iremos treinar.

Como tal procedemos a uma pesquisa de um *dataset* interessante, após uma breve recolha de opiniões, *datasets* tais como o do Titanic que prevê se um passageiro vai sobreviver, ou um que contém pacientes se estão ou não em risco de serem pacientes oncológicos, o grupo continuou a sua pesquisa até com um tema a nível global, as **alterações de temperatura média do Planeta Terra**.

O *dataset* que escolhemos para analisar é sobre a variação da temperatura da Terra, tanto na terra como no mar desde 1855 até 2013. Através deste o grupo pretende realizar uma aprendizagem através dos dados fornecidos de maneira a que seja possível prever a temperatura média apenas da terra.

No desenvolvimento é necessário ter em consideração, que ao usar um algoritmo árvore de decisão, ter em conta os valores de *AUC(Area Under The Curve)* e a curva *ROC (Receiver Operating Characteristics)*. Um valor base do modelo sem algoritmos de *Random Forest* ou a criar um modelo de overfitting é possível obter valores de 79% de taxa de *accuracy*. Para aprendizagem o programa KNIME é o programa aconselhado para treinar e testar a aprendizagem.

3.2 Análise dos dados

No presente *dataset* foi possível tratar os dados que em seguida enumeramos:

- **Date** - começa em 1750 para temperatura média da terra e 1850 para temperaturas máximas e mínimas da terra e temperaturas globais do oceano e da terra;
- **LandAverageTemperature** - temperatura média global da terra em graus *Celsius*;
- **LandAverageTemperatureUncertainty** - o intervalo de confiança de 95% em torno da média;
- **LandMaxTemperature** - média da temperatura global máxima da terra em graus *Celsius*;
- **LandMaxTemperatureUncertainty** - o intervalo de confiança de 95% em torno da temperatura máxima da terra;
- **LandMinTemperature** - média da temperatura global mínima da terra em graus *Celsius*;
- **LandMinTemperatureUncertainty** - o intervalo de confiança de 95% em torno da temperatura mínima da terra;
- **LandAndOceanAverageTemperature** - temperatura média global da terra e do mar em graus *Celsius*;
- **LandAndOceanAverageTemperatureUncertainty** - o intervalo de confiança de 95% em torno da média;

O nosso objetivo será prever o valor médio da temperatura da Terra, sendo, para isso, necessário enfatizar o atributo **LandAverageTemperature**. Para tal começamos por fazer uma análise gráfica da temperatura ao longo dos anos e, à semelhança do que aconteceu com a primeira parte do projeto, gerou-se o gráfico que é apresentado a seguir. Como se pode observar, a temperatura ao longo dos anos tem tendência a aumentar, algo a ter em consideração a quando da análise dos resultados.

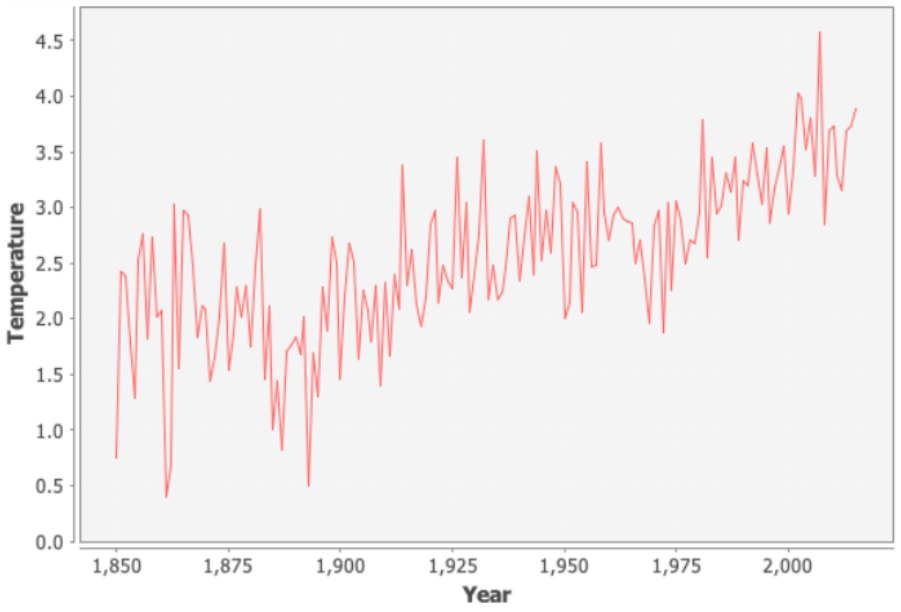


Figura 24: Distribuição da temperatura ao longo dos anos

3.3 Tratamento dos dados

Relativamente ao tratamento de dados para este *dataset* foram removidas as *rows* com *missing values* através do ***Missing Values***. Foi utilizado um ***Auto-Binner*** com o intuito de intervalar os valores da temperatura média da Terra.

De seguida, com o auxílio de um ***Cell Splitter***, partiu-se a data em *Year*, *Month* e *Day*, seguindo uma ideologia semelhante à primeira parte do projeto com o objetivo de melhorar a previsão.

Por fim, é usado um ***Column Filter*** de modo a selecionar as colunas que são realmente relevantes, que neste caso são todas com exceção do dia e a string data original.

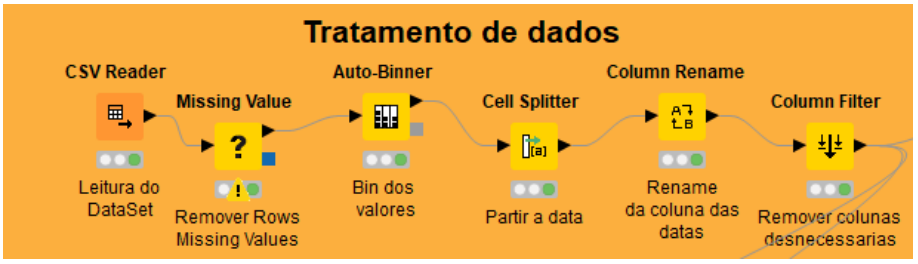


Figura 25: Tratamento de dados no *Workflow*

3.4 Tuning

Tal como no *dataset* anterior *(Previsão Tráfego)*, neste modelo também foi implementado um *tuning* inicial e um *tuning* final.

3.4.1 Tuning Inicial

No *tuning* inicial definiu-se os parâmetros de medição de qualidade, para tal usou-se ***Table Creator***, onde declarou-se os seguintes parâmetros ***Quality Measure*** que se obtém através do ***Gini Index***. De seguida aplicou-se ***Table Row To Variable Loop Start*** onde são efetuadas as iterações com a finalidade de calcular os valores correspondentes ás variáveis previamente declaradas. A controlar estes ciclos estão as variáveis implementadas em ***Parameter Optimization Loop Start***, neste *dataset* apenas está implementado o ***stop criteria*** que serve como critério de paragem.

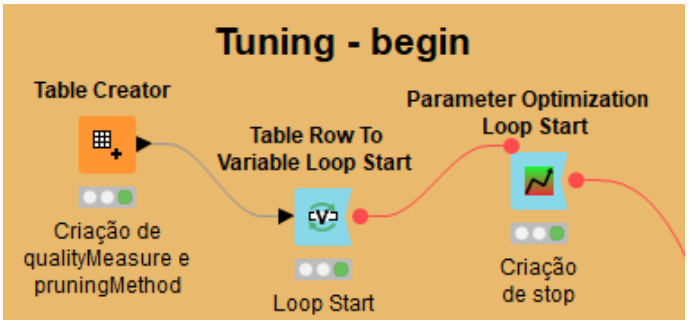


Figura 26: *Nodo Tuning Begin*

3.4.2 *Tuning* Final

Relativamente ao *Tuning* Final, tal como no *datset* anterior, procedeu-se á utilização de um ***GroupBy*** que recebe os valores do *X-Aggregator*, faz o cálculo da média de erros, e logo de seguida usa-se uma ***Math Formula*** para calcular a *accuracy* final. Após esse cálculo, usa-se um ***Parameter Optimization Loop End*** para maximizar o valor da *accuracy*, por fim são retornados os valores a ***Variable Loop End*** que retorna os melhores valores de parâmetro.

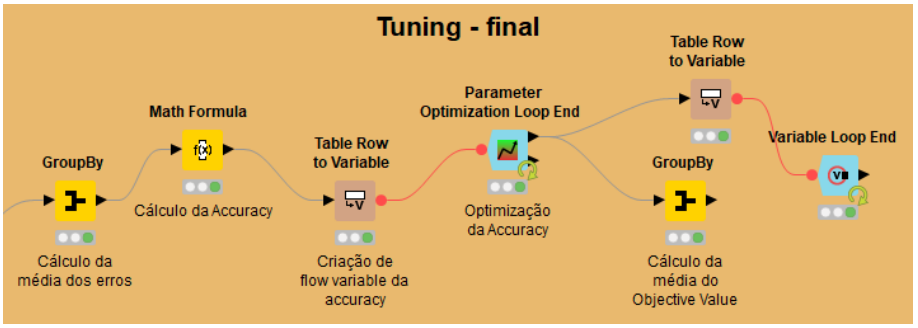


Figura 27: *Nodo Tuning Final*

3.5 *Cross-Validation*

Relativamente á parte de *cross validation*, para este *dataset* em especifico foi usada uma ***Decision Tree***, com este algoritmo, é atribuido um valor a cada nodo da árvore, e a cada ponto é dividido em dois caminhos, o algoritmo calcula a cada divisão *Gini Index* e o *Gini Ratio*. No final caso seja aplicado um ***Post-Prunning*** de modo a reduzir o tamanho da árvore ao retirar nodos redundantes, este método é baseado no principio ***Minimum Description Length***

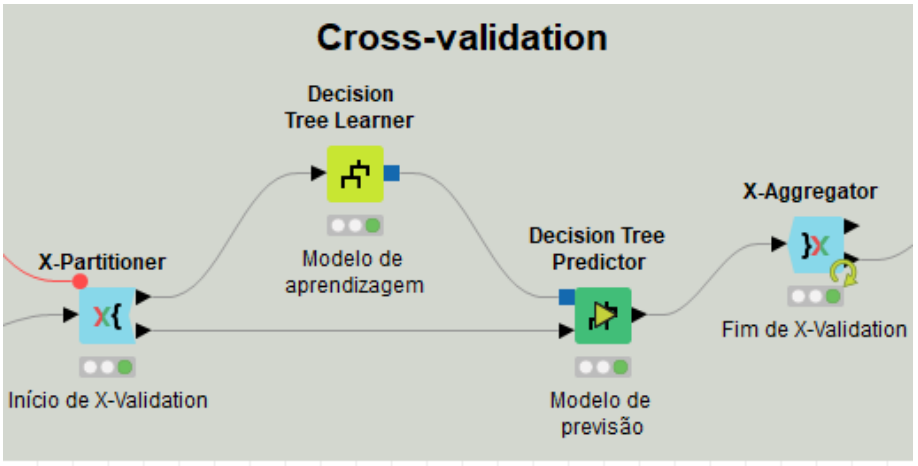


Figura 28: *Cross Validation*

3.6 *Workflow* desenvolvido

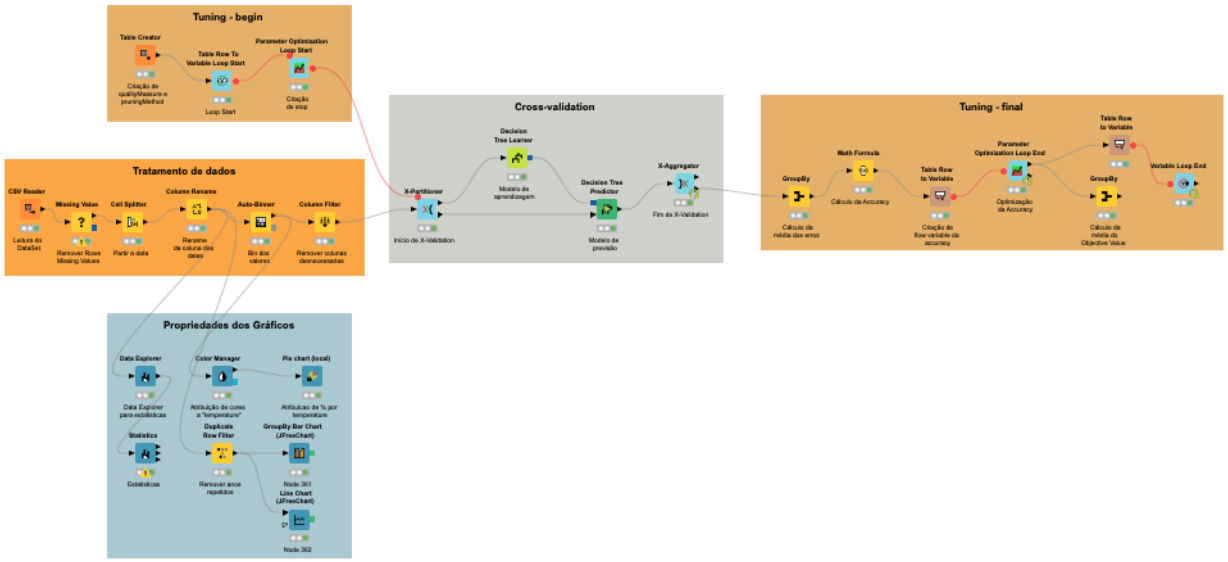


Figura 29: *Workflow* desenvolvido

Na figura 29 é possível observar o *workflow* desenvolvido e, para além do que já foi referido na secção anterior (tratamento de dados), este modelo é idêntico ao modelo desenvolvido para a competição realizada no *Kaggle*.

Excluindo o tratamento de dados, o modelo difere em relação ao anterior apenas na utilização de uma ***Decision Tree*** ao invés de uma *Random Forest*. Optou-se por se utilizar apenas uma *Decision Tree*, uma vez que os resultados obtidos foram bastante satisfatórios e não seria necessário envergar por uma opção mais poderosa.

3.7 Resultados obtidos

Por fim, nos resultados é possível verificar que o melhor resultado acontece quando **não se usa *pruning*** e utiliza-se o ***Gini Index***. Através destes resultados constata-se que o *pruning* retira informação importante a utilizar no treino e, como é necessário ter a árvore completa para obter um melhore *score*, neste modelo fazer *pruning* não compensa.

De resto os resultados obtidos são satisfatórios, uma vez que estão todos numa gama de valores acima dos 85%.

Row ID	I stopCriteria	D Objective value	S RowID	I curren...	I maxIt...	S qualityMeasure	S pruningMethod
Row0	3	85.945	Best parameters	0	4	Gain ratio	No pruning
Row1	3	85.543	Best parameters	1	4	Gain ratio	MDL
Row2	4	87.399	Best parameters	2	4	Gini index	No pruning
Row3	2	86.095	Best parameters	3	4	Gini index	MDL

Figura 30: Resultados obtidos

4 Conclusões

Ao longo deste trabalho, o grupo foi capaz de desenvolver várias competências na área de *Machine Learning* e na criação de modelos através da plataforma KNIME.

Em relação à primeira parte do projeto, fazendo uma retrospectiva, o grupo tomou o rumo descrito aqui, embora em certas situações fosse possível tomar outro tipo de decisões. Apesar dos resultados obtidos, o grupo sentiu ser capaz de atingir valores na ordem os 83% com a introdução dos novos conceitos como a distância e as velocidades, embora o tempo para desenvolver estas novas ideias tivesse sido escasso. Contudo, de um modo geral, o grupo acha que o modelo desenvolvido foi bastante satisfatório, já a escolha no *Kaggle* não foi realmente a melhor.

Quanto a segunda parte do projeto, escolhemos uma área que nos despertou interesse e construímos um modelo que, apesar das semelhanças com o modelo criado na primeira parte, foi capaz de nos ajudar a aprimorar ainda mais os conceitos do KNIME.

Em suma, através deste trabalho o grupo interessou-se por aprender e desenvolver competências nesta área de *Machine Learning*, nomeadamente as Árvores de Decisão, assumindo os desafios propostos e ultrapassando as diversidades que fomos encontrando neste percurso. De um modo geral, o grupo encontra-se satisfeito com o trabalho realizado.

5 Referências

https://www.knime.com/knime

https://www.knime.com/knime-introductory-course/chapter6/section3/random-forest

www.kaggle.com

Peter D. Grunwald, Título: Minimum Description Length Principle The Mit Press

A.Barron, J.Rissanen, Bin Yu Titulo : The minimum description length principle in coding and modeling IEEE Transactions on Information Theory