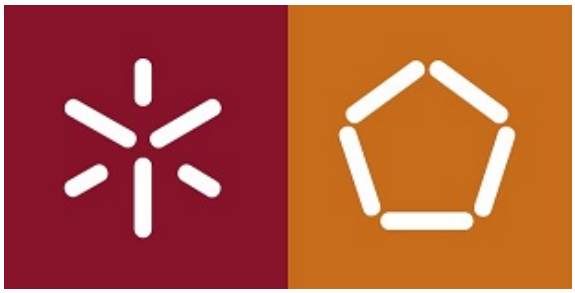

Universidade do Minho
Departamento de Informática

Conceção de Modelos de Aprendizagem

Aprendizagem e Decisões Inteligentes

Grupo 24

6 de maio, 2022


Beatriz Rodrigues
(a93230)


Francisco Neves
(a93202)


Gabriela Prata
(a93288)


Guilherme Fernandes
(a93216)

Índice

1 Introdução **3**

2 Previsão de Preços de Voos **4**
2.1 Características do *dataset* 4
2.2 Análise do *dataset* 5
2.2.1 *Airline* 5
2.2.2 *Flight* 5
2.2.3 *Source City* e *Destination City* 6
2.2.4 *Departure Time* e *Arrival Time* 6
2.2.5 *Stops* 7
2.2.6 *Class* 7
2.2.7 *Duration* 8
2.2.8 *Days Left* 8
2.2.9 *Price* 9
2.3 Pré-Processamento dos Dados 10
2.4 Modelação 11
2.5 Análise de Resultados 13

3 Classificação de Salários **14**
3.1 Características do *dataset* 14
3.2 Análise do *dataset* 15
3.2.1 *Salary Classification* 15
3.2.2 *Age* 15
3.2.3 *Education* 16
3.2.4 *Workclass* 17
3.2.5 *Occupation* 17
3.2.6 *Hours per Week* 18
3.2.7 *Relationship* e *Marital Status* 18
3.2.8 *Race* 19
3.2.9 *Gender* 19
3.3 Pré-Processamento dos Dados 20
3.4 Modelação 21
3.4.1 *Downsampling* 22
3.4.2 *Upsampling* 23
3.4.3 *Binning* 24
3.5 Análise de Resultados 26

4 Conclusões **27**

5 Referências **28**

1. Introdução

Neste trabalho prático foram considerados dois *datasets*: o primeiro, selecionado pelo grupo, contém informação acerca de vários voos entre seis das maiores cidades na Índia; o segundo, atribuído de acordo com o número do grupo, contém informação acerca de classificação de salários.

Utilizando os modelos de aprendizagem aprendidos, construíram-se modelos de *Machine Learning* com o objetivo de, respetivamente, prever o preço de um voo de acordo com as suas características (o que constitui um Problema de Regressão) e prever se um indivíduo terá um salário `<=50K` ou `>50k` (o que, por sua vez, constitui um Problema de Classificação).

2. Previsão de Preços de Voos

2.1 Características do *dataset*

Este *dataset* apresenta 300152 linhas e 11 colunas (as *features*). Estas últimas são as seguintes:

1. ***Airline***: Nome da companhia responsável pelos voos;
2. ***Flight***: Código identificador do avião;
3. ***Source City***: Cidade de partida;
4. ***Departure Time***: Altura do dia da partida;
5. ***Stops***: Número de paragens entre a cidade de partida e a cidade de chegada;
6. ***Arrival Time***: Altura do dia de chegada;
7. ***Destination City***: Cidade destino do avião;
8. ***Class***: Classe do lugar ocupado (pode ser *Business* ou *Economy*);
9. ***Duration***: Quantidade de tempo, em horas, necessária para viajar entre as cidades;
10. ***Days Left***: Dias que faltam desde o momento atual até ao momento da viagem;
11. ***Price***: Informação acerca do preço do bilhete de avião.

As *features* podem ser classificadas da seguinte forma:

Categóricas → *Airline*, *Flight*, *Source City*, *Departure Time*, *Stops*, *Arrival Time*, *Destination City*, *Class*

Contínuas → *Duration*, *Days Left*, *Price*

2.2 Análise do *dataset*

2.2.1 *Airline*

O *dataset* apresenta a seguinte distribuição relativamente às *airlines* abrangidas.

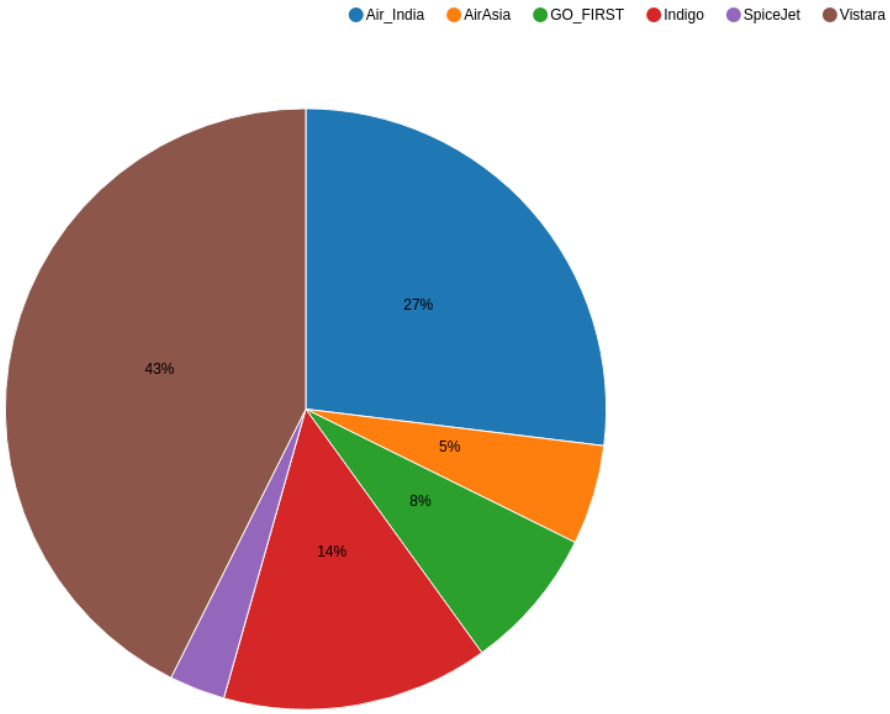

Figura 2.1: *airline pie chart*

É possível reparar que as *airlines* que ocorrem com mais frequência são a `Vistara` e `Air_India`.

2.2.2 *Flight*

Relativamente aos *flights*, é possível observar pela seguinte figura que existem demasiados valores únicos e que, por isso, é evidente que esta *feature* não iria contribuir para a aprendizagem do nosso modelo.

Column	Exclude Column	No. missings	Unique values	All nominal values	Frequency Bar Chart
flight	☐	0	>1000	UK-706, UK-852, UK-858, UK-808, UK-810, [...]!, SG-8193, G8-705, G8-107, SG-8913, 6E-5003	Not all nominal values calculated.

Figura 2.2: estatísticas dos *flights*

2.2.3 *Source City* e *Destination City*

Relativamente à proporção entre cidades de origem e entre cidades de destino, podemos comprovar que esta é bastante equilibrada.

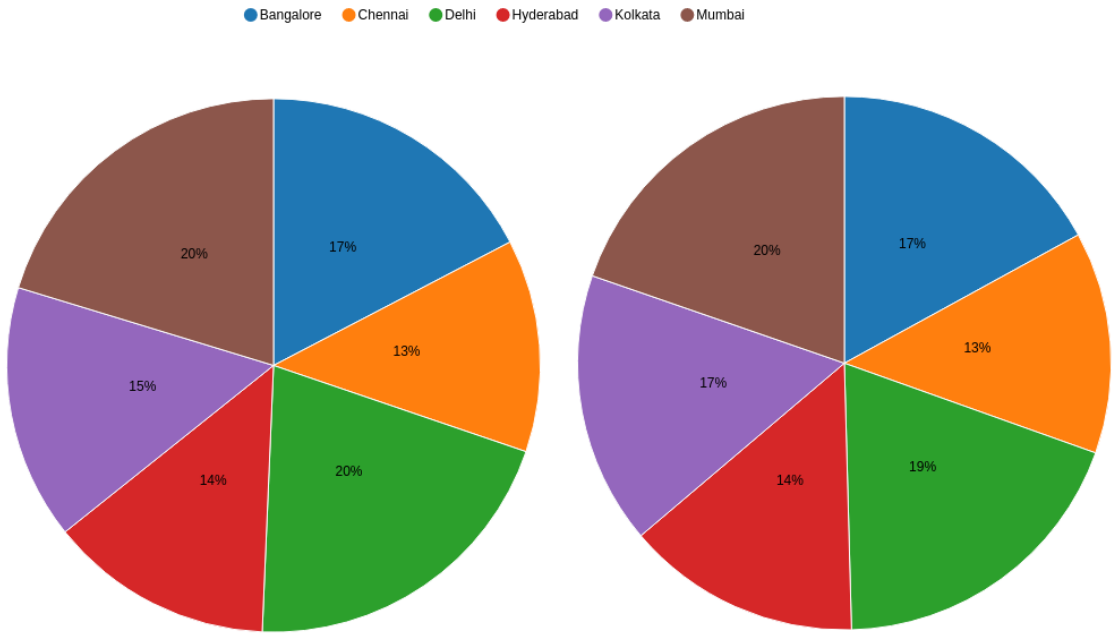

Figura 2.3: *source cities* e *destination cities pie chart* respetivamente

2.2.4 *Departure Time* e *Arrival Time*

A partir dos gráficos seguintes, podemos notar que existem bastante menos voos a partir da `late night` do que das restantes alturas do dia. No entanto, podemos também notar que, em relação com as restantes alturas do dia, há menos voos a chegar durante a `early morning`, `afternoon` e `late night`.

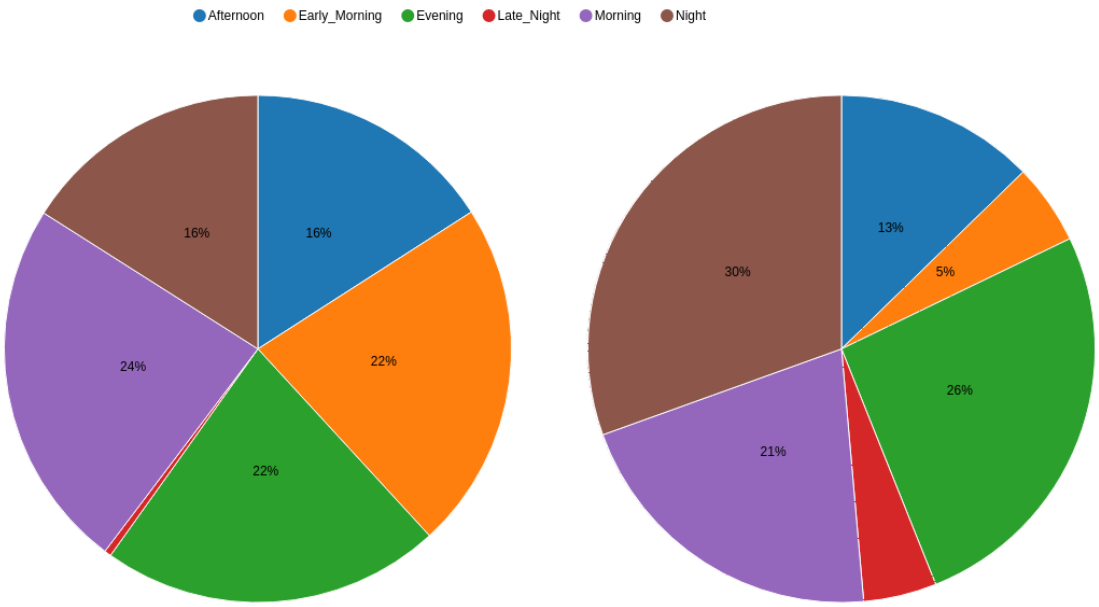

Figura 2.4: *departure time* e *arrival time pie chart* respetivamente

2.2.5 *Stops*

Esta *feature* especifica que a grande maioria dos voos apenas realiza uma paragem.

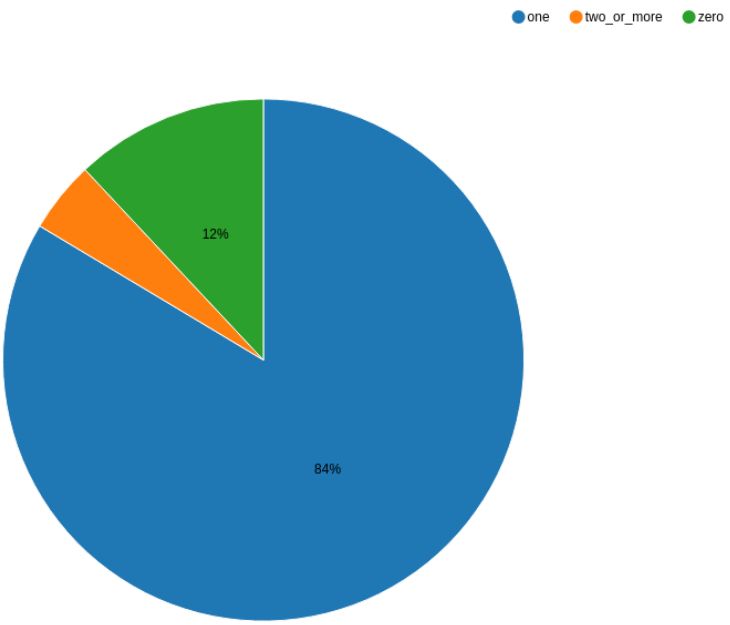

Figura 2.5: *stops pie chart*

2.2.6 *Class*

A partir do gráfico seguinte, compreende-se que existem mais dados relativos a viagens realizadas em classe de economia.

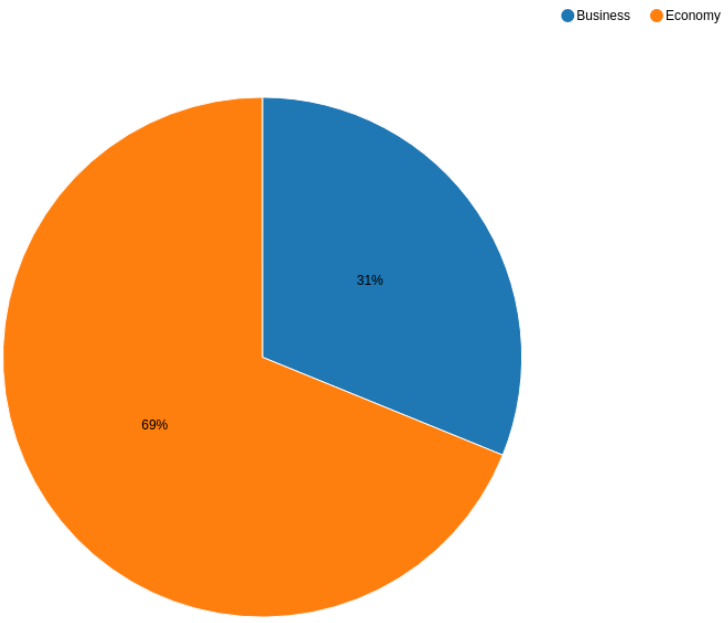

Figura 2.6: *class pie chart*

2.2.7 *Duration*

Ao analisarmos as durações das viagens, notamos que a maioria das viagens acaba por ter uma duração compreendida entre 0 e 15 horas, com um pico de frequência no intervalo entre as 6 e 11 horas de viagem.

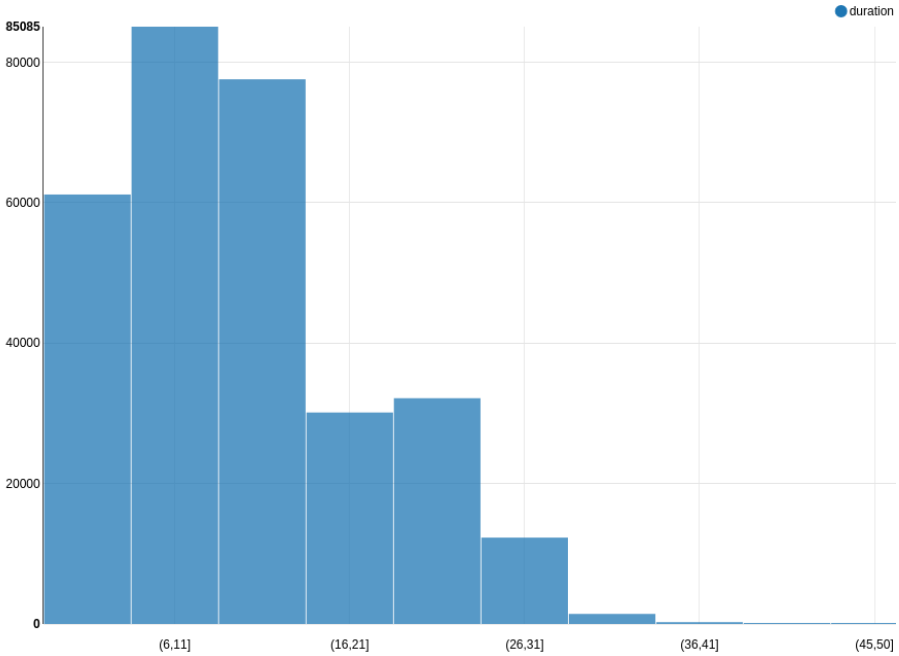

Figura 2.7: Histograma da *duration*

2.2.8 *Days Left*

Relativamente aos dias em falta para a viagem, podemos notar uma distribuição relativamente uniforme.

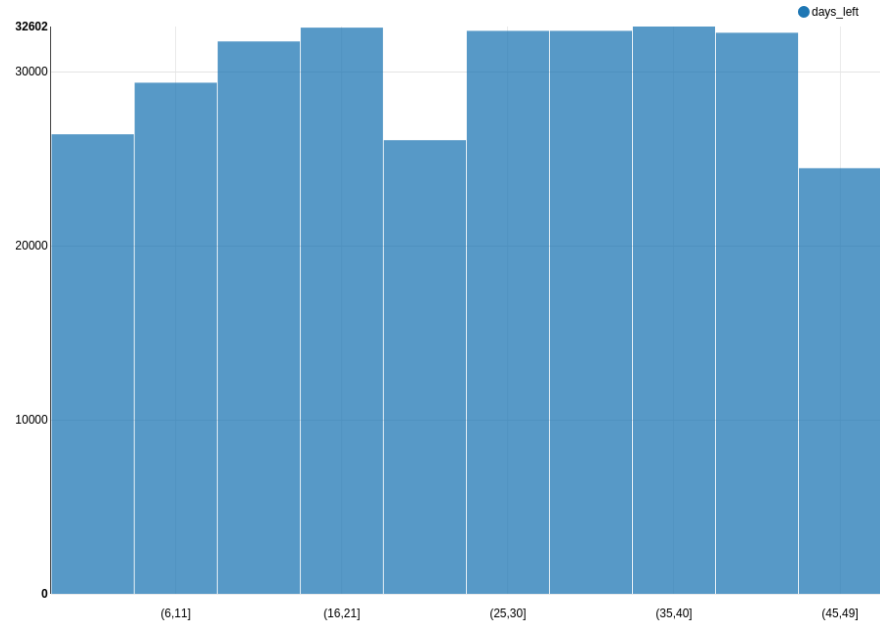

Figura 2.8: Histograma de *days left*

2.2.9 *Price*

A distribuição dos dados acerca dos preços das viagens é evidenciado pelo seguinte histograma.

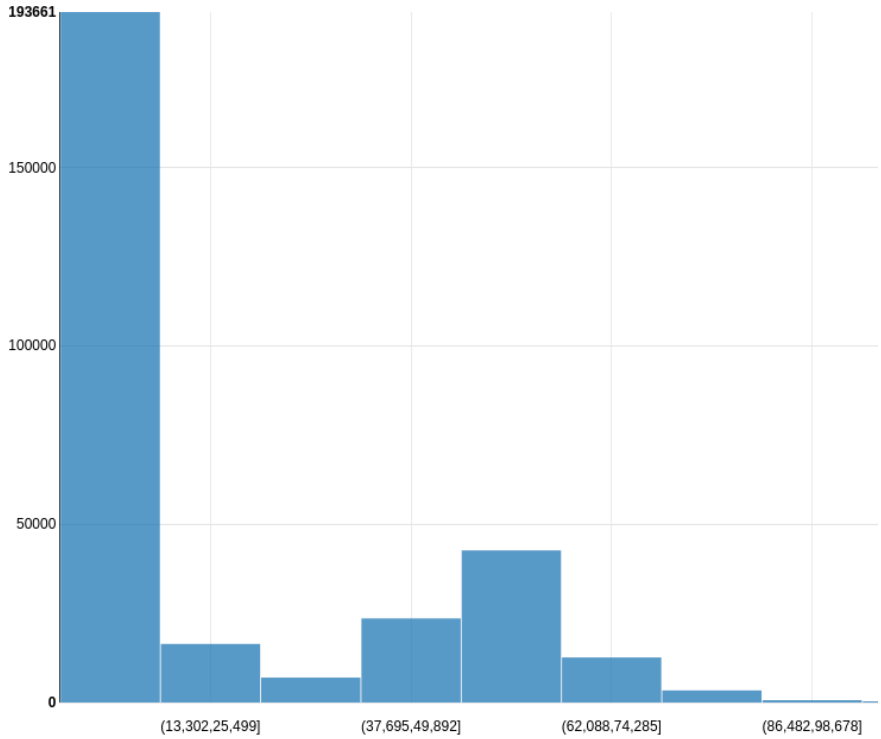

Figura 2.9: Histograma de *price*

É evidente que a maioria dos voos apresenta um preço contido no intervalo entre 0 e 13301.

2.3 Pré-Processamento dos Dados

O pré-processamento dos dados acabou por revelar-se bastante simples. Primeiramente, filtramos as colunas irrelevantes, como uma coluna *column* que não se trata de uma feature mas apenas a numeração de linhas e a featureflights visto que o facto de existirem tantos valores únicos não iria contribuir para a aprendizagem do modelo. Seguidamente, removeram-se *outliers* das variáveis numéricas.

Devido às características favoráveis das *features*, da quantidade de dados disponíveis e da ausência de *missing values*, acabamos por verificar que outros tratamentos de dados acabavam por não melhorar o modelo.

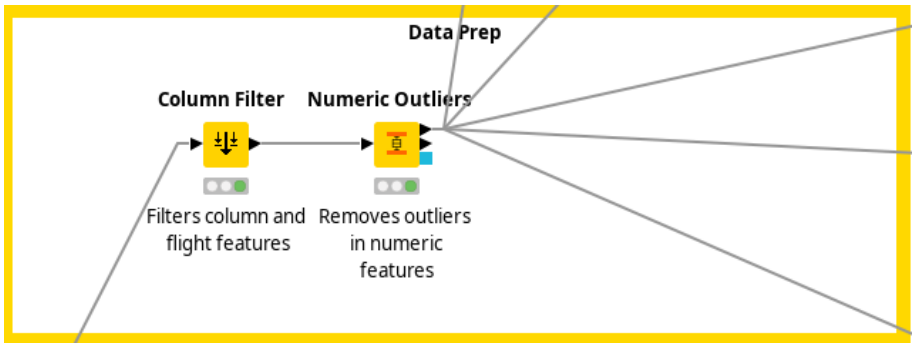

Figura 2.10: Pré-Processamento dos Dados

Além disso, para determinar a que combinação de *features* estão associados os melhores resultados, foram realizados dois processos de *feature selection*, inseridos em metanodos. Um deles utiliza o algoritmo *Single Regression Tree* e o outro *Linear Regression*.

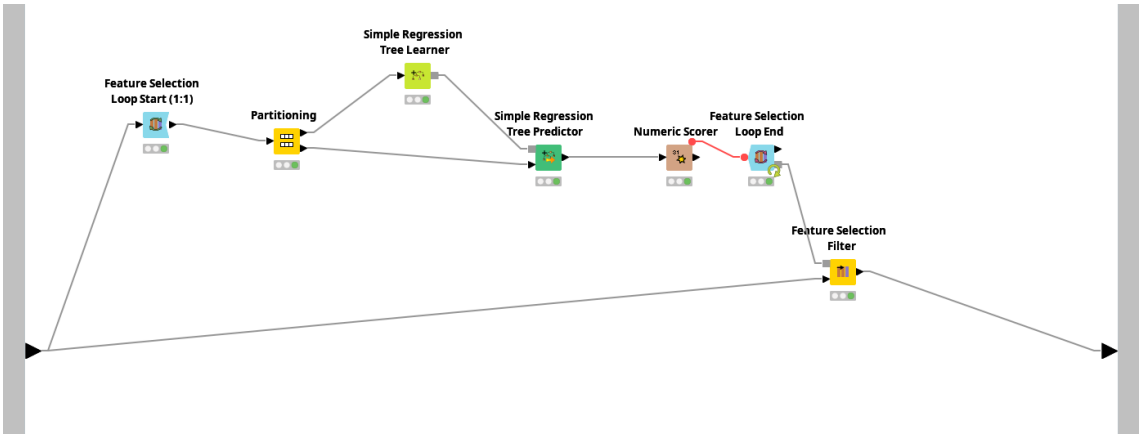

Figura 2.11: *Feature Selection*

Ambos os resultados obtidos indicaram que a utilização de todas as *features* seria a melhor opção.

2.4 Modelação

Como se trata de um problema de regressão, decidimos utilizar os algoritmos de aprendizagem *Linear Regression* e *Simple Regression Tree*.

Para além disso, utilizamos nodos de *cross-validation* designados por *X-Partitioner* e *X-Aggregator*, por permitirem a utilização de uma maior quantidade de dados. O *X-Partitioner* foi configurado para utilizar 10 validações e realizar *statified sampling*, de forma a manter as proporções da classe *price*.

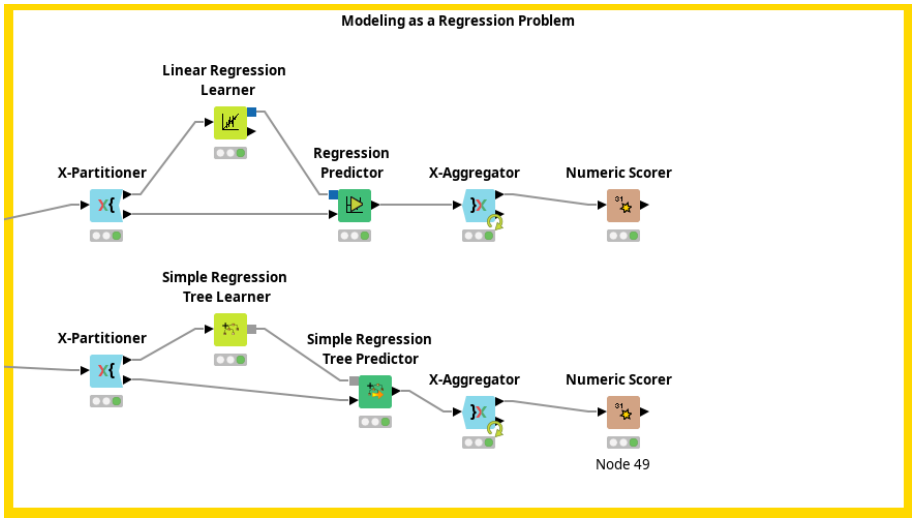

Figura 2.12: Teste com algoritmos de regressão

Para o algoritmo *Linear Regression* foi obtido um $r^2 = 0.912$, enquanto para o algoritmo *Simple Regression Tree* foi obtido um $r^2 = 0.977$.

Statistics - 0:38 - Numer

File	
R²:	0.912
Mean absolute error:	4,565.251
Mean squared error:	45,418,249.411
Root mean squared error:	6,739.306
Mean signed difference:	-0.044
Mean absolute percentage error:	0.464
Adjusted R²:	0.912

Statistics - 0:49 - Numer

File	
R²:	0.977
Mean absolute error:	1,152.569
Mean squared error:	11,781,714.452
Root mean squared error:	3,432.45
Mean signed difference:	44.326
Mean absolute percentage error:	0.074
Adjusted R²:	0.977

Figura 2.13: Resultados do *Linear Regression* e *Simple Regression Tree*, respetivamente

Para explorar novas alternativas, decidimos avaliar o desempenho deste problema se fosse convertido para um problema de classificação. Para isso, utilizamos um nodo *Numeric Binner*, que permitiu a divisão dos preços nas classes ilustradas na imagem seguinte.

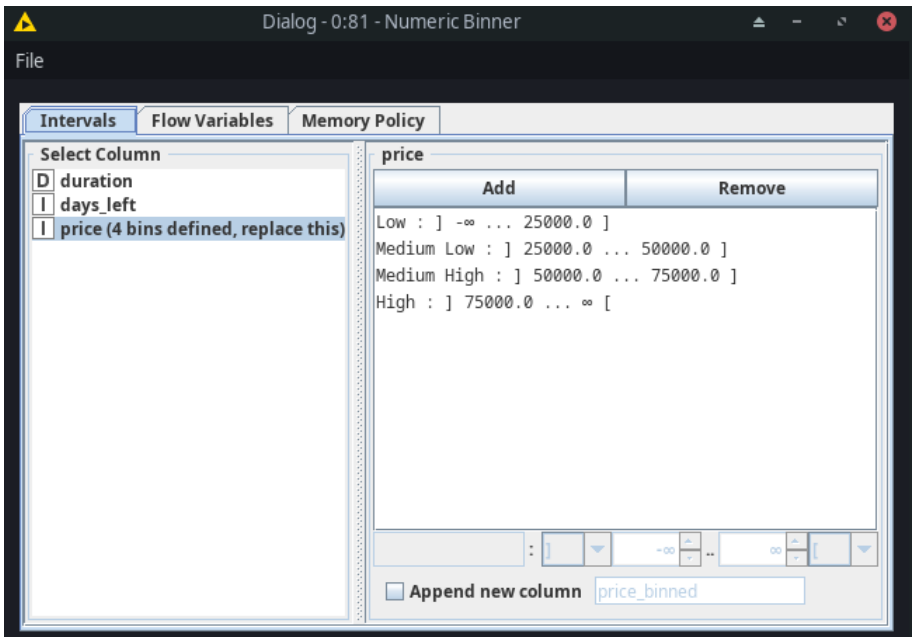

Figura 2.14: Resultados do *Linear Regression* e *Simple Regression Tree*, respetivamente

Assim, é possível agora utilizar novos algoritmos. Decidimos experimentar com os algoritmos *Naive Bayes*, *Random Forest Learner* e *Gradient Boosted Trees*. Novamente, estes foram complementados com a utilização de *X-Partitioner* e *X-Aggregator*. Os resultados obtidos dos algoritmos mencionados foram, respetivamente: *accuracy* de 90.41% e *Cohen's kappa* de 0.794; *accuracy* de 96.67% e *Cohen's kappa* de 0.928; *accuracy* de 94.60% e *Cohen's kappa* de 0.883.

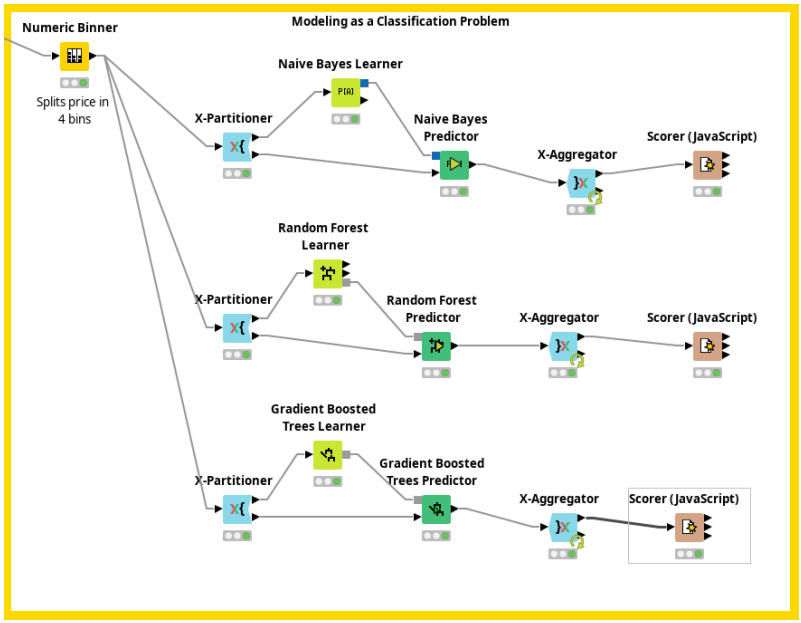

Figura 2.15: Teste com algoritmos adequados a problemas de classificação

2.5 Análise de Resultados

Desta forma, consideramos que este *dataset* continha uma quantidade de dados bastante proveitosa para efeitos de aprendizagem e que apenas necessitou de um tratamento simples para obter bons resultados. Se tratarmos o problema tendo em conta que se trata de um problema de regressão, o algoritmo que revelou melhores resultados foi o *Simple Regression Tree*.

Porém, quando este problema é convertido para um problema de classificação, verificamos que o algoritmo *Random Forest Learner* também revela uma accuracy favorável de 96.67% e Cohen's kappa de 0.928, compreensível tendo em conta as características do *dataset* selecionado.

3. Classificação de Salários

3.1 Características do *dataset*

Este *dataset* apresenta 48843 linhas e 15 *features*. Estas últimas são as seguintes:

1. ***Age***: Idade do indivíduo;
2. ***Workclass***: Classe do trabalho do indivíduo;
3. ***Fnlwgt***: Número de identificação do indivíduo;
4. ***Education***: Nível de educação do indivíduo;
5. ***Education-num***: Nível de educação do indivíduo de forma numérica;
6. ***Marital-status***: Estado matrimonial do indivíduo;
7. ***Occupation***: Tipo geral de ocupação do indivíduo;
8. ***Relationship***: Estado de relacionamento do indíviduo;
9. ***Race***: Raça do indivíduo;
10. ***Sex***: Sexo biológico do indivíduo;
11. ***Capital-gain***: Capital ganho pelo indivíduo;
12. ***Capital-loss***: Capital perdido pelo indivíduo;
13. ***Hours-per-week***: Número de horas de trabalho semanal do indivíduo;
14. ***Native-country***: País de origem do indivíduo;
15. ***Salary-classification***: Classificação do salário ($> 50k$ ou $<= 50k$).

As *features* podem ser classificadas da seguinte forma:

Categóricas → *Workclass*, *Education*, *Marital-status*, *Occupation*, *Relationship*, *Race*, *Sex*, Native-country, *Salary-classification*

Contínuas → *Age*, *Fnlwgt*, *Education-num*, *Capital-gain*, *Capital-loss*, *Hours-per-week*

3.2 Análise do *dataset*

3.2.1 *Salary Classification*

O *dataset* apresenta uma distribuição de 76% entradas classificadas com $<= 50K$ e 24% entradas classificadas com $> 50K$

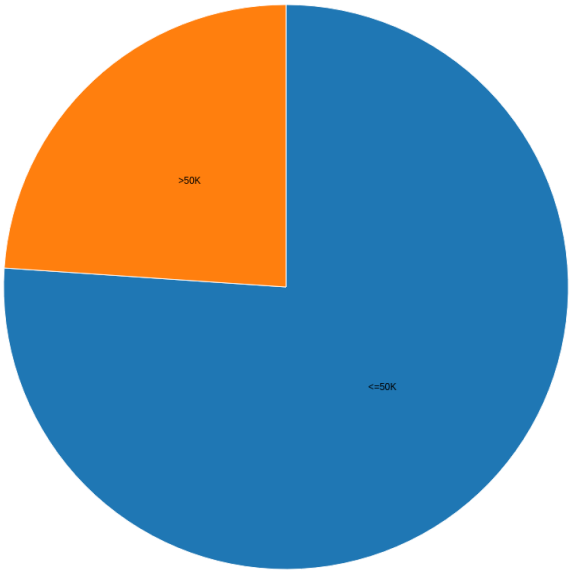

Figura 3.1: *salary-classification pie chart*

3.2.2 *Age*

As idades variam entre 17 e 90 anos, analisando a distribuição das mesmas, reparamos que existe uma densidade maior no intervalo de 19 a 47 anos.

Além disso, existe maior número de salários mais elevados no intervalo de 27 a 62 anos.

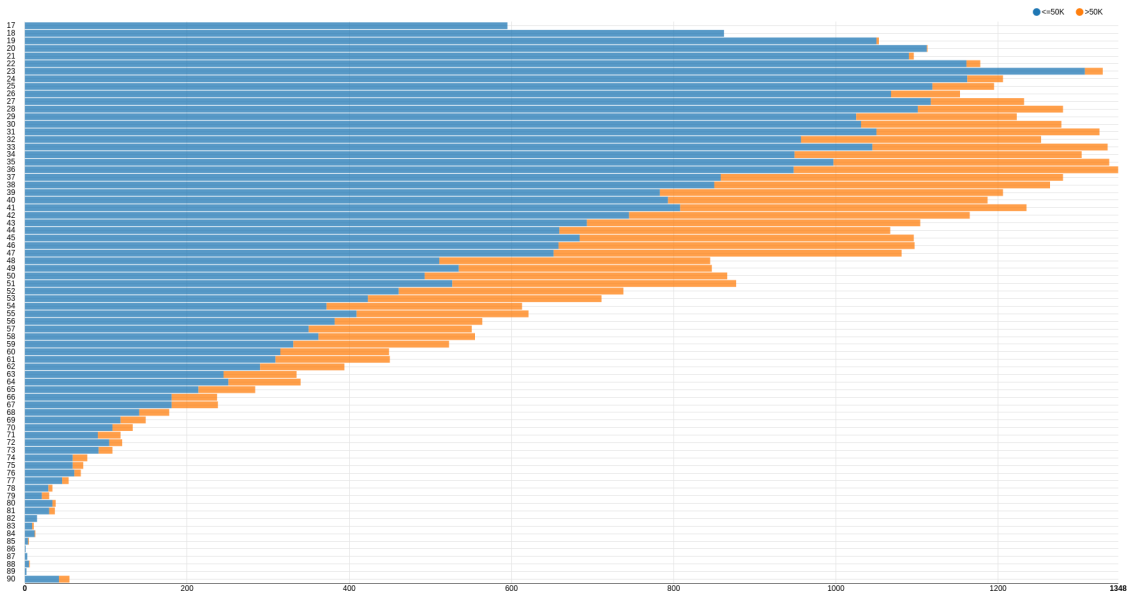

Figura 3.2: Gráfico de barras das idades

3.2.3 *Education*

Analisando os valores das colunas *education* e *education-num* observamos que estas são equivalentes.

education	education-num
Preschool	1
1st-4th	2
5th-6th	3
7th-8th	4
9th	5
10th	6
11th	7
12th	8
HS-grad	9
Some-college	10
Assoc-voc	11
Assoc-acdm	12
Bachelors	13
Masters	14
Prof-school	15
Doctorate	16

Figura 3.3: *education* e *education-num*

Também podemos ver que existem muitos mais indivíduos com os níveis de educação *HS-grad*, *Some-college* e *Bachelors* e que, geralmente, quanto maior o nível de eduação, maior é a percentagem de salários mais elevados.

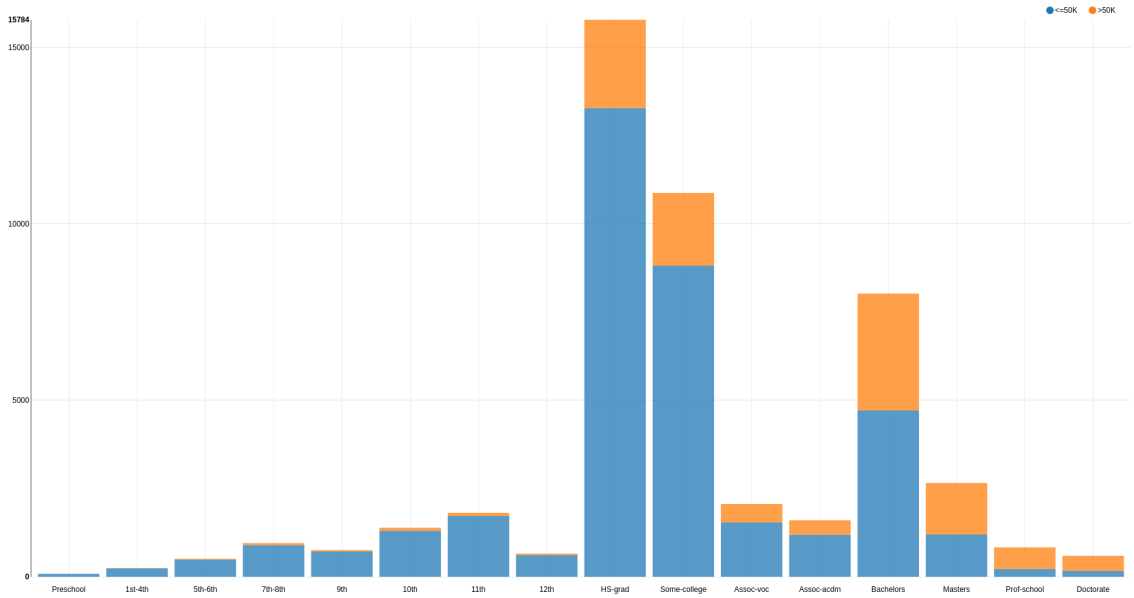

Figura 3.4: Gráfico de barras de educação

3.2.4 *Workclass*

Como podemos ver, a maioria dos indivíduos trabalha no setor privado e, em termos de salário, não existe muita diferença significativa, menos na classe de *Self-emp-inc* em que existe uma maior percentagem de salários elevados.

Podemos observar também que existem dados em falta que terão de ser tratados.

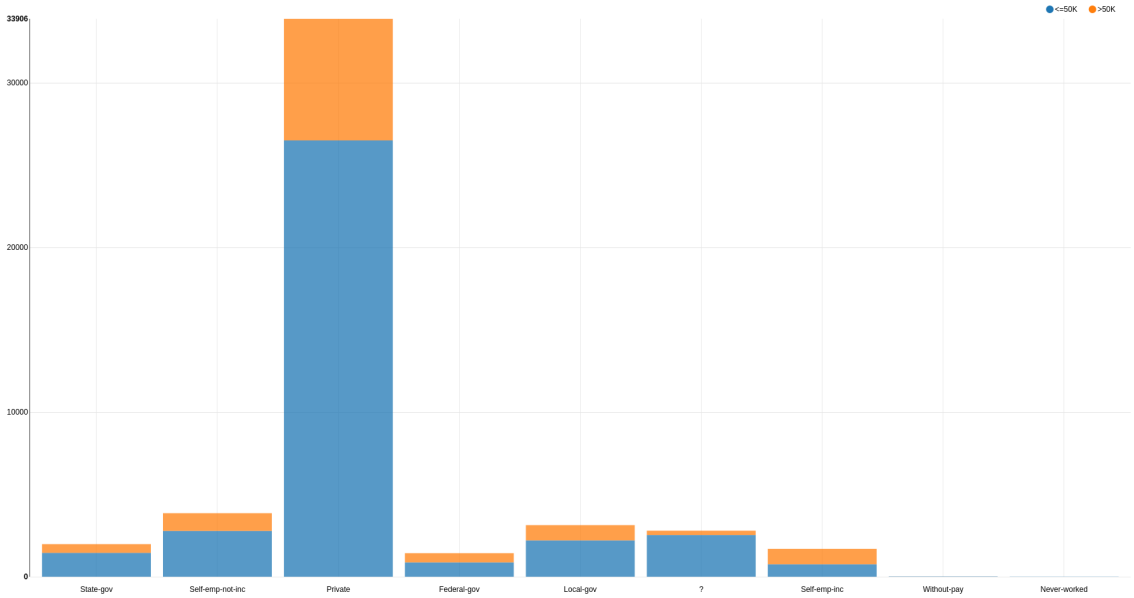

Figura 3.5: Gráfico de barras da classe de trabalho

3.2.5 *Occupation*

Podemos ver que a ocupação do indivíduo afeta significativamente o seu salário.

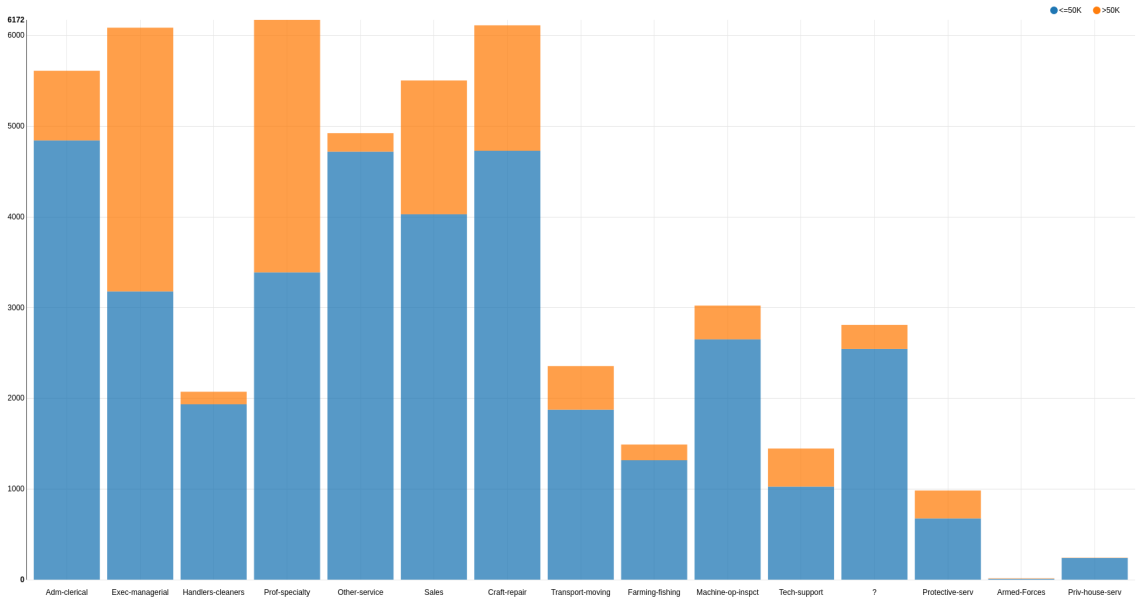

Figura 3.6: Gráfico de barras da classe da ocupação

3.2.6 *Hours per Week*

A maioria dos indivíduos trabalha 40 horas por semana e o número de indivíduos que ganha $> 50K$ aumenta significativamente com as horas de trabalho.

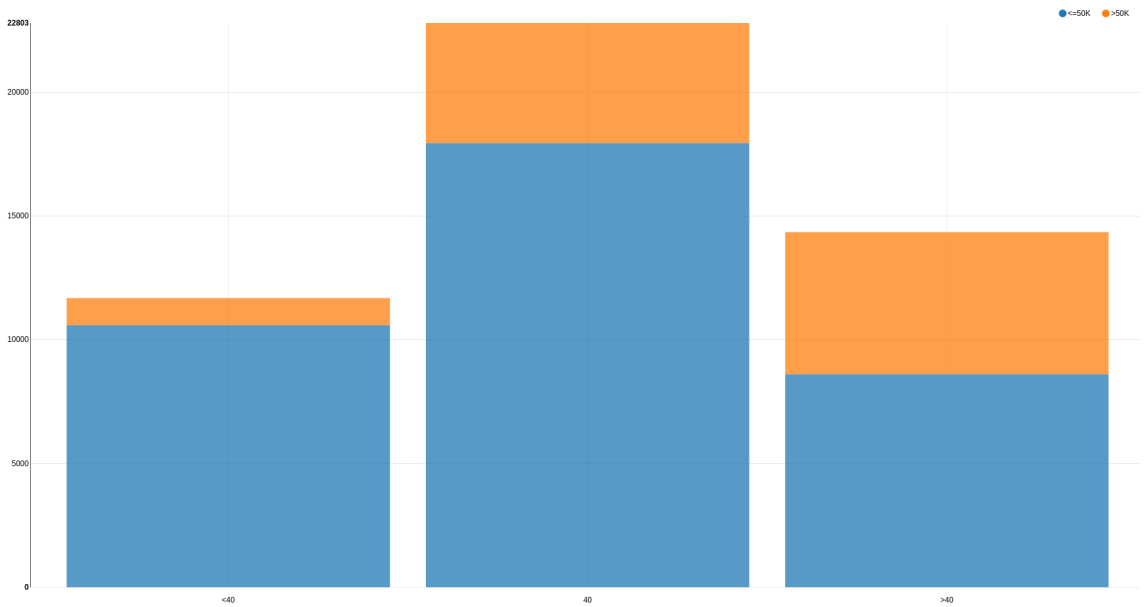

Figura 3.7: Gráfico de barras das horas por semana

3.2.7 *Relationship* e *Marital Status*

Estes dois dados estão diretamente relacionados. Podemos analisar que indivíduos casados têm salários mais elevados.

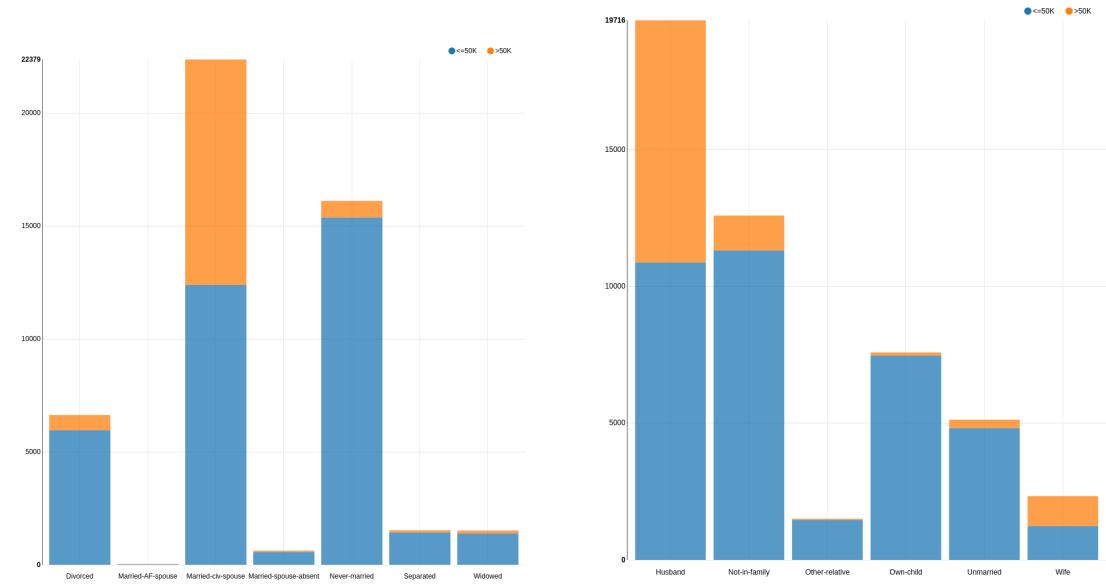

Figura 3.8: Gráficos de barras do relacionamento e estado civil

3.2.8 *Race*

Em relação à raça, podemos ver que existem muitos mais indivíduos de raça branca, e que, geralmente, estes têm um salário mais elevado.

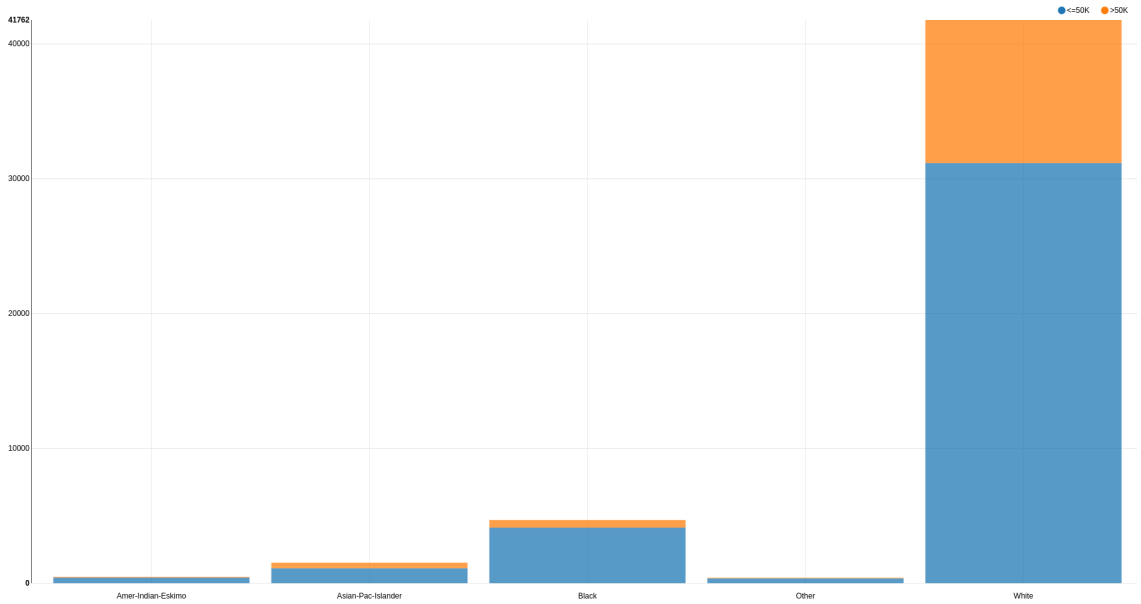

Figura 3.9: Gráfico de barras da raça

3.2.9 *Gender*

No *dataset* existem mais registos de indivíduos do género masculino do que feminino. Os primeiros, em média, possuem salários mais elevados.

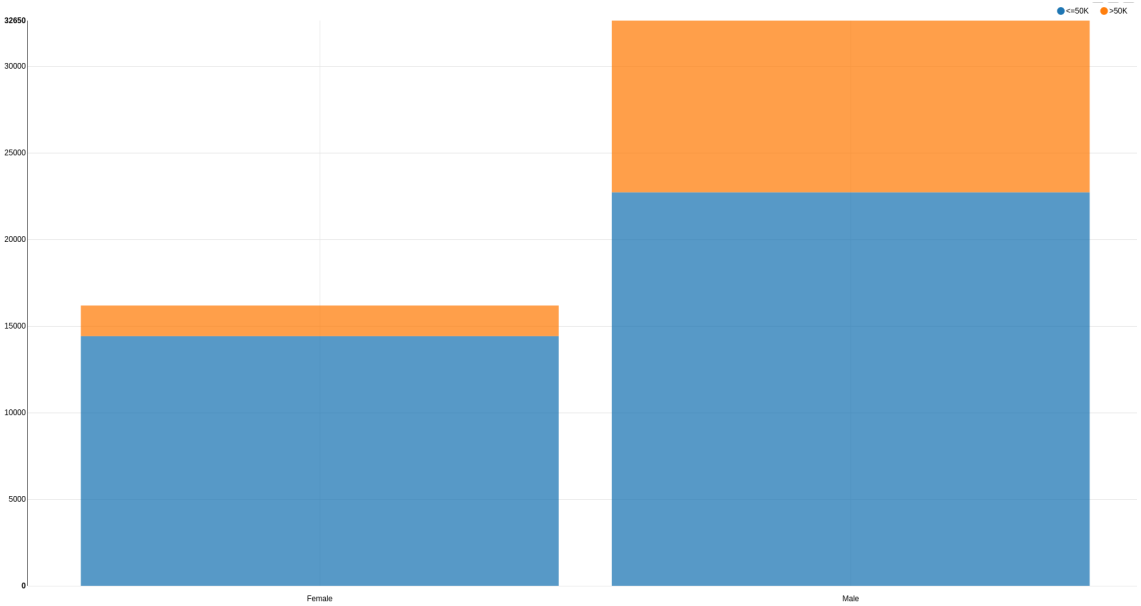

Figura 3.10: Gráfico de barras do género

3.3 Pré-Processamento dos Dados

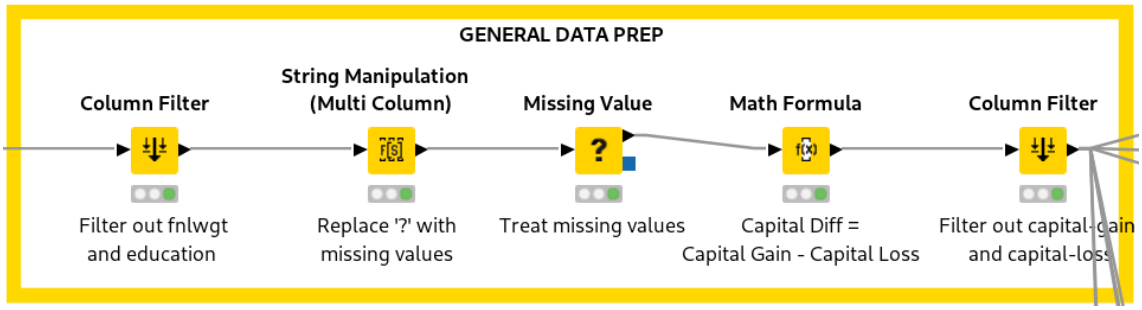

Figura 3.11: Pré-processamento dos dados no *KNIME*

Começamos por remover as colunas desnecessárias, neste caso optamos por remover a coluna *fnlwgt*, visto que os identificadores dos indivíduos são irrelevantes para a classificação do seu salário e a coluna *education* pois, como visto na análise dos dados, esta tem a mesma informação que a coluna *education-num*. Além disso, esta última contém valores numéricos, sugerindo uma ideia de ordem, isto é, quanto menor o número, menor o nível de educação, e quanto maior o número, maior o nível de educação. Este dado pode ser utilizado pelo algoritmo de decisão inteligente para prever de uma melhor forma a relação entre os dados.

Ao analisar o *dataset* reparamos que existem elementos em falta, representados por um ponto de interrogação (*?*). O *KNIME* não os deteta automaticamente como valores em falta, logo temos de os converter, utilizando o nodo *String Manipulation*.

Em seguida, substituímos estes valores em falta. Após alguns testes, decidimos substituir os valores numéricos pela média e os textuais pelo valor mais frequente.

Por fim, o *dataset* original apresentava duas colunas relacionadas: *capital-gain* e *capital-loss*. Comparando as duas, reparamos que só existe *capital-gain* quando não existe *capital-loss* e vice-versa, sabendo isto, podemos juntas as duas colunas numa só em que o resultado é a diferença de ambas ($capital - diff = capital - gain - capital - loss$), para este efeito utilizamos os nodos *Math Formula* seguido do *Column Filter*.

3.4 Modelação

Como forma de controlo de resultados, decidimos testar o *dataset* sem qualquer tratamento de dados adicionais.

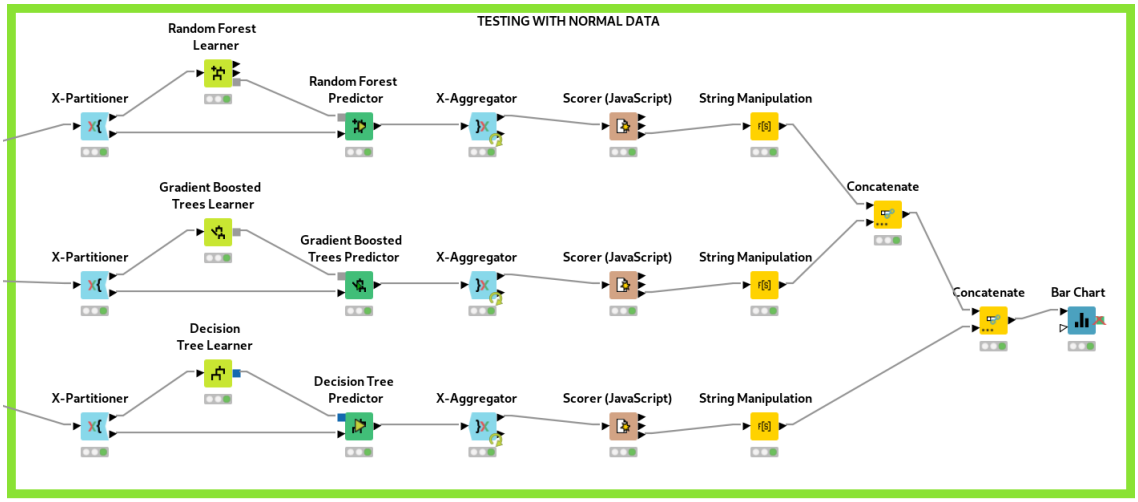

Figura 3.12: Testes com dados originais no *KNIME*

Foram utilizados nodos de *cross validation*, como o *X-Partitioner* e o *X-Aggregator* para criar partições de dados para aprendizagem e para teste.

Para os parâmetros do *X-Partitioner* optamos por utilizar 10 validações e, como existe uma discrepância significativa da percentagem de dados da classe *salary-classification*, foi utilizado o método de *stratified sampling* para manter a proporção de elementos de cada classe nas partições.

Visto que este é um problema de classificação, utilizamos os algoritmos de aprendizagem *Random Forest*, *Gradient Boosted Trees* e *Decision Tree*.

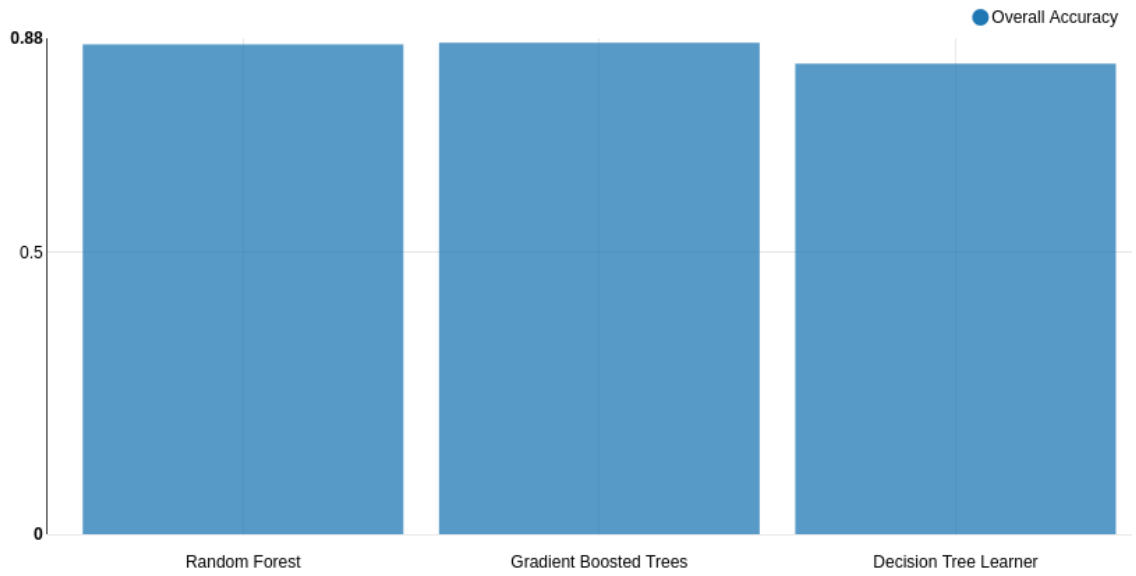

Figura 3.13: Precisão dos vários algoritmos de aprendizagem

Analisando a precisão de cada algoritmo, reparamos que, neste caso, o *Gradient Boosted Trees* obteve um melhor resultado, com 87.20% de precisão.

Confusion Matrix

	<=50K (Predicted)	>50K (Predicted)	
<=50K (Actual)	35153	2002	94.61%
>50K (Actual)	4252	7435	63.62%
	89.21%	78.79%	

Class Statistics

Class	True Positives	False Positives	True Negatives	False Negatives	Recall	Precision	Sensitivity	Specificity	F-measure
<=50K	35153	4252	7435	2002	94.61%	89.21%	94.61%	63.62%	91.83%
>50K	7435	2002	35153	4252	63.62%	78.79%	63.62%	94.61%	70.39%

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
87.20%	12.80%	0.623	42588	6254

Figura 3.14: *Scorer* do algoritmo *Gradient Boosted Trees*

Apesar dos resultados parecerem bons, reparamos que o valor *Recall* e *Precision* para os valores da classe $> 50K$ são um pouco baixos. Isto é algo expectável, visto que a percentagem de dados desta classe neste *dataset* é reduzida.

Para resolver este problema de dados desequilibrados existem algumas soluções, entre elas podemos reduzir os dados da classe que os tem a mais — *downsampling* — ou gerar novos dados da classe que os tem a menos — *upsampling*.

3.4.1 *Downsampling*

Utilizando o nodo *Equal Size Sampling* para remover dados da classe $<= 50K$ obtemos a seguinte distribuição de dados:

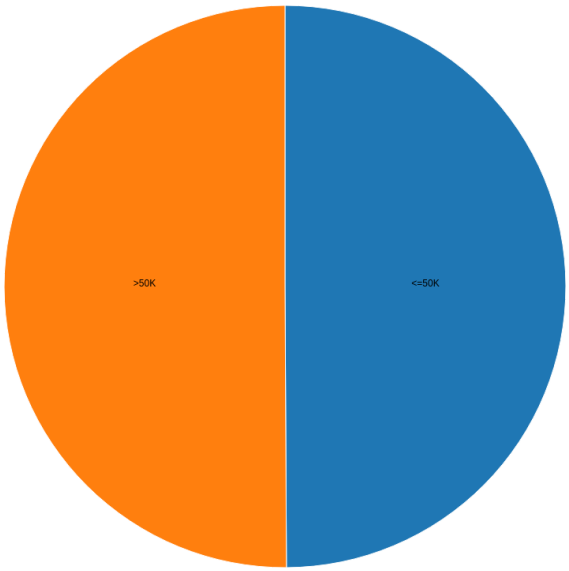

Figura 3.15: Percentagem de dados de cada classe após *downsampling*

Apesar desta distribuição de dados ser bastante melhor para o treino, o *dataset* diminuiu de 48,843 entradas para apenas 23,335 entradas, logo é expectável que seja obtida uma precisão geral mais baixa.

Realizando testes com os algoritmos utilizados anteriormente, apenas alterando o modo de *sampling* para aleatório, dado que o *dataset* já se encontra equilibrado, o algoritmo com melhor desempenho continua a ser o *Gradient Boosted Trees*.

Confusion Matrix

	<=50K (Predicted)	>50K (Predicted)	
<=50K (Actual)	9512	2136	81.66%
>50K (Actual)	1608	10079	86.24%
	85.54%	82.51%	

Class Statistics

Class	True Positives	False Positives	True Negatives	False Negatives	Recall	Precision	Sensitivity	Specificity	F-measure
<=50K	9512	1608	10079	2136	81.66%	85.54%	81.66%	86.24%	83.56%
>50K	10079	2136	9512	1608	86.24%	82.51%	86.24%	81.66%	84.34%

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
83.96%	16.04%	0.679	19591	3744

Figura 3.16: *Scorer* do algoritmo *Gradient Boosted Trees* no *dataset downsampled*

Apesar de a *overall accuracy* ter baixado significativamente, já não temos os problemas de *recall* e *precision* identificados anteriormente, o que significa que, dado uma nova entidade com classificação $> 50K$, este *workflow* tem uma maior probabilidade de o prever corretamente.

Utilizando o meta-nodo *backwards feature elimination* para testar várias combinações de *features* notamos que podemos remover as *features workclass* e *native-country* para atingir uma precisão de 84.15%.

3.4.2 *Upsampling*

Para gerar mais dados utilizamos o nodo *SMOTE*, no entanto, existem alguns problemas com esta abordagem. A geração de dados apenas pode existir para os dados de treino, nunca para os dados de teste isto obriga-nos a não poder utilizar *cross validation*, tendo que particionar o *dataset* em duas partes, uma para treino e outra para testesa. Além disso, o método *SMOTE* apenas funciona para características numéricas, e não para as textuais.

Realizando os testes, o algoritmo *Random Forest* foi aquele que demonstrou melhores resultados, no entanto, não tão bons como os métodos anteriormente demonstrados.

Confusion Matrix

	<=50K (Predicted)	>50K (Predicted)	
<=50K (Actual)	9512	2136	81.66%
>50K (Actual)	1608	10079	86.24%
	85.54%	82.51%	

Class Statistics

Class	True Positives	False Positives	True Negatives	False Negatives	Recall	Precision	Sensitivity	Specificity	F-measure
<=50K	9512	1608	10079	2136	81.66%	85.54%	81.66%	86.24%	83.56%
>50K	10079	2136	9512	1608	86.24%	82.51%	86.24%	81.66%	84.34%

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
83.96%	16.04%	0.679	19591	3744

Figura 3.17: *Scorer* do algoritmo *Random Forrest* no *dataset upsampled*

3.4.3 *Binning*

Como existem muitos valores possíveis para algumas colunas, decidimos efetuar a sua divisão em *bins*.

Visto que há poucos indivíduos do 12.º ano para baixo, e a maioria recebe um salário $<= 50K$, agrupamos todos no mesmo grupo.

Para a idade dividimos em três categorias, baseadas no número de salários elevados:

- Young: $]-\infty, 27[$
- Adult: $[27, 62]$
- Old: $]63, \infty[$

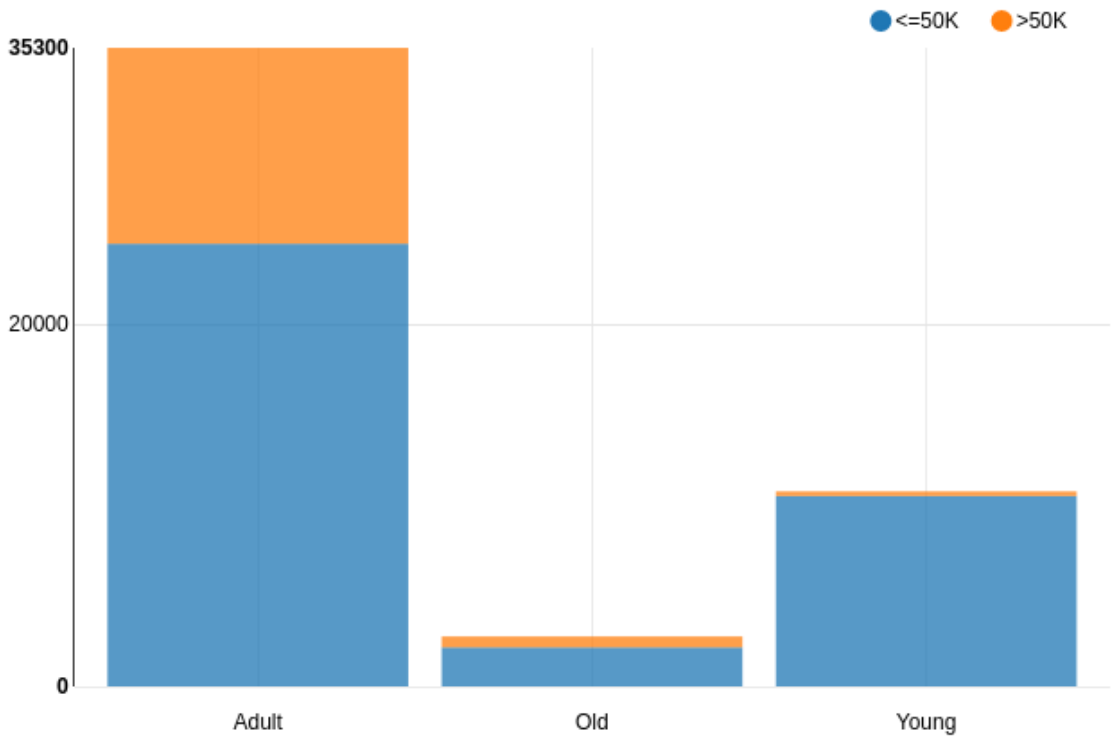

Figura 3.18: Gráfico de barras para os *bins* de idades

A diferença de capital foi dividida em dois *bins*:

- Low: $]-\infty, 5,000]$
- High: $]5,000, \infty[$

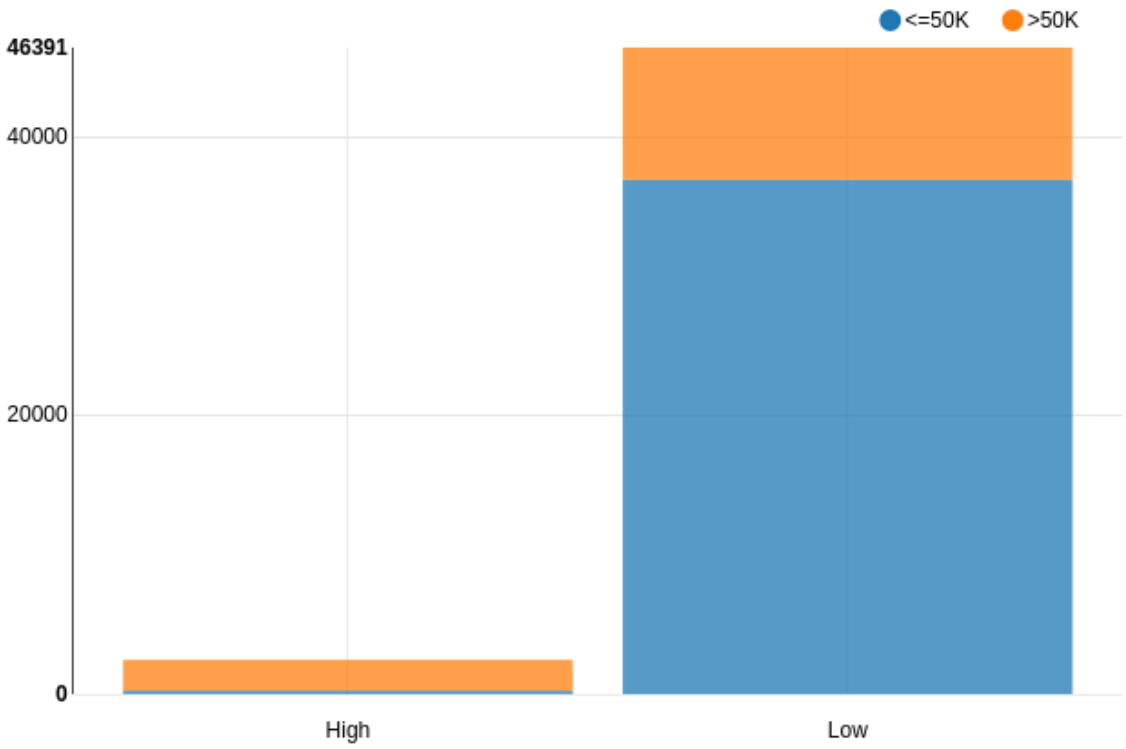

Figura 3.19: Gráfico de barras para os *bins* de diferença de capital

Executando os algoritmos, reparamos que não obtivemos resultados melhores comparando com os resultados de controlo, logo decidimos não otimizar este método.

Confusion Matrix

	<=50K (Predicted)	>50K (Predicted)	
<=50K (Actual)	34290	2865	92.29%
>50K (Actual)	4832	6855	58.65%
	87.65%	70.52%	

Class Statistics

Class	True Positives	False Positives	True Negatives	False Negatives	Recall	Precision	Sensitivity	Specificity	F-measure
<=50K	34290	4832	6855	2865	92.29%	87.65%	92.29%	58.65%	89.91%
>50K	6855	2865	34290	4832	58.65%	70.52%	58.65%	92.29%	64.04%

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
84.24%	15.76%	0.541	41145	7697

Figura 3.20: *Scorer* do algoritmo *Gradient Boosted Trees* no *dataset* com *bins*

3.5 Análise de Resultados

Embora o primeiro modelo tenha uma melhor precisão geral, prevendo corretamente mais vezes num *dataset* semelhante ao *dataset* original, se o *dataset* for equilibrado, consideramos que o segundo modelo, com *downsampling*, seria o mais adequando.

Um parâmetro que também tivemos em conta na análise dos resultados foi o *Cohen's kappa* que indica a confiabilidade dos resultados obtidos. Em todos os modelos que realizamos, este manteve-se entre 0,6 e 0,7. Isto é expectável, pois o *dataset* utilizado não é muito equilibrado em vários parâmetros. Além do parâmetro principal do salário, outros parâmetros como *age*, *education*, *workclass*, *race* e *gender* são também eles bastante desequilibrados.

4. Conclusões

Desta forma, foram aplicados bastantes conceitos associados à aprendizagem, desde as várias maneiras de efetuar o pré-processamento dos dados até à variedade de algoritmos que tivemos que filtrar, conforme a sua adequação aos *datasets* selecionados.

Especificamente relativamente ao *dataset* acerca de previsão de salários, compreendemos que existem dificuldades na aprendizagem associadas a um desequilíbrio das proporções da variável objetivo, isto é, existiam muito mais dados acerca de salários $<= 50K$.

No entanto, consideramos que obtivemos bons resultados com o tratamento que efetuamos aos *datasets* documentados no presente relatório.

5. Referências

- *Dataset* selecionado (Previsão do Preço de Voos)
  https://www.kaggle.com/datasets/shubhambathwal/flight-price-prediction