

Relatório de Data Mining

Trabalho de conclusão da disciplina

Professora Manoela Kohler

Kickstarter Projects



Mayta S. Custódio

Matrícula: 192.671.147

Sumário

Análise exploratória	3
Atributos Desnecessários:	4
Missing Values:	5
Separação de base em treino e teste	8
Normalização de Dados	8
Modelagem	9
k-NN	10
Random Forest	11
Decision Tree	12
Naive Bayes	13
Conclusão	14
Considerações finais	14

Análise exploratória

Na primeira análise realizada verificamos que o database possui 369.669 linhas e 17 colunas, e contém informações sobre projetos cadastrados no site kickstarter.com de 2009 a 2018. Link: <https://www.kaggle.com/kemical/kickstarter-projects>

Demonstra sua classificação na coluna 'state', informando se foi bem-sucedido ou não, nomeando-os como *successful*, *failed*, *live*, *canceled*, *suspended*, *undefined*;

Como possuo a licença de estudante concedida pelo Rapidminer, trabalharemos com a base completa sem limitação de *rows*.

O primeiro passo no Rapidminer foi abrir o operador 'Read csv' e importar a base de dados, colocando a primeira linha como 'header row', mantendo o separador default da vírgula; e trocando o papel da segunda coluna de ID com o mesmo nome('id'), e da coluna 'state' que será considerada o 'label' de tipo *polynominal*.

As datas contidas em deadline e launched foram reconhecidas como 'date' e o formato em dd/MM/yyyy pelo Rapidminer em 77% dos registros desses atributos.

As três primeiras linhas estão em conflito com a ordem das demais, então no *wizard* de importação de data, selecionei em 'start row - 5', pois assim, essas linhas serão desconsideradas.

Marquei também 'Replace erros with missings values', porque notei que diversas colunas possuem valores e *strings* com caracteres que podem dar erro no carregamento.

import data - specify your data format

Specify your data format

☒ Header Row

1

Start Row

5

Column Separator

Semicolon ;

File Encoding

windows-1252

Escape Character

\

Decimal Character

.

☒ Use Quotes

"

☐ Trim Lines

☒ Skip Comments

#

1	ID	name	category	main_ca...	currency	deadline	goal	launched	pledged	state	backers	couri
2	85964225	Debut Al...	Grace is ...	Indie Rock	Music	USD	17/04/20...	700.00	02/04/20...	850.00	successf...	32
3	9485259...	Oceanus -	A game I...	Video Ga...	Games	USD	23/02/20...	40000.00	24/01/20...	20.00	failed	8
4	9573385...	â€¢	Help sav...	Food	Food	USD	08/01/20...	1875.00	09/11/20...	545.00	failed	13
5	1339173...	Spirits of...	Tabletop...	Games	EUR	26/01/20...	20000.00	02/01/20...	3694.00	live	82	ES
6	1830173...	Digital D...	Art	Art	USD	01/02/20...	650.00	02/01/20...	7.00	live	1	US
7	2106246...	Help sav...	Food	Food	USD	16/01/20...	10000.00	02/01/20...	165.00	live	3	US
8	9747383...	EVO Pla...	Product...	Design	USD	09/02/20...	15000.00	02/01/20...	269.00	live	8	US
9	1486845...	America...	Hip-Hop	Music	USD	16/01/20...	500.00	02/01/20...	0.00	live	0	US

Row ...	ID	state	att1	name	category	main_categ...	currency	deadline	goal	launched	pledged	backers	country	usd pledg...	usd_pledge...	usd_goal_re...	att17
1	1830173355	live	?	Digit...	Art	Art	USD	Feb 1, 2018	650	Jan 2, 2018	7	1	US	7	7	650	?
2	2106246194	live	?	Help ...	Food	Food	USD	Jan 16, 2018	10000	Jan 2, 2018	165	3	US	165	165	10000	?
3	974738310	live	?	EVO ...	Product...	Design	USD	Feb 9, 2018	15000	Jan 2, 2018	269	8	US	269	269	15000	?
4	1486845240	live	?	Amer...	Hip-Hop	Music	USD	Jan 16, 2018	500	Jan 2, 2018	0	0	US	0	0	500	?
5	150815475	live	?	Carl ...	Graphic ...	Design	GBP	Feb 1, 2018	250	Jan 2, 2018	40	2	GB	54.010	54.560	340.990	?
6	473911584	live	?	A Ne...	Hip-Hop	Music	USD	Feb 1, 2018	7500	Jan 2, 2018	128	3	US	128	128	7500	?
7	80324970	live	?	T.P.C...	Woodwo...	Crafts	GBP	Feb 1, 2018	5000	Jan 2, 2018	0	0	GB	0	0	6819.890	?
8	272668251	live	?	Forg...	Crafts	Crafts	USD	Jan 20, 2018	10000	Jan 2, 2018	286	1	US	286	286	10000	?
9	163752222	live	?	Print...	Tabletop...	Games	EUR	Feb 1, 2018	300	Jan 2, 2018	1240	31	DE	192.140	1504.980	364.110	?
10	462436087	live	?	Charl...	Children'	Publishing	GBP	Feb 11, 2018	700	Jan 2, 2018	0	0	GB	0	0	954.780	?
11	754894401	live	?	Sprin...	Horror	Film & Video	USD	Feb 1, 2018	22000	Jan 2, 2018	6	2	US	0	6	22000	?
12	35294140	live	?	Fanta...	Tabletop...	Games	GBP	Feb 2, 2018	500	Jan 2, 2018	180	9	GB	135.020	245.520	681.990	?
13	712099998	live	?	The F...	Rock	Music	GBP	Feb 11, 2018	1000	Jan 2, 2018	0	0	GB	0	0	1363.980	?
14	1398085599	live	?	Feud...	Playing ...	Games	AUD	Feb 1, 2018	5500	Jan 2, 2018	0	0	AU	0	0	4330.030	?
15	1978733762	live	?	Healt...	Design	Design	JPY	Feb 1, 2018	34000	Jan 2, 2018	8500	1	JP	75.430	76.490	305.950	?
16	1036415983	live	7165	Aikya...	Music	Music	USD	Mar 3, 2018	10000	Jan 2, 2018	174	3	US	174	174	10000	?
17	2007608555	live	?	Trin...	Painting	Art	USD	Jan 17, 2018	1000	Jan 2, 2018	0	0	US	0	0	1000	?

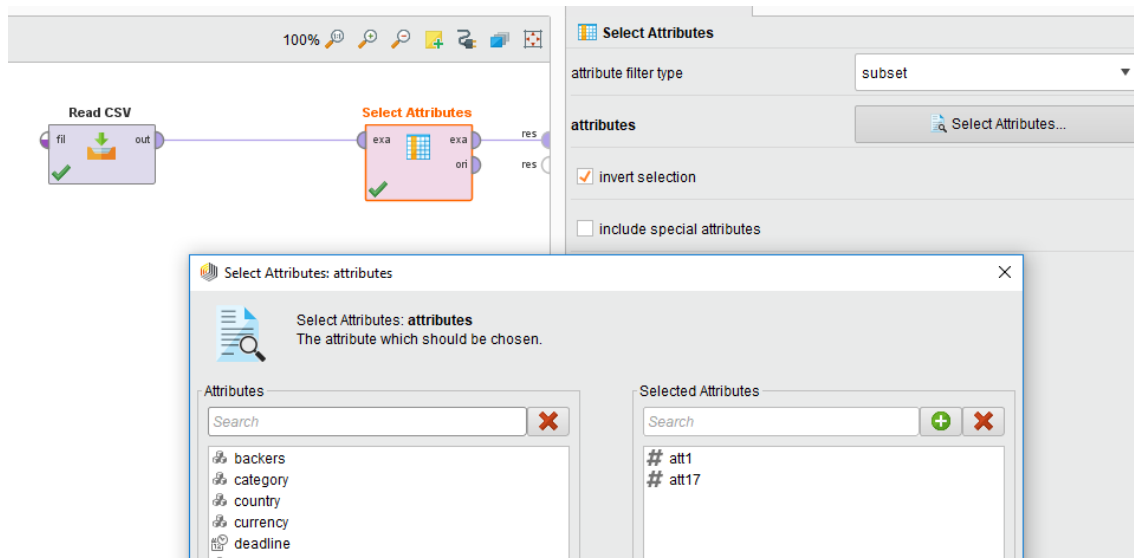
ExampleSet (369,665 examples, 2 special attributes, 15 regular attributes)

Atributos Desnecessários:

Temos uma grande quantidade missings em duas colunas, sendo necessária a retirada do database, pois atrapalha o modelo. É o caso dos atributos *'att1'* e *'att17'*, com 350.610 e 369.666 valores faltantes respectivamente.

Name	Type	Missing
att17	Real	369666
att1	Integer	350610

Foi aplicado o operador *'Select Attributes'*. No campo de *'filter type'*, selecionamos *'subset'* e no *'Select attributes'*, colocamos as colunas *'att1'* e *'att17'*, marcando por fim, *'invert selection'*, pois dessa forma irá selecionar esses atributos, excluindo-os.



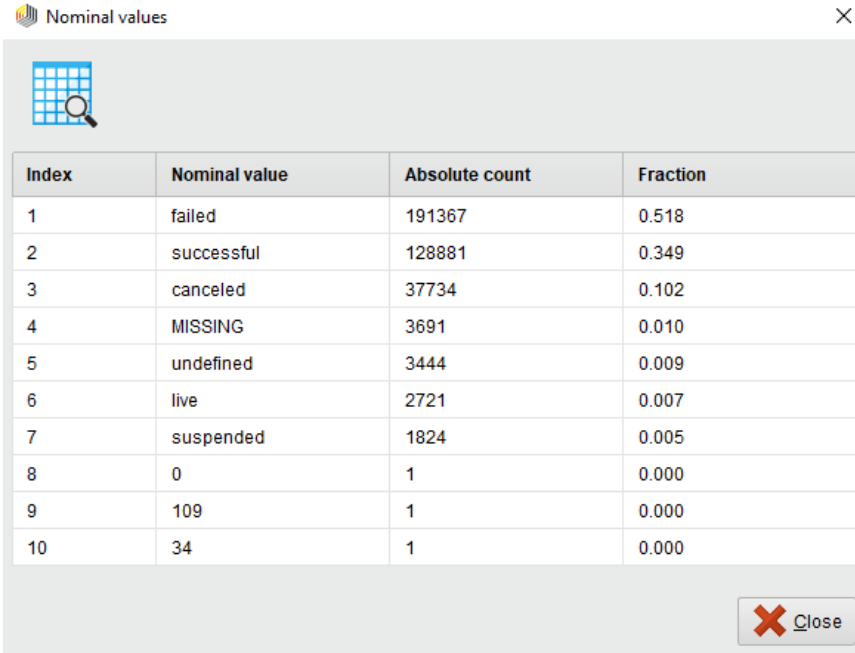
Como a quantidade de atributos é relativamente pequena, a consulta de sua importância é de fácil identificação, sendo assim, não se faz necessária a remoção de mais nenhuma coluna, pois todas contribuem com informações relevantes à composição do sucesso ou falha de cada projeto.

Missing Values:

Apliquei o operador *'Replace All Missings'* marcando *'include special attribute'*, pois também existem valores faltantes em *'ID'* e *'state'*.

Name	Type	Missing
ID	Integer	0
state	Polynomial	0
name	Polynomial	0
category	Polynomial	0
main_category	Polynomial	0
currency	Polynomial	0
deadline	Date	0
goal	Polynomial	0

Identifiquei três linhas que estão com números ao invés de classificações, assim como os missings que serão levados em consideração na modelagem.

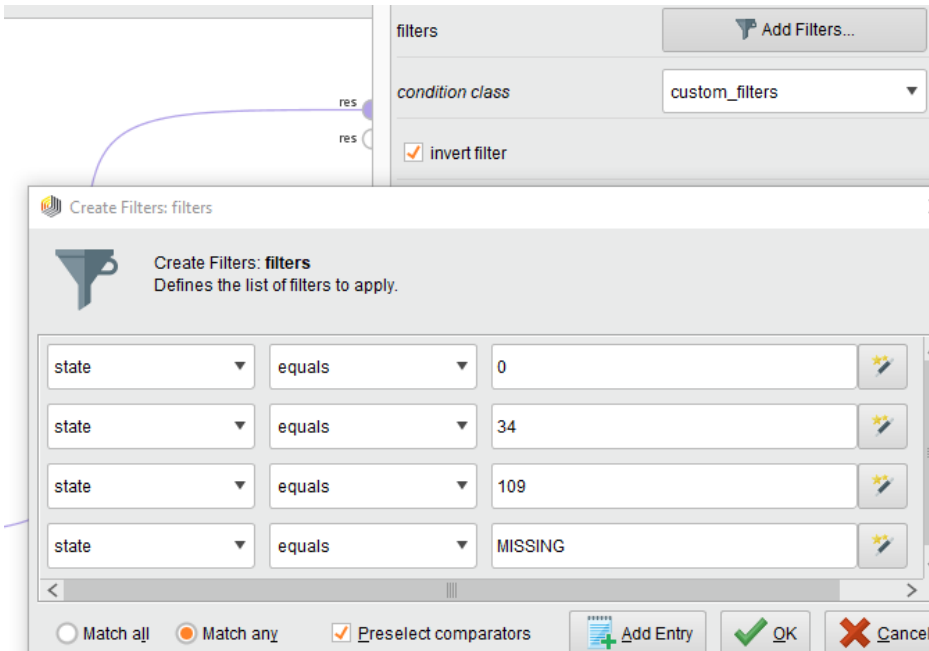


The image shows a 'Nominal values' dialog box with a table of nominal values. The table has four columns: Index, Nominal value, Absolute count, and Fraction. The data is as follows:

Index	Nominal value	Absolute count	Fraction
1	failed	191367	0.518
2	successful	128881	0.349
3	canceled	37734	0.102
4	MISSING	3691	0.010
5	undefined	3444	0.009
6	live	2721	0.007
7	suspended	1824	0.005
8	0	1	0.000
9	109	1	0.000
10	34	1	0.000

At the bottom right of the dialog is a 'Close' button with a red X icon.

Como essa é uma coluna de grande importância, apliquei o operador 'Filter Examples', criando os filtros 'State – equals –0 ; 34 ; 109; MISSING', para os rows discrepantes. Marquei 'Match any' e 'invert filter', ficando então com os dados que o operador identificou como relevantes de acordo com as condições apresentadas no filtro.



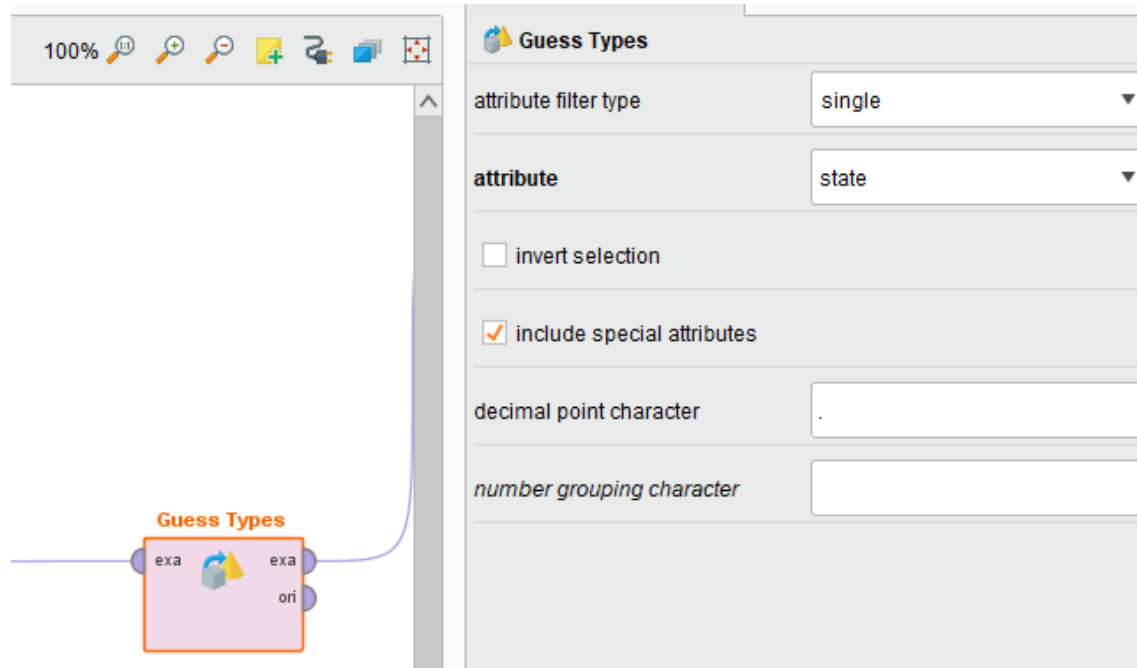
The image shows a filter configuration interface. At the top, there's a 'filters' section with an 'Add Filters...' button and a 'condition class' dropdown set to 'custom_filters'. Below this, there's a checkbox for 'invert filter' which is checked. A purple arrow points from the 'filters' section to a 'Create Filters: filters' dialog box.

The 'Create Filters: filters' dialog box has a title bar and a subtitle 'Create Filters: filters Defines the list of filters to apply.' It contains a list of filter entries, each with a dropdown for the variable name, a dropdown for the comparison operator, and a text input for the value. The entries are:

- state equals 0
- state equals 34
- state equals 109
- state equals MISSING

At the bottom of the dialog, there are radio buttons for 'Match all' and 'Match any' (selected), a checkbox for 'Preselect comparators' (checked), and buttons for 'Add Entry', 'OK', and 'Cancel'.

Em seguida, apliquei o operador 'Guess type'. Em 'attribute filter type', selecionei 'single', e em 'attribute', nosso label. Marquei 'include special attribute', porque a coluna 'state' se enquadra.



The screenshot shows the 'Guess Types' operator configuration. The 'attribute filter type' is set to 'single', and the 'attribute' is set to 'state'. The 'include special attributes' checkbox is checked. The 'decimal point character' is set to '.' and the 'number grouping character' is empty.

Finalizamos o tratamento de *missings values* e erros no *label*.

Nominal values

Index	Nominal value	Absolute count	Fraction
1	failed	191367	0.523
2	successful	128881	0.352
3	canceled	37734	0.103
4	undefined	3444	0.009
5	live	2721	0.007
6	suspended	1824	0.005

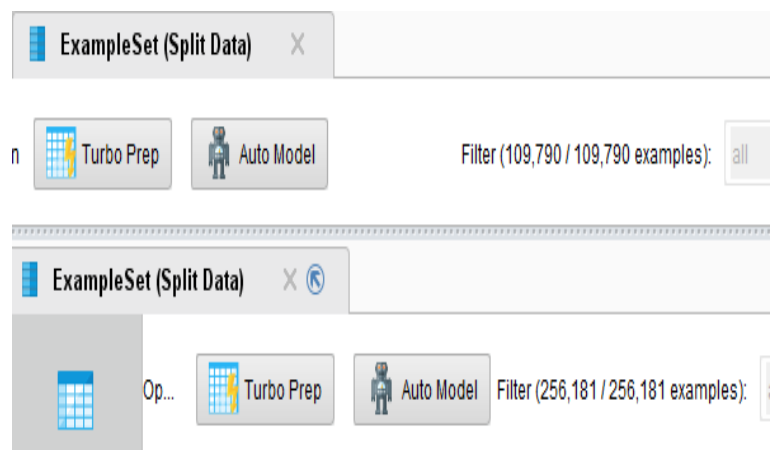
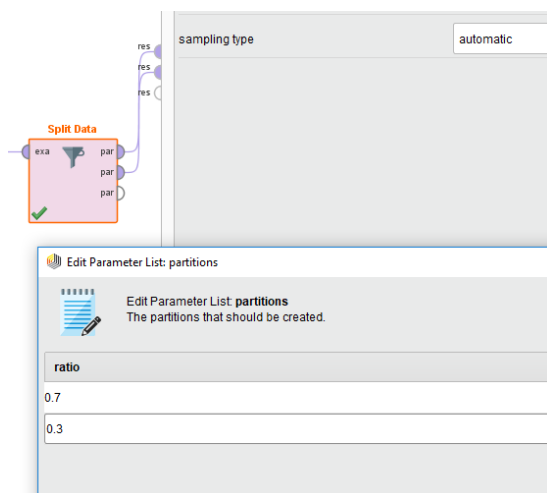
Close

Separação de base em treino e teste

Apliquei o operador 'split data'. Em 'partitions', separei a base em treino (0.7%) e teste (0.3%). Tipo de *sampling* no automático ou *stratified* por default, pois é um problema de classificação com base já rotulada. Dessa forma irá manter a proporção das classes nas bases separadas.

Base treino: 256.181 examples

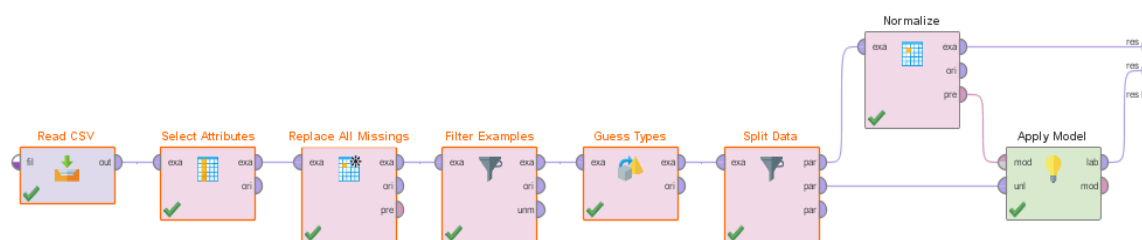
Base teste: 109.790 examples



Normalização de Dados

Foi aplicado o operador 'Normalize' para a normalização da base de treino tendo em vista que os valores dos dados tem grandezas discrepantes. Método: Z-transformation.

Em seguida, o operador 'apply model' foi inserido para aplicar a normalização na base de teste. Ligando a saída do 'Normalize – pre' ao 'mod' do operador e a saída da base de teste ao 'unl' do 'apply model'.

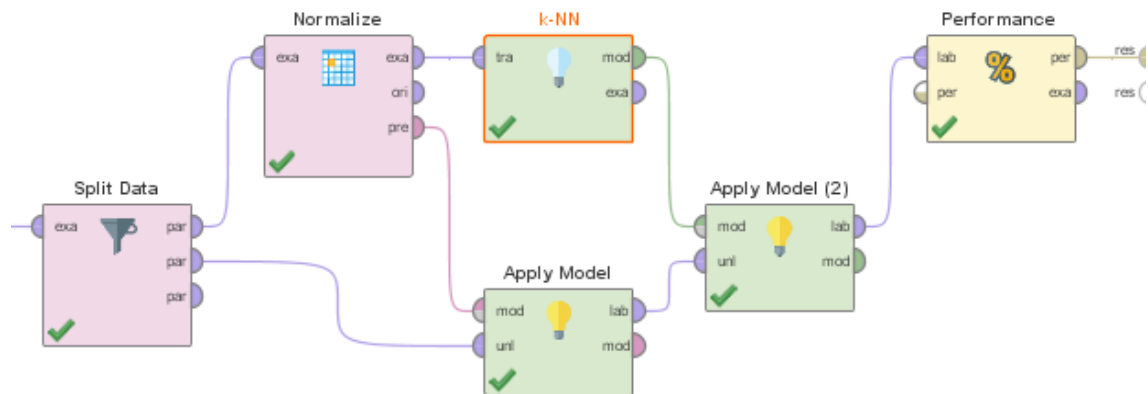


Modelagem

Modelos de classificação utilizados: k-NN, Random Forest, Decision Tree e Naive Bayes, com suas acurácias e kappa.

Após a separação das bases, normalização do treino, aplicação dessa normalização na base teste, foram incluídos os operadores dos modelos na base de treino e com o *'apply model'* ligado na porta de *'mod'* de ambos operadores. A base de teste foi ligada à segunda porta do *apply model 'unl' (unlabeled data)*, a partir do *'apply model'* anterior da normalização, na saída *'lab'*. Por fim, foi acrescentado, o operador de *'Performance (classification)'* para avaliar a o desempenho.

k-NN

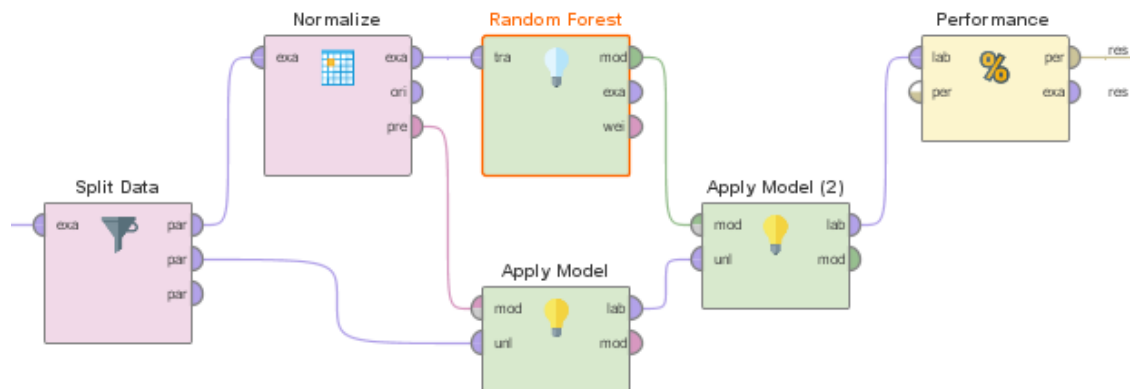


accuracy: 56.97%

	true live	true canceled	true success...	true failed	true suspen...	true undefined	class precisi...
pred. live	805	67	4	8	5	0	90.55%
pred. canceled	1	527	1223	2267	30	25	12.94%
pred. succes...	3	2626	18865	12959	121	230	54.20%
pred. failed	7	8089	18551	42144	389	572	60.42%
pred. suspen...	0	2	2	8	1	0	7.69%
pred. undefin...	0	9	19	24	1	206	79.54%
class recall	98.65%	4.66%	48.79%	73.41%	0.18%	19.94%	

k-NN	acc	kappa
5	56,97%	0.221
30	57,48%	0.188

Random Forest

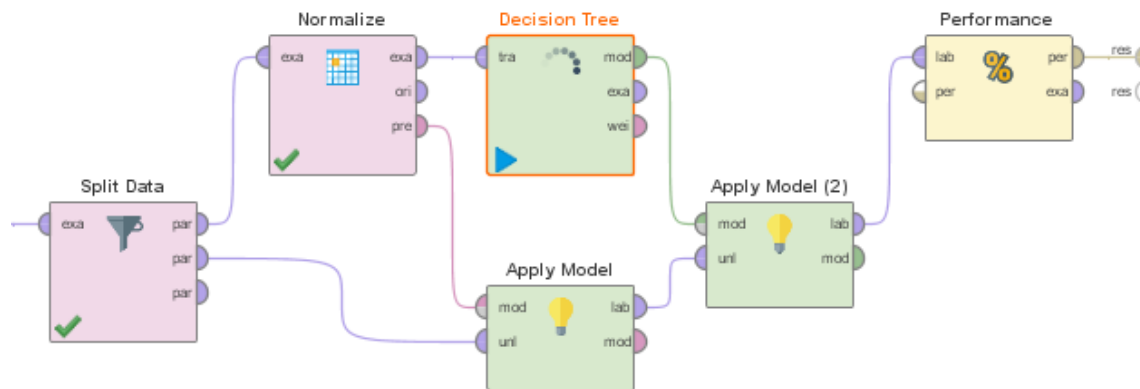


accuracy: 79.18%

	true live	true canceled	true success...	true failed	true suspen...	true undefined	class precisi...
pred. live	807	68	1	7	5	0	90.88%
pred. canceled	0	5	0	0	0	0	100.00%
pred. succes...	5	2387	37782	10099	153	0	74.93%
pred. failed	4	8860	879	47304	389	0	82.36%
pred. suspen...	0	0	0	0	0	0	0.00%
pred. undefin...	0	0	2	0	0	1033	99.81%
class recall	98.90%	0.04%	97.72%	82.40%	0.00%	100.00%	

RF	acc	kappa
100	79,18%	0.631
300	79,09%	0.630

Decision Tree

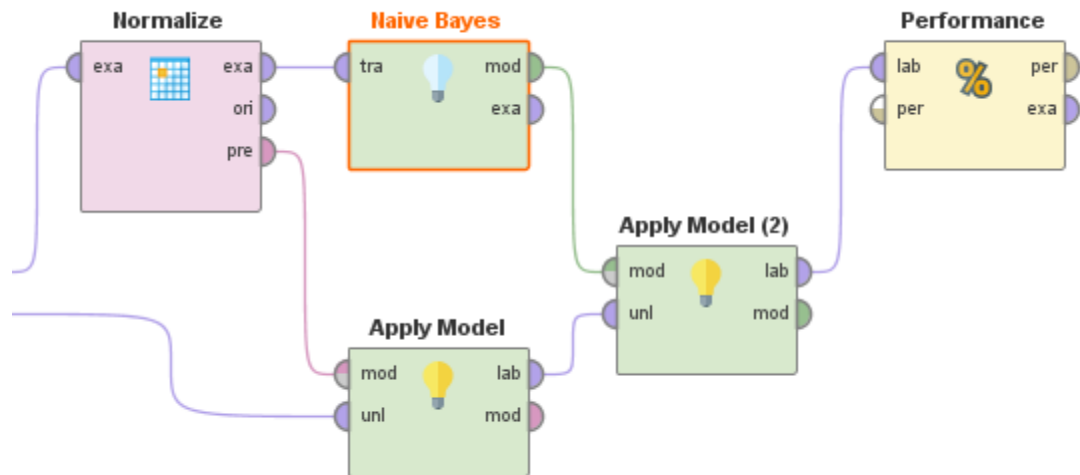


accuracy: 86.33%

	true live	true canceled	true successful	true failed	true suspended	true undefined	class precision
pred. live	807	69	4	9	5	0	90.27%
pred. canceled	5	40	53	133	5	0	16.95%
pred. successful	2	565	38567	2928	119	1	91.43%
pred. failed	2	10646	37	54338	418	0	83.03%
pred. suspended	0	0	2	2	0	0	0.00%
pred. undefined	0	0	1	0	0	1032	99.90%
class recall	98.90%	0.35%	99.75%	94.65%	0.00%	99.90%	

DT	acc	kappa
10	76,83%	0.592
20	85,96%	0.746
40	86,33%	0.753

Naive Bayes



accuracy: 61.92%

	true live	true canceled	true successful	true failed	true suspended	true undefined	class precision
pred. live	786	316	842	1067	16	0	25.97%
pred. canceled	5	537	2396	1174	23	1	12.98%
pred. successful	10	204	11297	587	51	1	92.98%
pred. failed	11	10158	23803	54331	449	6	61.21%
pred. suspended	4	97	294	219	7	0	1.13%
pred. undefined	0	8	32	32	1	1025	93.35%
class recall	96.32%	4.74%	29.22%	94.64%	1.28%	99.23%	

NB	acc	61,92%
	kappa	0.287

Conclusão

O modelo que apresentou melhor performance foi o Decision Tree, com acurácia de 86,33%. Kappa de 0.753, com *maximal depth* de 40, mesmo tendo baixo rendimento nos 'canceled' e 'suspended'.

O modelo que apresentou pior performance foi o k-NN = 5, tendo $\text{acc} = 56,97\%$ e $\text{kappa} = 0.221$.

Considerações finais

É muito interessante ser capaz de executar toda a linha de desenvolvimento do trabalho e observar um resultado muito próximo ao que se faria em sala. Procurei a minha base de dados no site kaggle.com/ e encontrei diversos temas interessantes. Apesar de gostar da sugestão da base 'horses', queria trabalhar com uma do 'zero' e explorar a database sozinha.

Comecei realizando os primeiros passos no 'R', porém a base necessitava de inúmeros tratamentos que ainda não domino completamente na linguem, então, passei para o Rapidminer afim de conseguir visualizar melhor por quais etapas precisava passar.

Na etapa do pré-processamento, testei incluir o operador '*Nominal to Numerical*' para a transformação dos atributos nominais em dados numéricos e ser capaz de aplicar o modelo SVM. Porém, o programa não rodou devido ao grande volume de dados e falta de memória. Considerei realizar um *downsampling*, porém teria o risco de deletar dados importantes da base. O modelo aprenderia padrões em cima desses dados que não representariam tão bem as classes. Sendo assim, optei por não usar o SVM.

Agradeço muito a Professora Manoela Kohler pela paciência e explicações durante a diciplina. Irei continuar praticando em 'R' para saber desenvolver em vários estilos.