

Sistemas Baseados em Similaridade

Enunciado Prático Individual 5

José Virgílio Silva Loureiro (PG52252)

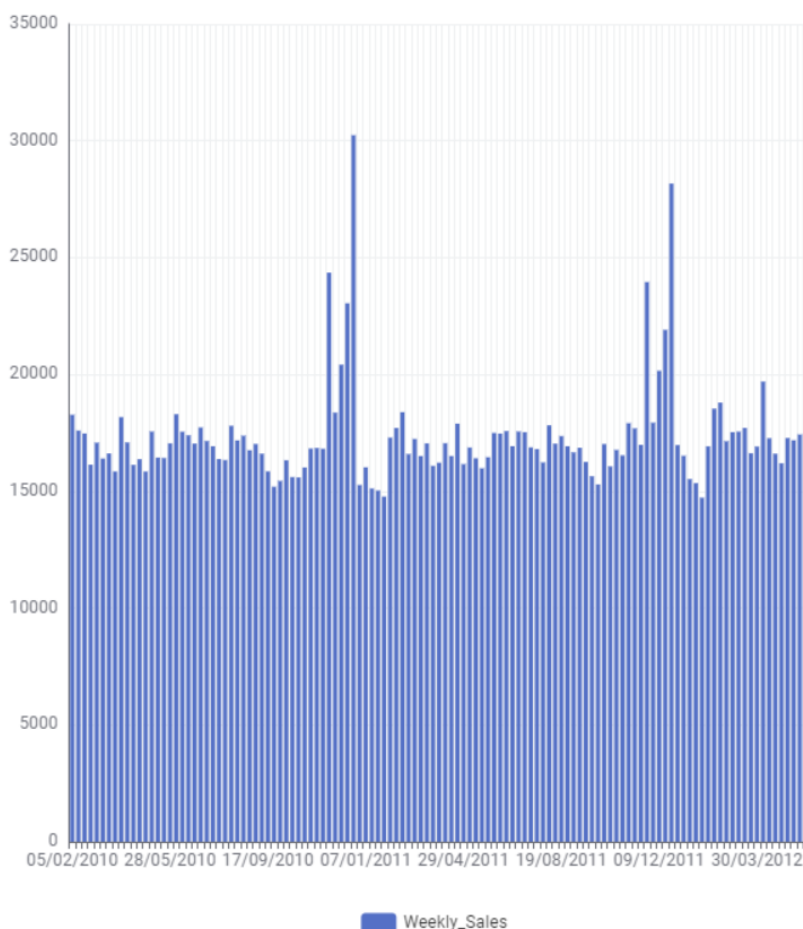
T1. Nesta tarefa carreguei no Knime os respetivos datasets. Utilizei o nodo “Table Reader” para ler o conjunto de dados que contém informação sobre cada uma das lojas, incluindo o seu tipo e tamanho. E para o segundo dataset utilizei o nodo “CSV Reader”. De seguida apliquei um nodo “Joiner” que juntou ambos os datasets pelo atributo “Store”.

Para a exploração dos dados criei um metanode com o nome “Dados visualmente”, onde primeiramente utilizo o nodo “Data Explorer”, de onde tirei as seguintes conclusões:

- A feature que apresenta um desvio-padrão maior é o “Size”;
- As features que apresentam maior dispersão em torno da média (Variância) são o “Size” e a “Weekly_Sales”;
- A feature que apresenta maior Skewness e maior Kurtosis é a “Weekly_Sales”;
- Não existem “Missing Values”.

Depois usei o nodo Bar Chart para visualizar a média de vendas por data, onde escolhi a “Date” vs “Weekly_Sale”, assim obtive o seguinte gráfico:

Bar Chart



↓

Data

Category dimension

Date

Aggregation

☐ None ☐ Occurrence count ☐ Sum

☒ Average

Frequency dimensions

Manual Wildcard Regex Type

Search Aa

Excludes

Store
Size
Dept

Any unknown columns

Includes

Weekly_Sales

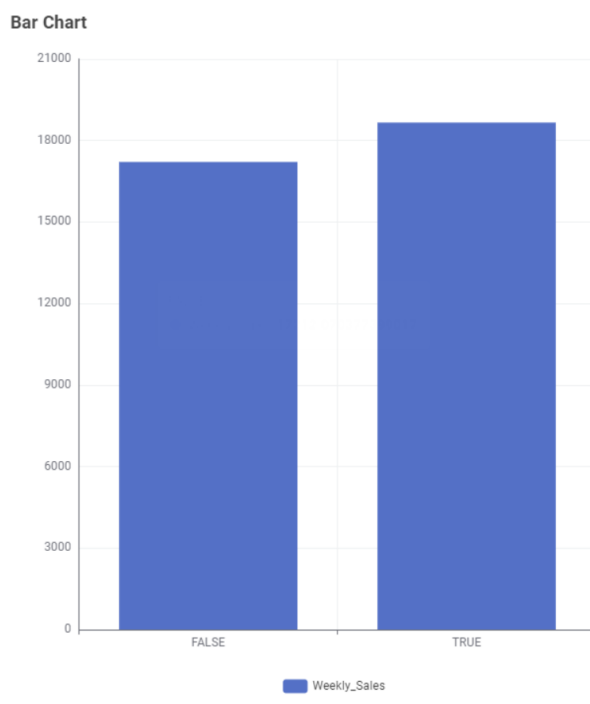
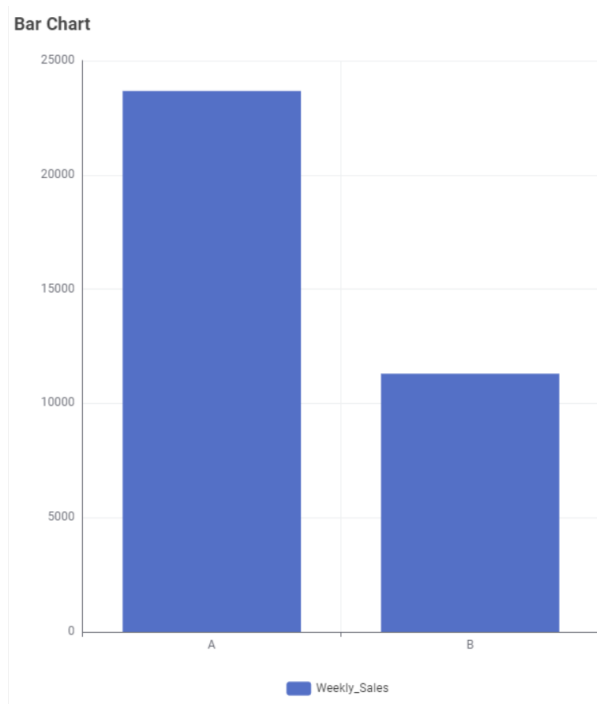
Plot

Title

Cancel Ok

Concluindo que nos dias anteriores ao Natal é onde se verifica o maior número de vendas.

Usando esse node comparei também o “Type” com a “Weekly_Sale “ e o “IsHoliday” com a “Weekly_Sale” onde obtive os seguintes gráficos:



Que me levam a concluir que o “Type” A tem médias de venda mais altas que o B e que existe também uma média de venda maiores quando existe feriados.

Para analisar os dados relativamente à correlação entre features, usei o nodo “Rank Correlation” onde obtive a seguinte tabela de correlação:

| | | | | | | | |
|--------------|-------|------|------|------|------|------|-------|
| | Store | Type | Size | Dept | Date | W... | Is... |
| Store | | | | | | | |
| Type | | | | | | | |
| Size | | | | | | | |
| Dept | | | | | | | |
| Date | | | | | | | |
| Weekly_Sales | | | | | | | |
| IsHoliday | | | | | | | |

Onde observei que existe uma clara correlação negativa entre o “Type” e o “Size”, e uma pequena correlação negativa entre o “Type” e a “Weekly_Sales”.

T2. A) Nesta tarefa transformei o atributo *isHoliday* de modo que este passasse de dar valores TRUE e FALSE e para apresentar valores 1 e 0, respetivamente para isso utilizei o nodo “Rule Engine” com as seguintes configurações:

Flow Variable List
knime.workspace

?

1

// enter ordered set of rules, e.g.:

?

2

// \$double column name\$ > 5.0 => "large"

?

3

// \$string column name\$ LIKE "*blue*" => "small and blue"

?

4

// TRUE => "default outcome"

|

5

\$IsHoliday\$ LIKE "TRUE" => 1

|

6

\$IsHoliday\$ LIKE "FALSE" => 0

Append Column:

prediction

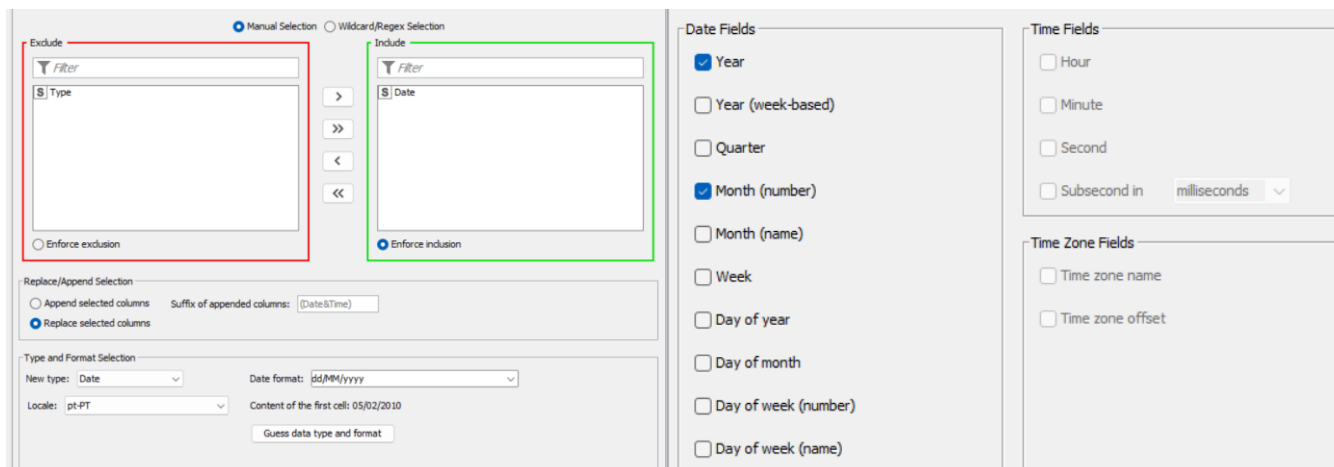
I

Replace Column:

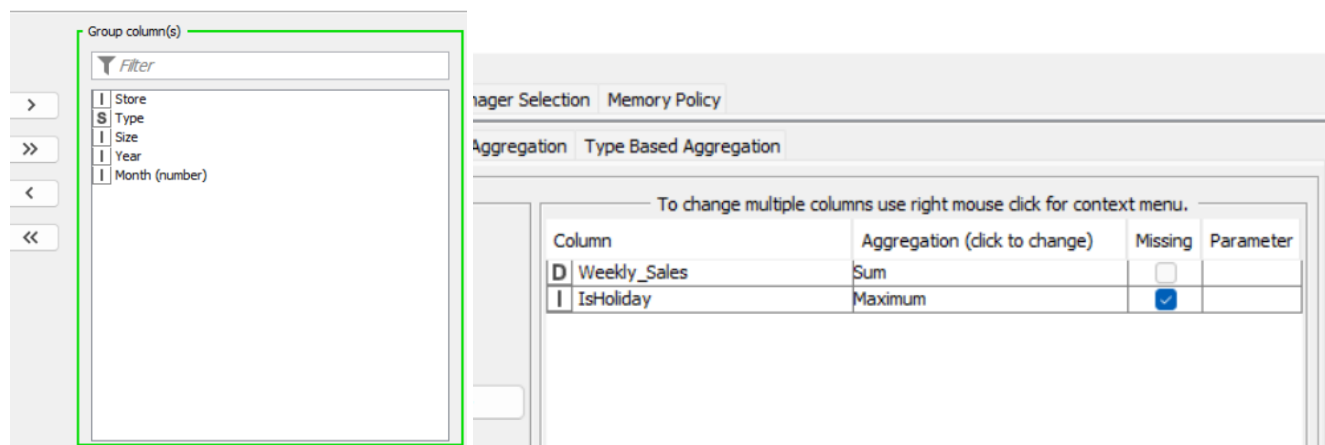
\$ IsHoliday

▼

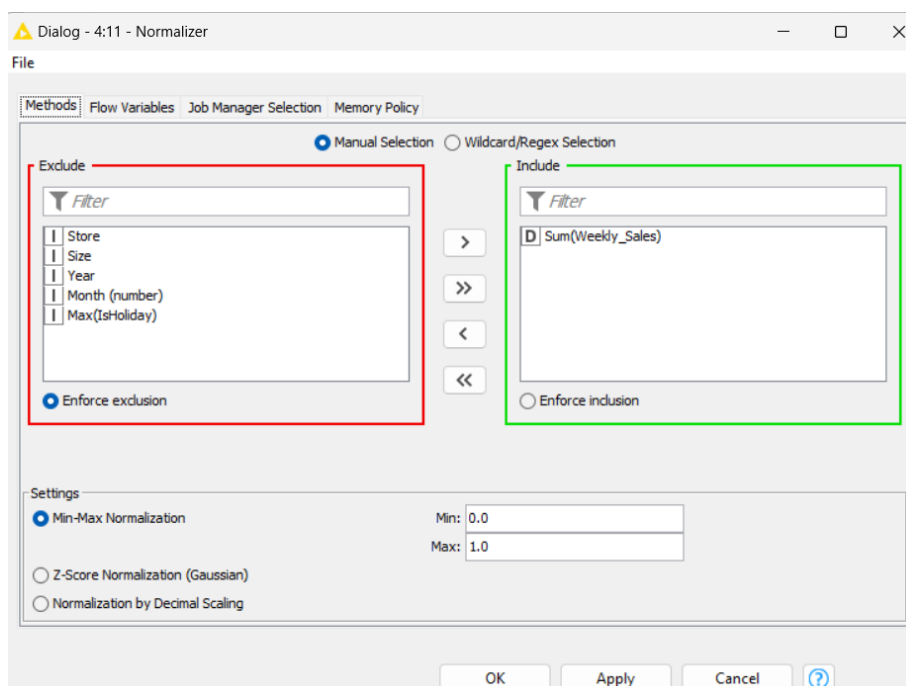
T2. B) Para adicionar a cada registo as features ano e mês utilizei um nodo “String to Date&Time” aplicado ao atributo “Date” para o transformar num formato que seja reconhecido pelo nodo seguinte “Extract Date&Time Fields” que adiciona as colunas indicadas na configuração, Year e Month (number):



T2. C) Depois utilizei o nodo “GroupBy”, onde agrupei os registos por loja, tipo, tamanho, ano e mês, agregando assim de forma a obter o somatório das vendas semanais de cada loja “(Sum (Weekly_Sales))” e a indicação da existência de feriados nesse mês(Max(IsHoliday)).



T2. D) Na realização desta alínea, utilizei o nodo “Normalizer” de modo a normalizar o somatório das vendas semanais neste nodo utilizei a configuração da transformação linear Min-Max entre 0 e 1.



T2. E) e F) Aqui comecei por utilizar o nodo “Auto-Binner” para criar 4 bins de igual frequência sobre o valor normalizado da feature do somatório das vendas semanais (deixando ligada a opção “replace target column(s)”).

Dialog - 4:12 - Auto-Binner

File

Auto Binner Settings Number Format Settings Flow Variables Job Manager Selection Memory Policy

☒ Manual Selection ☐ Wildcard/Regex Selection

Exclude

Filter

- Store
- Size
- Year
- Month (number)
- Max(IsHoliday)

☒ Enforce exclusion

Include

Filter

- Sum(Weekly_Sales)

☐ Enforce inclusion

Binning Method

☒ Fixed number of bins

Number of bins: 4

Equal: frequency

☐ Sample quantiles

Quantiles (comma separated): 0.0, 0.25, 0.5, 0.75, 1.0

Bin Naming

☒ Numbered e.g.: Bin 1, Bin 2, Bin 3

☐ Borders e.g.: [-10,0], (0,10], (10,20]

☐ Midpoints e.g.: -5, 5, 15

☐ Force integer bounds

☒ Replace target column(s)

OK Apply Cancel ?

Depois criei a tabela seguinte com o objetivo de renomear cada Bin, ou seja, o primeiro corresponde a Low, o segundo a Medium, o terceiro a High e o quarto a Very High. Para efetivamente substituir esses bins pelos novos, utilizei o nodo “Value lookup”:

| | | |
|------|-------|-----------|
| Row0 | Bin 1 | Low |
| Row1 | Bin 2 | Medium |
| Row2 | Bin 3 | High |
| Row3 | Bin 4 | Very High |

Matching

Lookup column (data table)

Sum(Weekly_Sales)

Key column (dictionary table)

Sum(Weekly_Sales)s

If multiple rows match

☒ Use first ☐ Use last

If no row matches

☒ Insert missing values ☐ Match next smaller ☐ Match next larger

Output

Append columns (from dictionary table)

☒ Manual ☐ Wildcard ☐ Regex ☐ Type

Search

As

Excludes

Sum(Weekly_Sales)s

Any unknown columns

Includes

Sum(Weekly_Sales)

Cancel OK

T3. A) e B) e C) Nesta tarefa usei um nodo “Decision Tree Learner” que recebe os dados já processados do dataset para treinar o modelo de modo a estimar as vendas mensais de cada uma das suas lojas.

Dialog - 4:14 - Decision Tree Learner

File

Options

PMMLSettings

Flow Variables

Job Manager Selection

General

Class column

S

Sum(Weekly_Sales)

Quality measure

Gini index

Pruning method

No pruning

☒
 Reduced Error Pruning

Min number records per node

2

Number records to store for view

10 000

☒
 Average split point

Number threads

4

☒
 Skip nominal columns without domain information

Root split

☐
 Force root split column

Root split column

I

Max(IsHoliday)

Binary nominal splits

☐
 Binary nominal splits

Max #nominal

10

☐
 Filter invalid attribute values in child nodes

Depois, apliquei um nodo “Decision Tree Predictor” que recebe os dados de teste de um nodo “CSV Reader” e também recebe o modelo de árvore de decisão treinado e prevê o valor de vendas de cada mês. Depois usei o nodo “Scorer” para poder visualizar a accuracy e a matriz de confusão:

Confusion matrix - 5:55:03 - Scorer

File

Hilite

| Sum(Week... | Very High | High | Low | Medium |
|-------------|-----------|------|-----|--------|
| Very High | 14 | 6 | 0 | 0 |
| High | 9 | 11 | 0 | 4 |
| Low | 0 | 0 | 12 | 8 |
| Medium | 0 | 1 | 6 | 14 |

Correct classified: 51

Accuracy: 60%

Cohen's kappa (κ): 0.467%

Wrong classified: 34

Error: 40%

T4. A) Para a realização desta tarefa utilizei o nodo “Parameter Optimization Loop Start” que cria um ciclo de testes com uma flow variável “New parameter” e depois usei o nodo “Parameter Optimization Loop End” que fecha o ciclo, para fazer o tuning do modelo, experimentando todos os valores, entre 2 e 10, para o número mínimo de registo por nodo, assim usei as seguintes configurações, respetivamente

Dialog - 4:18 - Parameter Optimization Loop Start

File

Standard settings

Flow Variables

Job Manager Selection

Parameters

| Parameter | Start value | Stop value | Step size | Integer? |
|---------------|-------------|------------|-----------|-------------------------------------|
| New parameter | 2.0 | 10 | 1.0 | <input checked="" type="checkbox"/> |

+ Add new parameter

Strategy settings

Search strategy

Brute Force

4:19 - Parameter Optimization Loop End

Flow Variables

Job Manager Selection

Memory Policy

Flow variable with objective function value

Accuracy

Function should be...

☒ maximized

☐ minimized

E obtive que o melhor número mínimo de registo foi 6 com uma accuracy de 65.882%

► 1: Best parameters

► 2: All parameters

Flow Variables

Rows: 9

Columns: 2

Table

Statistics

| # | Row... | New parameter Number (integer) | Objective value Number (double) |
|---|--------|-----------------------------------|------------------------------------|
| 1 | Row0 | 2 | 0.6 |
| 2 | Row1 | 3 | 0.647 |
| 3 | Row2 | 4 | 0.647 |
| 4 | Row3 | 5 | 0.671 |
| 5 | Row4 | 6 | 0.682 |
| 6 | Row5 | 7 | 0.682 |
| 7 | Row6 | 8 | 0.671 |
| 8 | Row7 | 9 | 0.647 |
| 9 | Row8 | 10 | 0.659 |

| Sum(Week... | Very High | High | Low | Medium |
|-------------|-----------|------|-----|--------|
| Very High | 14 | 6 | 0 | 0 |
| High | 6 | 14 | 0 | 4 |
| Low | 0 | 0 | 10 | 10 |
| Medium | 0 | 0 | 3 | 18 |

Correct classified: 56

Wrong classified: 29

Accuracy: 65,882%

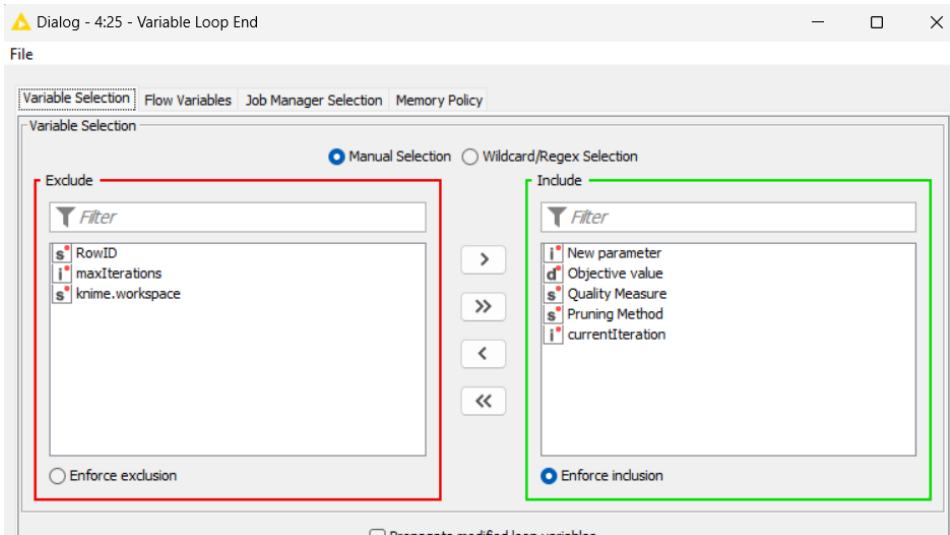
Error: 34,118%

Cohen's kappa (κ): 0.645%

T4. B) e C) e D) Nesta tarefa comecei por usar um outro loop para testar as possibilidades de medida de qualidade e do método de pruning em simultâneo. Para isso criei um nodo “Table Creator”, com os registos seguintes, e liguei a um nodo “Table Row To Variable Loop Start”.

| | S Pruning ... | S Quality ... |
|------|---------------|---------------|
| Row0 | No pruning | Gini index |
| Row1 | No pruning | Gain radio |
| Row2 | MDL | Gini index |
| Row3 | MDL | Gain radio |

Para fechar estes ciclos, primeiramente utilizei o nodo “Table Row To Variable”, com os inputs “Objective Value” e “New parameter”, depois usei o nodo “Variable Loop End”, com as seguintes configurações:



Com isso obtive os seguintes resultados:

| # | Row... | New parameter <small>Number (integer)</small> | Objective value <small>Number (double)</small> | Quality Measure <small>String</small> | Pruning Method <small>String</small> | currentiteration <small>Number (integer)</small> |
|---|--------|--|---|--|---|---|
| 1 | Row0 | 6 | 0.682 | Gini index | No pruning | 0 |
| 2 | Row1 | 2 | 0.694 | Gini index | MDL | 1 |
| 3 | Row2 | 3 | 0.682 | Gain ratio | No pruning | 2 |
| 4 | Row3 | 2 | 0.706 | Gain ratio | MDL | 3 |

Onde podemos ver que a combinação que oferece melhor performance é utilizando 2 registros por nodo com a medida de qualidade Gain Ratio e o método de pruning MDL. Apesar de haver um aumento de performance não é um aumento muito significativo e não existe muita discrepância. Mesmo usando a melhor combinação de fatores a nossa accuracy continua a ser apenas 68%.

| Sum(Week... | Very High | High | Low | Medium |
|-------------|-----------|------|-----|--------|
| Very High | 16 | 4 | 0 | 0 |
| High | 9 | 11 | 0 | 4 |
| Low | 0 | 0 | 15 | 5 |
| Medium | 0 | 0 | 5 | 16 |

Correct classified: 58

Wrong classified: 27

Accuracy: 68,235%

Error: 31,765%

Cohen's kappa (κ): 0,578%

Para guardar a tabela de performance da melhor combinação dos hiper parâmetros usei o nodo “Csv Writer”.

T5. Em primeiro lugar, utilizei o mesmo procedimento que para o modelo da Decision Tree mas desta vez para uma Random Forest. Utilizei o nodo “Random Forest Learner” ao qual fiz chegar os dados preparados anteriormente para treino, e liguei-o ao nodo “Random Forest Predictor” que recebe também os dados para teste. Para medir a precisão utilizei um nodo “Scorer” que avaliou este modelo com uma accuracy superior à accuracy do modelo inicial do Decision Tree.

Confusion Matrix - 4:33:17 - Scorer

| Sum(Weekl... | Very High | High | Low | Medium |
|--------------|-----------|------|-----|--------|
| Very High | 14 | 6 | 0 | 0 |
| High | 7 | 13 | 0 | 4 |
| Low | 0 | 0 | 18 | 2 |
| Medium | 0 | 0 | 7 | 14 |

Correct classified: 59 Wrong classified: 26
 Accuracy: 69,412% Error: 30,588%

Depois, usei um loop para experimentar valores para o número máximo de níveis. Para isso apliquei o nodo “Parameter Optimization Loop Start” que passa uma variável “maxlevels” que varia nos inteiros entre 2 e 10 para o Forest Tree Learner e fechei o ciclo com o nodo “Parameter Optimization Loop End”. Obtive o seguinte resultado:

Rows: 9 | Columns: 2

| # | Row... | maxLevels Number (integer) | Objective value Number (double) |
|---|--------|-------------------------------|------------------------------------|
| 1 | Row0 | 2 | 0.529 |
| 2 | Row1 | 3 | 0.6 |
| 3 | Row2 | 4 | 0.682 |
| 4 | Row3 | 5 | 0.694 |
| 5 | Row4 | 6 | 0.694 |
| 6 | Row5 | 7 | 0.694 |
| 7 | Row6 | 8 | 0.694 |
| 8 | Row7 | 9 | 0.694 |
| 9 | Row8 | 10 | 0.694 |

Depois utilizei outro loop para testar as possibilidades de split criterion. Para isso associei um nodo “Table Creator”, com os registos seguintes, a um nodo “Table Row To Variable Loop Start”.

| | \$ splitcriteria |
|------|------------------|
| Row0 | Gini |
| Row1 | InformationGain |
| Row2 | InformationG... |

Para terminar os ciclos, novamente utilizei o nodo “Table Row To Variable”, com os inputs “Objective Value” e “maxLevels”, depois usei o nodo “Variable Loop End”, obtendo a seguinte tabela:

Rows: 3 | Columns: 4

TableStatistics

| # | Row... | maxLevels <small>Number (integer)</small> | Objective value <small>Number (double)</small> | splitcriteria <small>String</small> | currentiteration <small>Number (integer)</small> |
|---|--------|--|---|--|---|
| 1 | Row0 | 5 | 0.694 | Gini | 0 |
| 2 | Row1 | 5 | 0.694 | InformationGain | 1 |
| 3 | Row2 | 5 | 0.694 | InformationGainRatio | 2 |

Por fim para testar se diferentes números de modelos fariam o Forest Tree Learner prever com mais precisão adicionei duas variáveis no nodo “Parameter Optimization Loop Start”:

Standard SettingsFlow VariablesJob Manager Selection

Parameters

| Parameter | Start value | Stop value | Step size | Integer? | |
|-----------|-------------|------------|-----------|-------------------------------------|--|
| maxLevels | 2 | 10 | 1.0 | <input checked="" type="checkbox"/> | |
| nrminimo | 2 | 10 | 0.1 | <input checked="" type="checkbox"/> | |
| forests | 100 | 200 | 2.0 | <input checked="" type="checkbox"/> | |

E configurei o nodo “Forest Tree Learner”:

OptionsFlow VariablesJob Manager SelectionMemory Policy

| | | |
|--------------------------------|---------------|--|
| targetColumn | | |
| seed | | |
| maxLevels | maxLevels | |
| minNodeSize | nrminimo | |
| minChildSize | | |
| dataFraction | | |
| isDataSelectionWithReplacement | | |
| hardCodedRootColumn | | |
| columnSamplingMode | | |
| columnFractionPerTree | | |
| columnAbsolutePerTree | | |
| isUseDifferen...tesAtEachNode | | |
| nrModels | forests | |
| splitCriterion | splitcriteria | |
| missingValueHandling | | |
| useAverageSplitPoints | | |
| useBinaryNominalSplits | | |
| fingerprintColumn | | |
| columnFilterConfig | | |
| ignoreColumnsWithoutDomain | | |
| nrHilitePatterns | | |
| saveTargetDistributionInNodes | | |
| rowSamplingMode | | |

Obtive por fim os seguintes resultados:

| # | Row... | maxLevels <small>Number (integer)</small> | nrminimo <small>Number (integer)</small> | forests <small>Number (integer)</small> | Objective value <small>Number (double)</small> | splitcriteria <small>String</small> | currentiteration <small>Number (integer)</small> |
|---|--------|--|---|--|---|--|---|
| 1 | Row0 | 10 | 2 | 104 | 0.718 | Gini | 0 |
| 2 | Row1 | 10 | 4 | 100 | 0.706 | InformationGain | 1 |
| 3 | Row2 | 10 | 2 | 102 | 0.706 | InformationGainRatio | 2 |

Fazendo a tabela de confusão da melhor combinação dos hiperparametros obtenho :

| Sum(Week... | Very High | High | Low | Medium |
|-------------|-----------|------|-----|--------|
| Very High | 14 | 6 | 0 | 0 |
| High | 7 | 13 | 0 | 4 |
| Low | 0 | 0 | 18 | 2 |
| Medium | 0 | 0 | 6 | 15 |

Correct classified: 60

Wrong classified: 25

Accuracy: 70,588%

Error: 29,412%

Cohen's kappa (κ): 0,609%

Ou seja, apesar do tuning do modelo ele não conseguiu melhorar assim muito a sua accuracy. O que considero ter haver com o dataset sobre o qual estamos a treinar o modelo. Com mais data acredito que o modelo tivesse maior previsão de accuracy.

T6. Olhando agora para as performances dos modelos treinados em T4 e em T5, posso concluir que o modelo de tunning de uma Random Forest foi me mais lento a fazer tuning do que o modelo de tunning de uma Decision Tree no entanto o modelo que apresenta melhor Accuracy é o modelo da Random Forest uma vez que tem Accuracy de 70,5% enquanto o modelo da Tuned da Decision Tree tem de Accuracy de 68%. Acredito que com mais dados a diferença pudesse ser maior visto que fazer o tuning do modelo da Random Forest não alterou o seu nível de previsão significativamente.