Universidade do Minho

Escola de Engenharia

Relatório Fase 2 do Trabalho Prático da Unidade Curricular Aprendizagem e Decisão Inteligentes

Grupo 48

A93253 David Alexandre Ferreira Duarte

A90614 Pedro Aquino Martins de Araújo

A94166 Samuel de Almeida Simões Lira

Table of Contents

Formatado: Inglês (Reino Unido)

Introdução

No desenvolvimento deste projeto o grupo decidiu abordar os seguintes paradigmas de aprendizagem, com supervisão e sem supervisão, para cada um dos *dataset* escolhidos. no entanto são utilizadas diferentes técnicas de aprendizagem, escolhidas a fim de melhor explorar cada *dataset*.

Metodologia Escolhida

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.

MADRID CLIMATE

Este *dataset* corresponde a recolha de dados relacionados ao clima, observados na cidade de Madrid dentre o período de 01/07/2008 e 20/04/2019. Esta amostra contém o total de 3946 linhas. Nos próximos capítulos, iremos indicar os passos de desenvolvimento de modelos de *machinelearning* aplicados ao d*ataset* em questão, seguindo a metodologia mencionada anteriormente.

Estudo dos Dados

No caso deste *dataset* o repositório que o continha, não tinha nenhum problema associado ao mesmo. Por essa razão o grupo decidiu analisar os dados e estudar o comportamento dos mesmos com o intuito de criar um problema para de seguida fazer o modelo que soluciona-se o mesmo.

Para atingir esse objetivo primeiramente foi observado quais são as variáveis existentes na amostra, demonstradas nas seguintes imagens e explicadas a seguir:

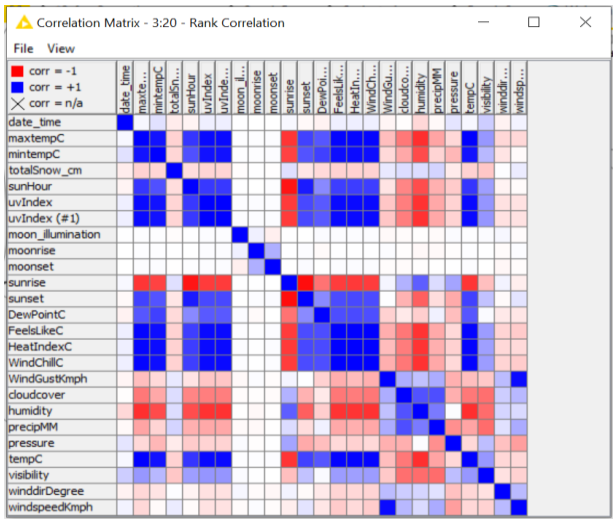
date_time	maxtempC	mintempC	totalSnow_cm	sunHour	uvIndex	uvIndex (#1)	moon_illumination	moonrise	moonset	sunrise	sunset	DewPointC
2008-07-01	33	17	0	14.5	6	6	4	04:32 AM	08:41 PM	06:48 AM	09:49 PM	6

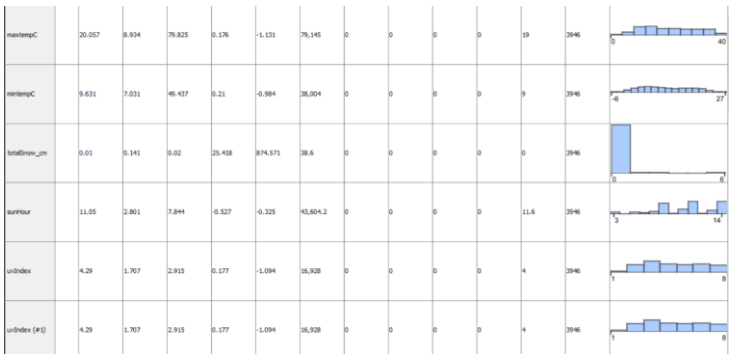
FeelsLikeC	HeatIndexC	WindChillC	WindGustKmph	cloudcover	humidity	precipMM	pressure	tempC	visibility	winddirDegree	windspeedKmph
28	28	28	11	5	28	0	1014	33	10	203	8

- date_time : representa a data exata que foi coletado os dados
- TotalSnow_cm: total de neve no dia
- MintempC: mínimo de temperatura coletada o dia
- MaxtempC: máximo de temperatura coletada no dia
- FeelslikeC: temperatura ambiente
- Humidity: humidade registada no dia
- PrecipMM: preciptação de chuva no dia
- Moon_ilumination: Índice de iluminação da lua

- DewPointC: Corresponde aos Pontos Orvalho, que é a temperatura no qual o vapor de água no ar passa ao estado líquido
- HeatIndexC: Índice de calor, uma medida importante para identificar o calor sentido no dia
- WindChillC: Temperatura sentida de acordo com o vento
- CloudCover: Indice de cobertura no céu que as nuvens tiveram
- Pressure: Pressão do ar
- TempC: temperatura registada no dia
- Visibility: Índice de visibilidade do ambiente
- WindspeedKmph : velocidade do vento em km/h
- UvIndex: indice dos raios ultravioleta

Após verificar as variáveis que o dataset possui, foi verificado também qual é o seu comportamento. Para este fim, foi utilizado o nodo "*Rank Correlation*" para verificação do relacionamento entre as variáveis, e também o nodo "*Statistics*" com o objetivo de verificar as estatísticas das mesmas.





Após a análise do comportamento de cada variável, é possível identificar quais possuem aspetos mais interessantes, variação de valores consideráveis e correlação com outras variáveis. Seguindo estes critérios o grupo atribuiu uma maior relevância para o decorrer do projeto às seguintes variáveis:

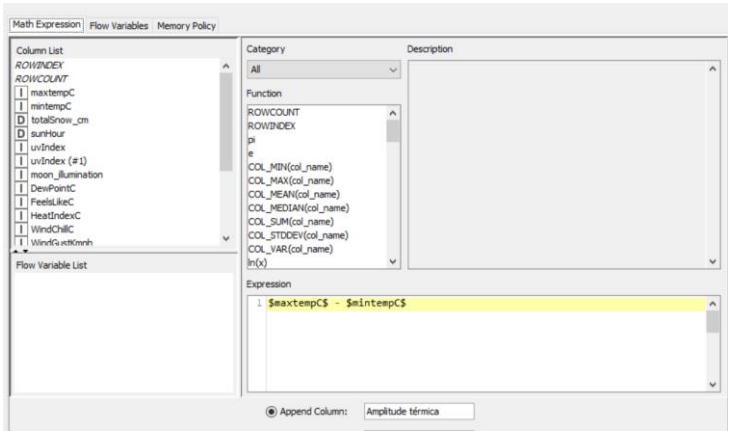
- date_time
- MintempC
- MaxtempC
- FeelslikeC
- Humidity
- PrecipMM
- DewPointC
- WindChillC
- UvIndex
- TempC
- HeatIndex
- WindSpeedkmph

Preparação dos dados

Após ter sido realizada a análise dos dados, foi realizado o tratamento destes a fim de ser possível criar modelos que obtenham os melhores resultados possíveis.

Primeiramente é necessário verificar se há valores em falta na tabela, os chamados *missing values,* se houver retirá-los pois são danosos para os resultados. Na amostra escolhida não se observou nenhum *missing value* em nenhum registo, observado no nodo "*Statistics"* anterior.

Observou-se pela análise dos dados que os parâmetros **minTempC** e **maxTempC** possuem uma diferença grande registada em cada dia, assumindo a hipótese de em Madrid existir grandes amplitudes térmicas. A fim de verificar com maior detalhe esta característica, foi criado uma nova coluna 'Amplitude térmica' com auxílio do nodo "*Math Formula"*.



A fim de obter uma análise melhor do comportamento das variáveis ao decorrer de datas diferentes, e poder assim, manipular a data especificada, foi utilizado os nodos "*String to Date&Time"*, aplicado no campo **date_time** , e o nodo "*Extract Date&TimeFields"*, para extrair o dia, mês e ano.

Após obter os campos relacionados às datas e à Amplitude térmica do registo, o grupo realizou análises ao comportamento das variáveis por ano e por mês. Para atingir este fim foi utilizado o nodo "*GroupBy"*.

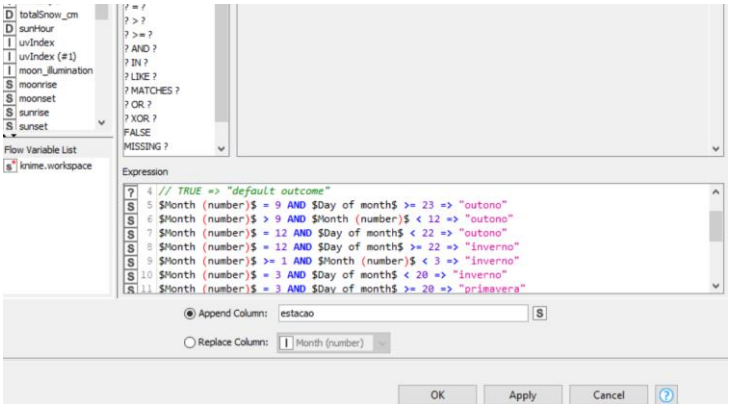
Row ID	Year	Mean(humidity)	Mean(pressure)	Mean(Amplitude térmica)	Mean(uvIndex)	Mean(FeelsLikeC)	Mean(maxtempC)	Mean(mintempC)	Mean(DewPointC)	Mean(HeatInde...	Mean(Wir
Row0	2008	56.495	1,017.44	10.978	4.435	15.685	21.098	10.12	5.848	16.549	15.799
Row1	2009	57.589	1,016.444	11.408	4.304	14.6	20.362	8.953	5.345	15.564	14.704
Row2	2010	62.299	1,015.049	10.778	3.975	13.164	18.904	8.126	5.438	14.255	13.214
Row3	2011	60.816	1,018.463	10.795	4.205	14.852	20.058	9.263	6.573	15.63	14.899
Row4	2012	58.273	1,018.503	10.749	4.167	13.88	19.301	8.552	5.21	14.885	13.956
Row5	2013	59.025	1,017.679	10.51	4.134	13.578	19.142	8.633	5.104	14.732	13.644
Row6	2014	59.09	1,016.54	10.43	4.304	14.718	20.079	9.649	6.151	15.668	14.778
Row7	2015	51.942	1,019.638	10.501	4.49	16.016	21.249	10.748	5.132	16.756	16.164
Row8	2016	56.918	1,018.333	10.309	4.317	15.049	20.336	10.027	5.732	15.877	15.145
Row9	2017	50.447	1,019.312	10.496	4.605	16.592	21.551	11.055	5.205	17.282	16.729
Row10	2018	52.362	1,016.967	8.479	4.49	15.773	20.129	11.649	5.436	16.638	15.918
Row11	2019	51.218	1,021.709	8.855	3.736	10.655	16.527	7.673	1.109	11.773	10.655

File Edit Hilite Navigation View

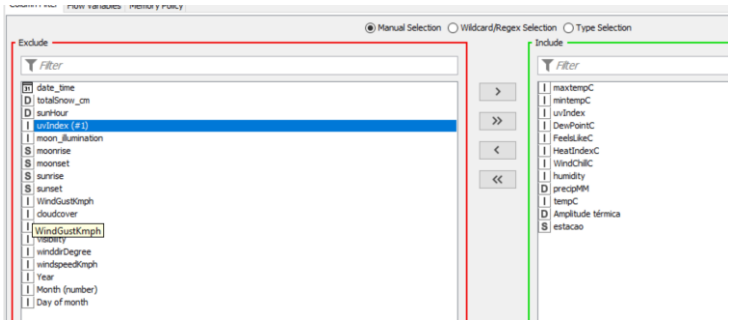
Table "default" - Rows: 12 Spec - Columns: 11 Properties Flow Variables

Month ...	Mean(FeelsLikeC)	Mean(humidity)	Mean(WindChillC)	Mean(uvIndex)	Mean(Amplitude térmica)	Mean(tempC)	Mean(HeatIndexC)	Mean(DewPointC)	Mean(maxtempC)	Mean(mintempC)
1	4.39	73.739	4.39	2.381	7.692	9.836	6.27	1.463	9.836	2.144
2	4.819	68.803	4.819	2.603	8.658	11.181	7.055	1.081	11.181	2.523
3	8.475	64.783	8.475	3.211	10.519	14.9	10.106	2.862	14.9	4.381
4	11.888	63.294	11.888	3.725	11.381	18.094	13.025	5.244	18.094	6.712
5	16.997	54.152	17.01	4.71	12.016	22.6	17.519	6.987	22.6	10.584
6	23.213	42.813	23.407	5.87	12.18	28.327	23.337	8.643	28.327	16.147
7	27.05	33.199	27.452	6.625	12.935	32.208	27.073	8.226	32.208	19.273
8	27.035	33.622	27.443	6.625	13.018	32.276	27.012	8.437	32.276	19.258
9	22.227	43.579	22.303	5.603	11.842	27.024	22.336	8.048	27.024	15.182
10	16.513	57.724	16.504	4.49	9.768	20.768	16.938	7.284	20.768	11
11	8.933	71.27	8.933	3.17	7.661	13.445	10.258	4.764	13.445	5.785
12	5.61	73.085	5.61	2.513	7.651	10.334	7.141	2.235	10.334	2.683

Após realizar a análise das tabelas acima, concluiu-se que as variáveis em cada ano no geral não variaram muito e possuem valores muito próximos. Em relação ao agrupamento realizado por mês, foi verificado um comportamento mais interessante, existe uma variação entre as variáveis dependendo mês registado. Outra análise verificada acerca da tabela dos agrupamentos por mês, foi de que os valores têm tendência a variarem de acordo com a época do ano que estão. A partir disso, para ser possível analisar o registo de acordo com a estação do ano, o que é mais interessante no caso estudado do que uma data específica, foi criado uma nova coluna para indicar a correspondente estação do ano de cada registo. Para atingir tal objetivo, foi utilizado o Nodo "*Rule Engine*".



Após foi realizado o filtro das colunas que se mostraram mais relevantes no caso estudado, após análises estatísticas, correlações e das que foram consideradas importantes para criação dos modelos. Vale informar que para chegar a estas colunas houve diversas mudanças a partir de testes realizados aos modelos.



Modelação

Após se ter realizado a análise e tratamento dos dados, foi possível criar modelos de aprendizagem. Entretanto, como o *dataset* não veio associado nenhum problema, como mencionado anteriormente, houve a necessidade e a possibilidade de criar problemas. Com o objetivo de diversificar e colocar em prática uma maior quantidade de conhecimentos aprendidos nas aulas, o grupo criou um problema de classificação, e outro de regressão.

Problema de Classificação

O grupo definiu a previsão do campo de estação de ano como problema de classificação. Chegou-se a este problema através da análise realizada anteriormente das variáveis que tinham valores distintos em determinados meses e épocas do ano. Realizou-se uma outra análise para verificar com maior detalhe a variação dos dados por estação do ano, e por fim foi utilizado agrupamentos de dados a partir do nodo "*GroupBy*".

Table "default" - Rows: 4 | Spec - Columns: 11 | Properties | Flow Variables

...	S estacao	D Mean(FeelsLikeC)	D Mean(humidity)	D Mean(WindChillC)	D Mean(HeatIndexC)	D Mean(Amplitude térmica)	D Mean(maxtempC)	D Mean(mintempC)	D Mean(uvIndex)	D Mean(DewPointC)	D Mean(t
...1	inverno	5.476	70.053	5.476	7.388	8.599	11.333	2.734	2.629	1.653	11.333
...2	outono	11.752	65.317	11.748	12.723	8.644	16.226	7.582	3.653	5.396	16.226
...3	primavera	15.213	56.963	15.24	16.005	11.596	21.022	9.426	4.377	6.235	21.022
...4	verao	25.983	35.478	26.313	25.998	12.78	31.118	18.338	6.397	8.262	31.118

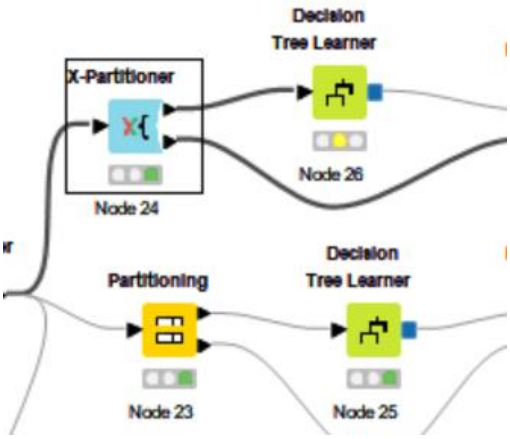
A partir da tabela acima, é possível verificar que no verão e no inverno as variáveis possuem comportamentos muito distintos, e a primavera e o outono têm valores intermédios e próximos um do outro. Como todo problema de *machine learning* surge de uma hipótese criada a partir de observações e estudos, o grupo chegou à seguinte hipótese: "A cidade de Madrid possui 4 estações bem definidas?".

Com o objetivo de testar esta hipóteses definida foram criados modelos a partir de duas técnicas de classificação: *Decision Tree Learner* e *Clustering* .

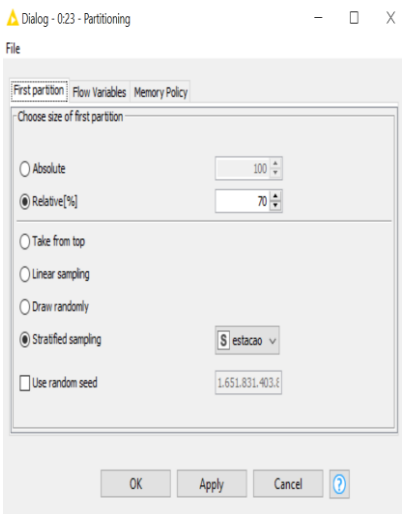
- **Decision Tree Learner**

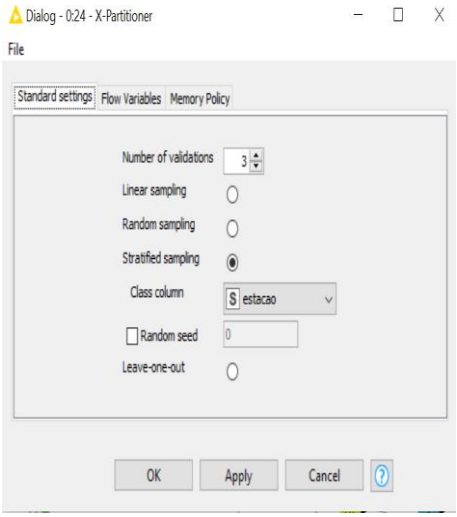
Nesta técnica foi utilizado o modelo com supervisão, e com o objetivo de validar e analisar se o modelo obteve resultados satisfatórios, é necessário ter dados de treino, que servirão para a criação do modelo, e dados de teste, que terão a utilidade de testar o modelo criado.

A fim de cumprir este requisito, foi utilizado duas técnicas de separação de dados, *X-Partitioner* e *Partitioning*.



Foi também utilizado como critérios de separação no nodo *X-partitioner* 3 validações, pelo motivo de ser pouca quantidade de dados e ter sido o que apresentou melhores resultados, e para o modo com que é escolhido os dados foi utilizado o *Stratified Sampling* para a coluna 'estação', de forma que os dados sejam escolhidos de igual forma para cada estação, para assim evitar enviesamento, o chamado *bias* problema muito encontrado no estudo de *machine learning*. E para o *Partitioning* foi utilizado a taxa de 70% para dados de treino e o restante para teste, com a justificação de apresentar melhores resultados e também por serem poucos dados.





Avaliação dos resultados

Nos parâmetros utilizados no nodo "*Decision Tree Learner"* foi utilizado como *target* o campo estação, por ser a coluna objetivo, ou seja, a coluna que se pretende prever, e como *quality measure,* o *Gain ratio* e *pruning MDL*,uma vez que conseguem produzir melhores resultados.

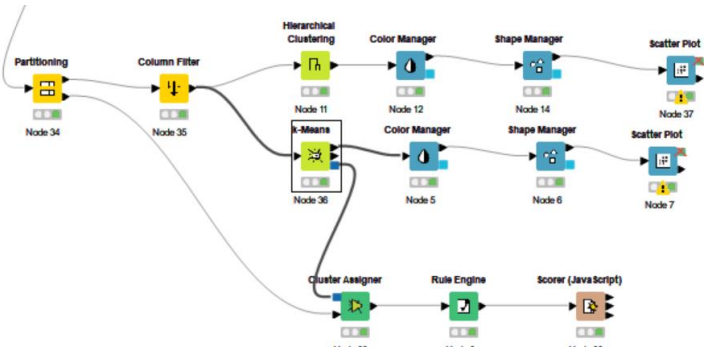
- **Clustering**

Foi utilizado a técnica *clustering* com aprendizagem sem supervisão, com objetivo de verificar se é possível categorizar os dados e separar eles de acordo com seu comportamento.

Dentre as técnicas de *clustering* foram utilizadas: *k-means* e *Hierarchical Clustering*. Especificou-se 4 clusters em cada modelo, sabendo que havia 4 estações, e assim é possível verificar se os dados estão bem distribuídos, e assim comprovar se é possível criar 4 categorias com comportamento diferente.

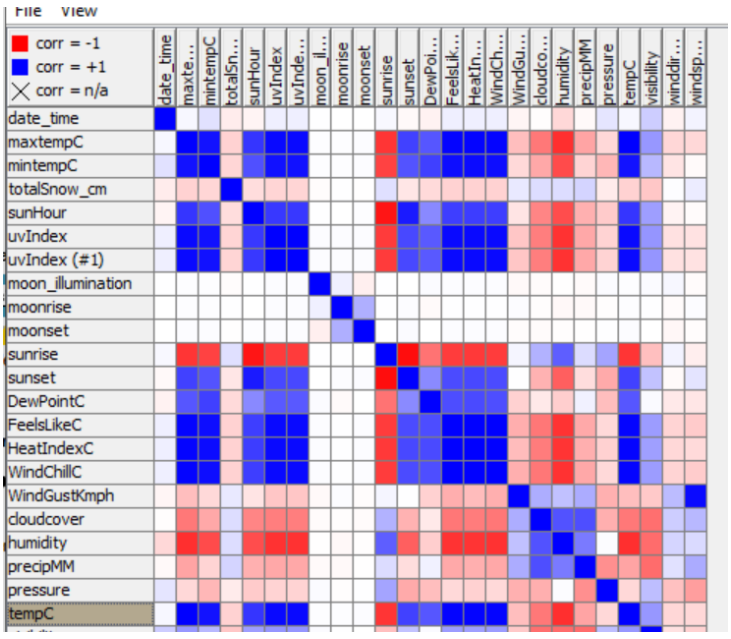
Foi utilizado o "*Partitioning*" com as mesmas configurações utilizadas anteriormente, a fim de testar o modelo *k-means* a partir do nodo "*cluster assigner"*. Para verificar se o cluster atribuído teve sucesso ou não de acordo com a estação do ano, utilizou-se o "*rule engine"* para atribuir ao nome do *cluster* a estação do ano correspondente, para assim conseguir comparar de forma efetiva os parâmetros de treino e teste.

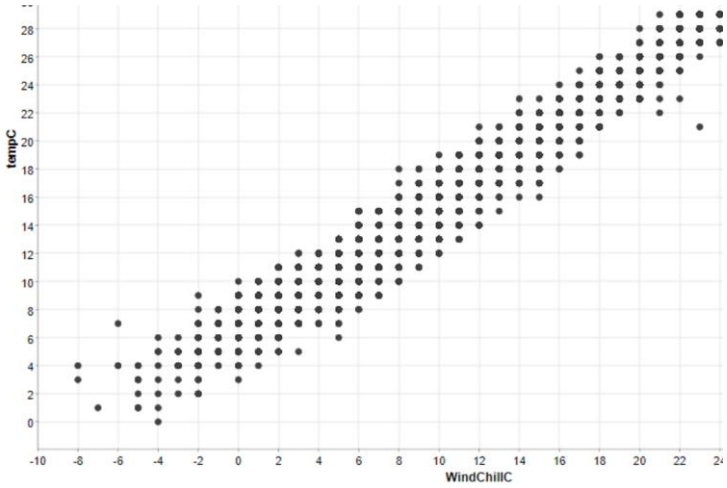
Na imagem a seguir mostra como ficou o *pipeline*.

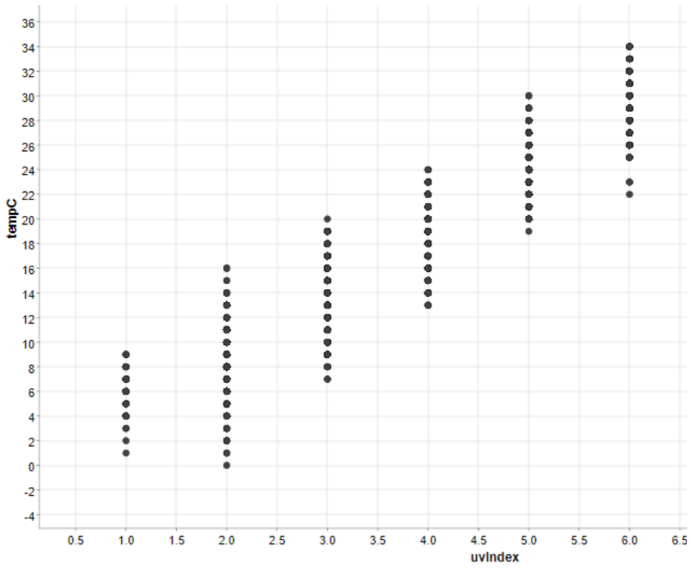


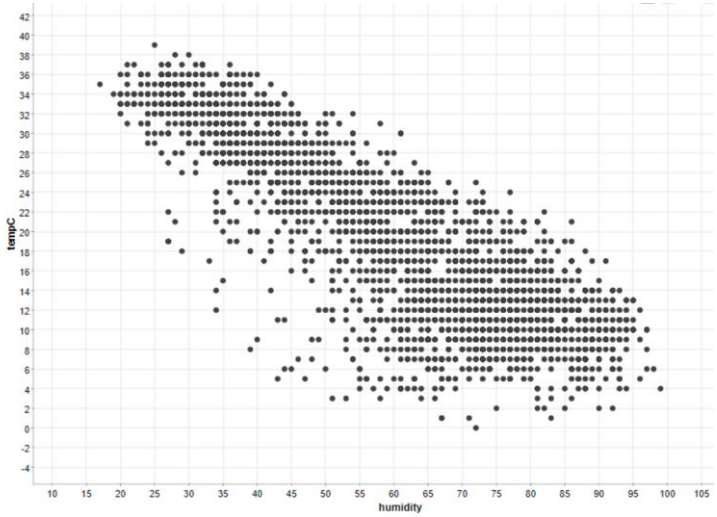
Problema de Regressão

Criou-se um problema de regressão, escolhendo como variável dependente e aquela que determinamos prever a coluna 'tempC', que corresponde à temperatura registada no dia. Primeiramente foi realizado uma análise na matriz de correlação para verificar aquelas variáveis que possuíam maior relação à variável dependente . Após observou-se o comportamento de algumas das quais possuíam maior relação, através dos gráficos resultantes do nodo Scatter Plot.




tempC
WindChillC

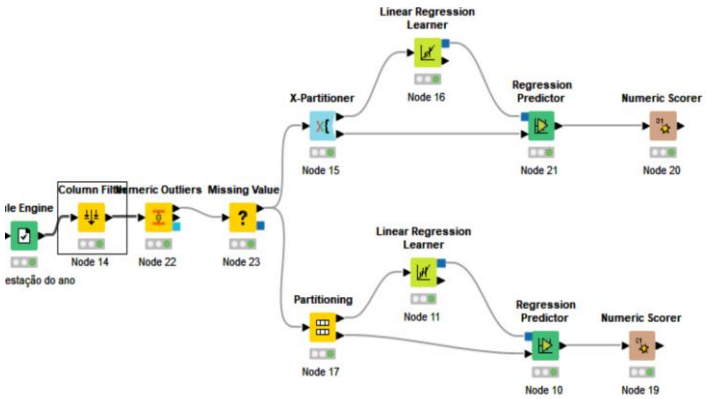

tempC
uvIndex



A partir destas observações e análises, observou-se que o campo 'tempC' possui um comportamento linear em detrimento da variação de outras variáveis. Dito isto, criou-se um modelo *Linear Regression*, a fim de verificar se um modelo baseado na equação linear consegue atingir índices de previsão satisfatórios para a variável fixa, no caso o campo 'tempC'.

Para atingir este objetivo o grupo realizou mais alguns tratamentos nos dados na tentativa de obter os melhores resultados no modelo, o que inclui a remoção do *outlier* e em seguida remoção dos valores em falta. Foi também utilizado as técnicas de separação de dados ie *partitioning* para o teste do modelo, utilizando basicamente as mesmas configurações dos modelos já vistos anteriormente.

Obtendo-se assim o seguinte pipeline:



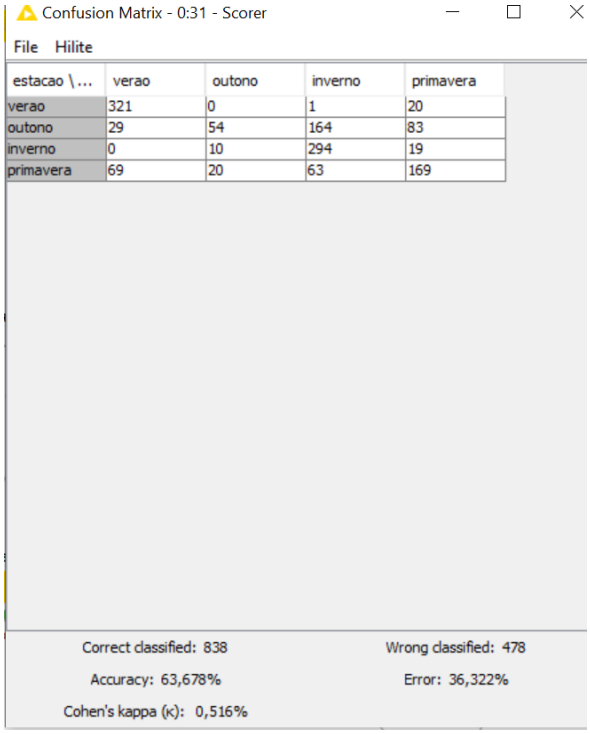
Avaliação dos Resultados

A análise dos resultados produzidos pelos modelos serão divididos separadamente por problema, o problema de Classificação e de Regressão.

Problema de Classificação

- **Resultados**

Os resultados produzidos pelos modelos de Decision Tree, a partir dos nodos X-partitioner e Partitioning consequentemente:



estacao \...	verao	outono	inverno	primavera
verao	321	0	1	20
outono	29	54	164	83
inverno	0	10	294	19
primavera	69	20	63	169

Confusion Matrix - 0:30 - Scorer

File Hilite

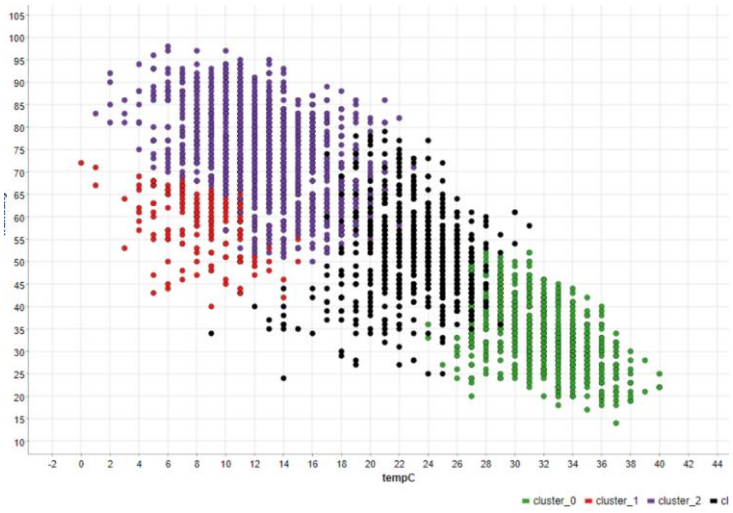
estacao \...	verao	outono	inverno	primavera
verao	276	8	0	23
outono	12	83	109	93
inverno	4	26	233	28
primavera	61	28	50	150

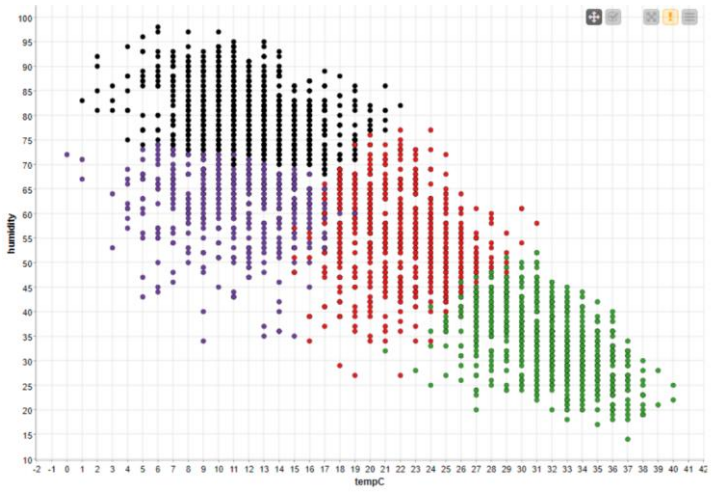
Correct classified: 742 Wrong classified: 442

Accuracy: 62,669% Error: 37,331%

Cohen's kappa (κ): 0,502%

Gráficos produzidos pelos modelos de clustering utilizando as variáveis 'tempC' e 'humidity', por hierarchical clustering e k-means, consequentemente:





E o resultado produzido pelo scorer a partir do modelo k-means:

Scorer View

Confusion Matrix

	inverno (Predicted)	outono (Predicted)	primavera (Predi...	verao (Predicted)	
inverno (Actual)	148	122	21	0	50.86%
outono (Actual)	132	48	94	23	16.16%
primavera (Actual)	51	30	156	52	53.98%
verao (Actual)	1	0	29	277	90.23%
	44.58%	24.00%	52.00%	78.69%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
53.13%	46.88%	0.375	629	555

- **Análise**

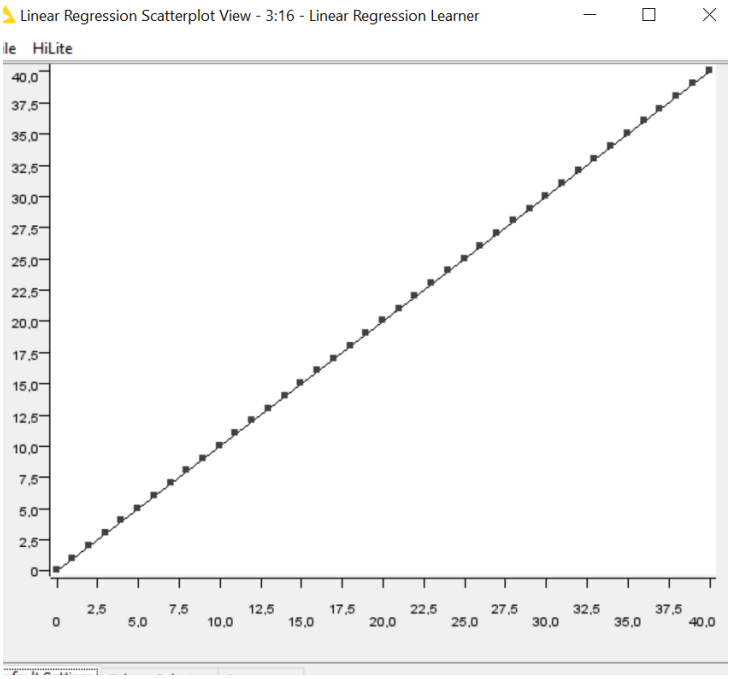
A partir dos resultados apresentados é possível inferir alguns aspetos acerca do clima de Madrid e refutar a hipótese do problema realizado anteriormente. Observa-se que nas estações de inverno e verão há uma grande distinção de variáveis climáticas, deduz-se a partir do alto índice de acerto nas previsões e partir da análise de dados feitas. Entretanto, já as estações de primavera e outono possuem valores semelhantes as outras estações sendo de difícil acerto, principalmente o outono onde obteve-se índices de acerto muito baixos.

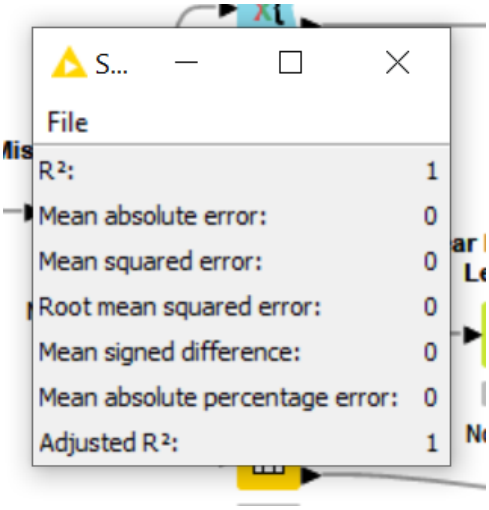
Conclui-se, que a partir dos dados utilizados deste *dataset* do clima de Madrid, que a hipótese que Madrid possui 4 estações bem definidas não se confirma, entretanto, apresenta 2 estações muito bem definidas, sendo que as outras estações possuem valores mais semelhantes às demais.

Problema de Regressão

- **Resultados**

Os resultados produzidos pelo modelo linear regression:





Vale destacar que os resultados tanto do X-Partitioning quanto do Partitioning foram iguais.

- **Análise**

Verifica-se pelos resultados que o modelo teve 100% de eficácia com o R-Square Value = 1, correspondendo ao entendimento da variância de todos os campos, e com o Root Mean Square Error (RMSE) = 0, ou seja, não houve nenhum erro apresentado.

Conclui-se que a análise realizada da linearidade e correlação das variáveis selecionadas em detrimento ao campo tempC , estavam corretas, e que desta forma implica-se que a temperatura de Madrid possui valores que são relacionados a outros fatores climáticos também.

SALARY CLASSIFICATION

Para a análise dos dados deste dataset, o grupo optou por seguir a metodologia CRISP-DM. Por se tratar de um projeto com fins académicos, decidiu-se dar prioridade às seguintes etapas da metodologia: estudo dos dados, preparação dos dados, modelação e avaliação.

Estudo dos dados

O *dataset salary classification* possui dados referentes a 48842 registos distintos, e contém 15 campos de informação (colunas) são estas:

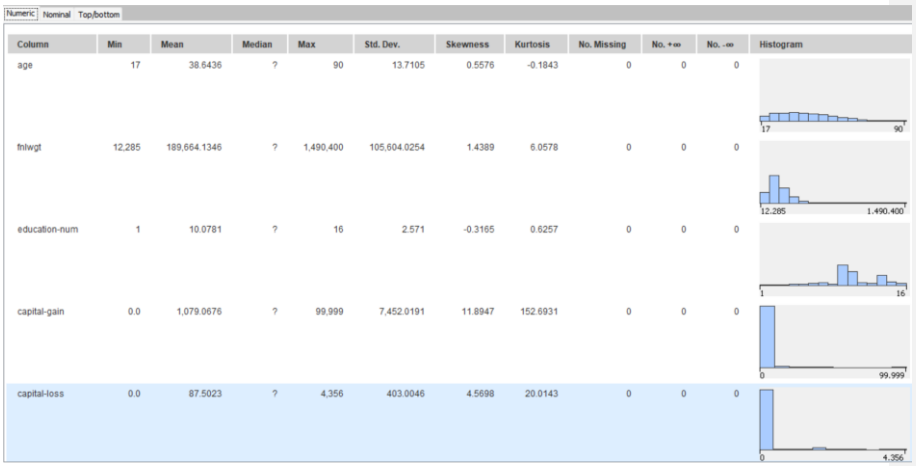
- *age*: idade de um indivíduo. Inteiro maior que 0.
- *workclass*: termo geral que representa a categoria de trabalho de um indivíduo.
- *fnlwgt*: peso final. Em outras palavras, número de pessoas as quais esse registo representa. Inteiro maior que 0.
- *education*: o nível mais alto de educação alcançado por um indivíduo.
- *education-num*: o nível mais alto de educação alcançado por um indivíduo em forma numérica. Inteiro maior que 0.
- *maritalstatus*: marital status of an individual.
- *occupation*: tipo geral de ocupação profissional de um indivíduo.
- *relationship*: representa o que esse indivíduo é em relação a outros. Cada entrada tem apenas um atributo de relacionamento e se confunde em muito com a coluna *maritalstatus*.
- *race*: descrição da etnia de um indivíduo.
- *sex*: sexo biológico de um indivíduo.2 valores únicos possíveis.
- *capitalgain*: ganho de capital para um indivíduo.
- *capitalloss*: Perda de capital para um indivíduo.
- *hoursperweek*: horas de trabalho um indivíduo relatou ter trabalhado.
- *nativecountry*: país de origem de um indivíduo.
- *the label*: Rótulo que indica se um indivíduo ganha anualmente mais de $50,000. Os únicos valores possíveis são *<=50k* e *>50k*.

Objetivos

Com a informação disponibilizada o grupo decidiu aplicar os conhecimentos aprendidos sobre o paradigma de aprendizagem com supervisão, com o objetivo de classificar se um indivíduo ganha anualmente mais de $50,000 ou não.

Preparação dos dados

O *dataset* não possui valores em falta. No entanto, apresenta algumas colunas redundantes, como as colunas *education* e *education-num* que indicam a mesma informação, sendo que a primeira de forma textual e a segunda de forma numérica. Com a limpeza dos dados em mente, filtrou-se a coluna *education*. Algo semelhante ocorre com as colunas *relationship* e *marital-status*, onde se utilizou o nodo "*Rule Engine"* para se alterar os valores da coluna *marital status* de forma que a coluna *relationship* se torne completamente redundante, portanto descartável considerando o objetivo descrito acima. Ao analisar os histogramas das colunas numéricas, através do nodo "*Statistics"*, notou-se uma particularidade com as colunas '*capital-gain'* e '*capital-loss'* ambas tinham como maioria absoluta o zero como valor, e níveis de *skewness* muito elevados, 11.8947 e 4.5698, respetivamente, como se pode observar na figura abaixo. As colunas filtradas até ao momento são: '*education', 'education-num', 'capital-gain'* e *'capital-loss'*.


Numeric Nominal Top/bottom

Column	Min	Mean	Median	Max	Std. Dev.	Skewness	Kurtosis	No. Missing	No. +∞	No. -∞	Histogram
age	17	38.6436	?	90	13.7105	0.5576	-0.1843	0	0	0	17 90
fnlwgt	12,285	189,664.1346	?	1,490,400	105,604.0254	1.4389	6.0578	0	0	0	12,285 1,490,400
education-num	1	10.0781	?	16	2.571	-0.3165	0.6257	0	0	0	1 16
capital-gain	0.0	1,079.0676	?	99,999	7,452.0191	11.8947	152.6931	0	0	0	0 99,999
capital-loss	0.0	87.5023	?	4,356	403.0046	4.5698	20.0143	0	0	0	0 4,356

Por se tratar de apenas um ficheiro *csv*, portanto apenas uma fonte de dados, não houve a necessidade se aplicar técnicas de integração.

Os dados também passam pelos nodos *Category to Number* e *Normalizer,* que respectivamente transformam os valores textuais que representam categorias em números e normalizam os valores da categoria *fnlwgt* para melhor processamento dos mesmos pelos nodos de aprendizagem.

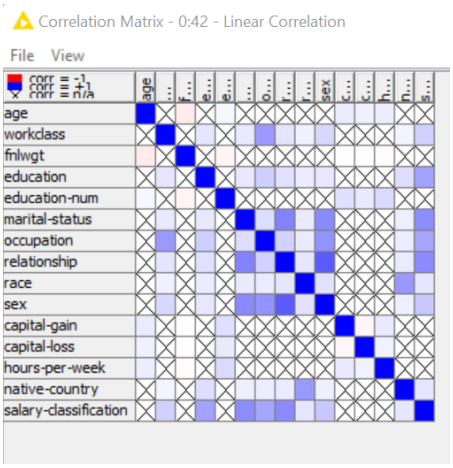
Modelação

Para este *dataset* foi tomada a decisão de se utilizar duas técnicas de aprendizagem. Um *workflow Knime* foi criado utilizando árvores de decisão de tipo discreto e o outro utilizando redes neuronais.

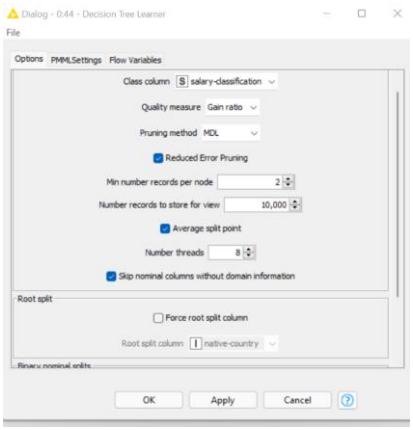
Modelos com a utilização de árvores de decisão

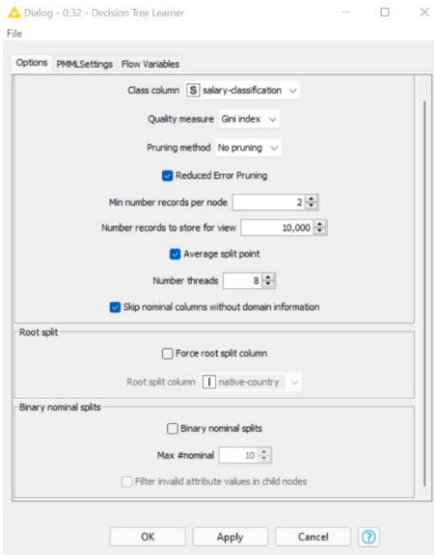
O grupo aplicou a técnica de árvores de decisão de tipo discreto por ter em vista que o atributo de decisão (*salary-classification*) se divide em duas possíveis categorias se um indivíduo ganha ou não anualmente mais de $50,000.

Esta técnica de aprendizagem segue o paradigma *bottom up*, ou seja, modelo é construído pela identificação das relações entre os atributos do *dataset*. Esta técnica foi escolhida pela sua fácil compreensão e configuração, e atrelado ao fato de haver poucos atributos com níveis significativos de correlação entre si (ver figura abaixo), não apresenta vulnerabilidades muito expressivas.



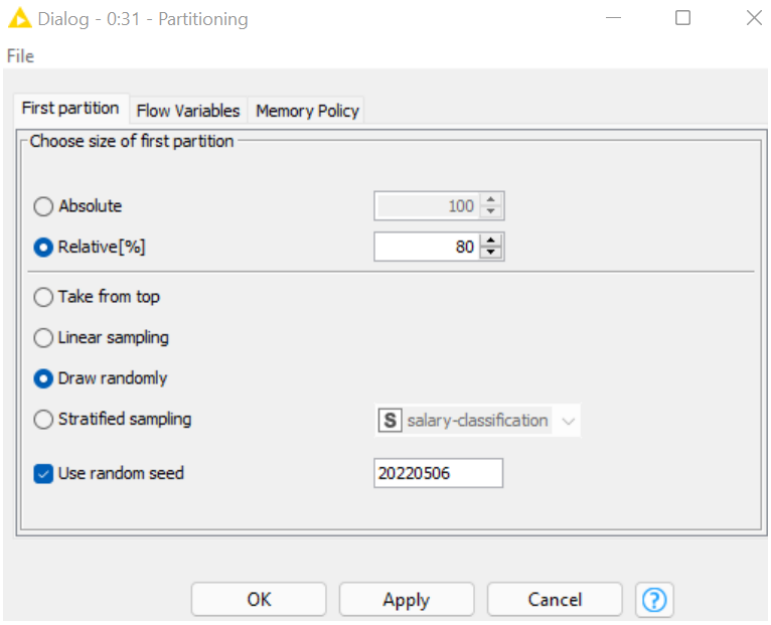
O grupo decidiu fazer dois modelos utilizando árvores de decisão, um utilizando como métrica de qualidade *gain ratio* e como método *MDL* de *pruning* e o outro utilizando o *gini index* sem métodos de *pruning*. Através do pruning o tamanho das árvores de decisão é reduzido ao remover partes da árvore que são redundantes e não importantes para aprendizagem, tornando-se assim possível prevenir *overfitting* do modelo aos dados aumentando a precisão da capacidade de previsão.



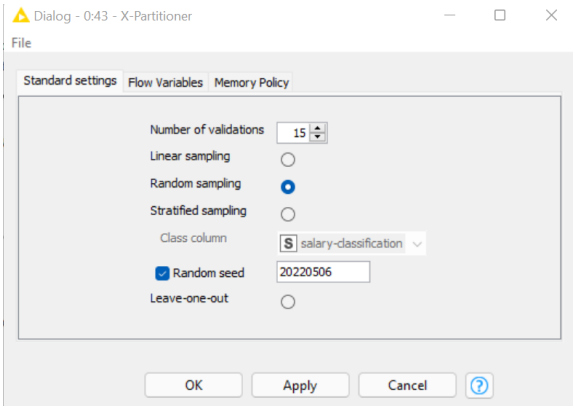


Após a preparação dos dados, o grupo recorreu a duas técnicas de *partitioning*, o nodo *"Partitioner"* e o nodo "*X-Partitioner"*.

O nodo "*Partitioner"* divide os dados em duas partições, uma com 80% dos dados de forma aleatória com recurso a uma *random seed* que garante a possibilidade de refazer as partições com os mesmos dados, visto que o tamanho se mantenha. A partição com 80% dos dados é então passada para o nodo "*Decision Tree Learner"* como mencionado anteriormente, enquanto o restante dos dados é passado junto com modelo gerado pelo nodo "*Decision Tree Learner"* para o nodo "*Decision Tree Predictor* "que testa o modelo utilizando os dados recebidos.



Já o nodo "*X-Partitioner*", efetua a validação por cruzamento de dados, dividindo o conjunto de dados em *k folds*, onde durante *k* interações e em cada interação são utilizadas *k*-1 *folds* para treino e 1 para teste, neste caso em concreto o valor de *k* é 15, ver figura abaixo para a configuração do nodo.



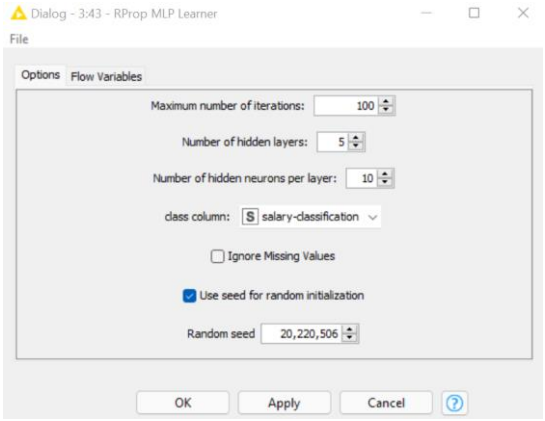
Modelo com redes neuronais

Uma Rede Neuronal Artificial é um sistema computacional de base conexionista inspirado no sistema nervoso central dos seres humanos, definido por uma estrutura interligada de unidades computacionais com capacidade de aprendizagem, tais unidades são designadas neurónios. Redes neuronais podem ser construídas de acordo com 3 tipos de arquitetura distintos, para este modelo foi escolhido a arquitetura *Feed-Foward* multicamada, arquitetura organizada por camadas, podendo

possuir 1 ou mais camadas intermédias, com conexões unidirecionais, a cada camada adicional maior é a capacidade da rede em modelar funções de maior complexidade.

Os dados são particionados com o nodo "*Partitioner"* já previamente explicado, devido a simplicidade deste tipo de validação tendo em vista o aumento na complexidade da técnica de aprendizagem, o nodo é configurado de forma semelhante ao modelo anterior.

O nodo "*RProp MLP Learner"* foi configurado de forma a permitir um extensivo processo de aprendizagem com um razoável número de camadas e número de neurónios por camadas, ao passo que o tempo de execução do nodo não se estenda demasiadamente.



Avaliação

Para avaliação dos resultados o nodo "*Scorer(Javascript)"* foi utilizado para uma mais agradável visualização da matriz de confusão e das estatísticas gerais do modelo.

A matriz de confusão permite um melhor entendimento da precisão do modelo.

Modelo por *Decision Tree, gini index* sem *pruning*

A matriz de confusão produzida por este modelo apresenta uma alta precisão ao classificar corretamente dados que ganham até $50,000 por ano, no entanto essa precisão cai ao classificar dados que ganhem mais de $50,000 por ano. No entanto produzindo uma precisão geral razoável de 79.19% da capacidade do modelo. *Cohen's kappa* é uma outra métrica que pode ser utilizada para se medir a performance de um modelo, com uma gama de valores entre -1 e 1, para dados mais balanceados o *Cohen's kappa* tende a aumentar.

Scorer View

Confusion Matrix

	<=50K (Predicted)	>50K (Predicted)	
<=50K (Actual)	6482	991	86.74%
>50K (Actual)	1042	1254	54.62%
	86.15%	55.86%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
79.19%	20.81%	0.417	7736	2033

Modelo por *Decision Tree, gain ratio* com *pruning MDL*

A matriz de confusão produzida por este modelo apresenta uma alta precisão ao classificar corretamente dados que ganham até $50,000 por ano, no entanto essa precisão cai ao classificar dados que ganhem mais de $50,000 por ano, contudo menos do que o apresentado pelo modelo anterior que não aplica um método de *pruning*. Não obstante produzindo uma precisão geral razoável de 83.39% para este modelo, o melhor resultado entre os modelos utilizados. Também possui o maior *Cohen's Kappa* dentre os 3 modelos utilizados.

Scorer View

Confusion Matrix

	<=50K (Predicted)	>50K (Predicted)	
<=50K (Actual)	2223	220	90.99%
>50K (Actual)	321	493	60.57%
	87.38%	69.14%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
83.39%	16.61%	0.538	2716	541

Modelo por *Artificial Neural Networks*

A matriz de confusão produzida por este modelo apresenta uma maior diferença ao classificar corretamente os valores, tendo precisão muito baixa, 42.86% ao classificar corretamente se um registo ganha mais de $50,000 por ano. No entanto produziu uma precisão geral razoável de 81.77% de capacidade de previsão do modelo.

Scorer View

Confusion Matrix

	<=50K (Predicted)	>50K (Predicted)	
<=50K (Actual)	7004	469	93.72%
>50K (Actual)	1312	984	42.86%
	84.22%	67.72%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
81.77%	18.23%	0.419	7988	1781

Sugestões e Recomendações

Após análise dos resultados obtidos com os modelos envolvendo o *dataset salary_classification,* a técnica de aprendizagem mais robusta, redes neuromais, não necessariamente produziu o melhor resultado. A fim de melhorar os resultados obtidos nomeadamente com as redes neuronais podia-se ter recorrido ao uso de técnicas de *feature selection*, que iriam indicar as features mais bem qualificadas para aprendizagem autónoma presentes neste dataset.