Aprendizagem e Extração de Conhecimento

Catarina Cruz, Mariana Marques, Miguel Solino, and Paulo Lima

University of Minho, Department of Informatics, 4710-057 Braga, Portugal
[a84011,a85171,a86435,a89983]@alunos.uminho.pt
Grupo 4
10 de Janeiro de 2021

Resumo Este trabalho surgiu no âmbito do segundo trabalho prático da unidade curricular de aprendizagem e extração de conhecimento do perfil de Sistemas Inteligentes da Universidade do Minho. Neste trabalho pretende-se que seja desenvolvido um modelo classificador utilizando o ambiente de desenvolvimento *Python/Sklearn*.

Keywords: SKLearn · Dataset · Data Processing · Data Transformation · Feature Selection · Model Validation

1 Introdução

O problema apresentado consiste em prever o nível salarial anual de um indivíduo. Desta forma, é necessário começar por observar e analisar o dataset fornecido que diz respeito a características de vários indivíduos.

Seguidamente tem de se realizar o tratamento dos dados, para melhorar a performance de classificação do modelo.

Por último, é necessário treinar e validar cada um dos modelos que nos foram apresentados nas aulas.

2 Análise do dataset

Inicialmente, com o objetivo de analisar melhor os dados com que iríamos trabalhar e perceber as relações existentes entre os diversos atributos e o nível salarial anual de um indivíduo, primeiramente é importante perceber quais as características e tipos de dados pelo qual o nosso *dataset* é constituído.

3 Visualização dos dados

O principal objetivo consiste em tratar todos os dados de forma a ser possível relacioná-los com a variável *target*, neste caso o "salary-classification", que adquire o valor " <=50K"caso o nível salarial anual desse individuo seja menor ou igual a 50K e adquire o valor " >50K"caso o nível salarial anual desse individuo seja superior a 50K.

3.1 Informação dos atributos

- **age:** contínuo
- **Workclass:** Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
- **fnlwgt:** contínuo
- **education:** Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
- **education-num:** contínuo

- **marital-status:** Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
- **occupation:** Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.
- **relationship:** Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
- **race:** White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
- **sex:** Female, Male.
- **capital-gain:** contínuo.
- **capital-loss:** contínuo.
- **hours-per-week:** contínuo.
- **native-country:** United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, TrinadadTobago, Peru, Hong, Holand-Netherlands.
- **salary-classification:** é a *target variable* que indica se o nível salarial de um individuo é <=50K ou >50K.

3.2 Gráficos

Nesta secção serão apresentados gráficos para visualização da ocorrência dos atributos do *dataset*, de modo a tentar encontrar alguns padrões que auxiliem a interpretação dos dados e previsão dos resultados.

Heat map

Primeiro, construímos um *heat map*, de forma a perceber a correlação existente entre as várias características do *dataset*.

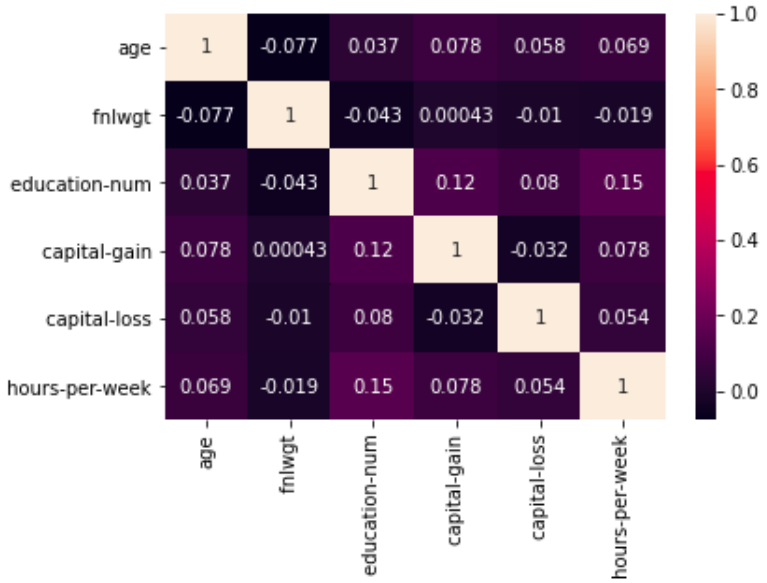

Figura 1. heat map

Nível salarial por idade

Para percebermos se o nível salarial dos funcionários sofria alterações com o decorrer da idade optámos por desenvolver um histograma que mostra as alterações de cada nível salarial para cada idade.

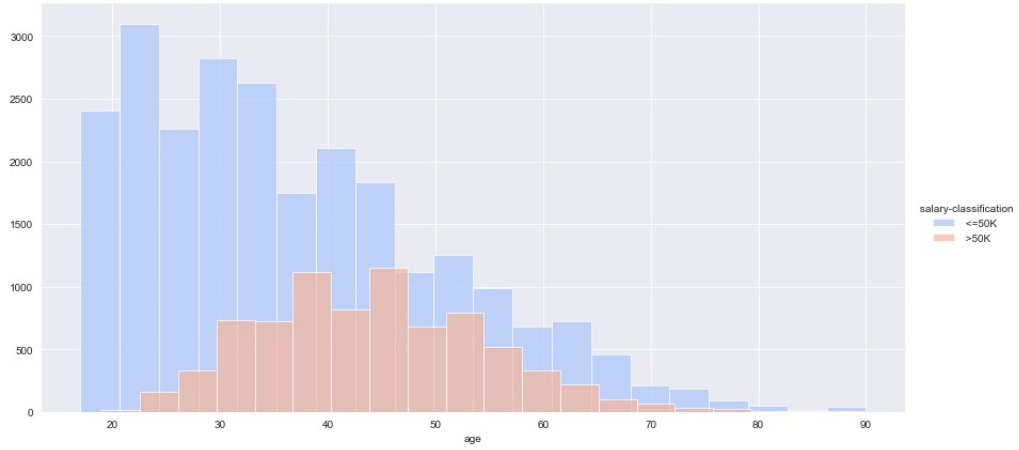

Figura 2. Relação ente Age e a target variable salary-classification

Nível salarial por classe de trabalho

Para percebermos a relação entre o nível salarial e a classe de trabalho optámos por inicialmente fazer uma análise separadamente à variável "workclass"e para isso utilizámos um gráfico circular que nos agrupou os vários tipos de classe de trabalho pelo total de funcionários e nos permitiu também ver a percentagem de cada um e através deste conseguimos concluir que, por exemplo, a maior parte dos indivíduos em estudo pertencem a uma classe de trabalho privada. Seguidamente procedemos à conceção de um histograma para assim percebermos a relação entre o nível salarial e a classe de trabalho e através deste conseguimos observar a quantidade de indivíduos organizados por classe de trabalho para cada nível salarial.

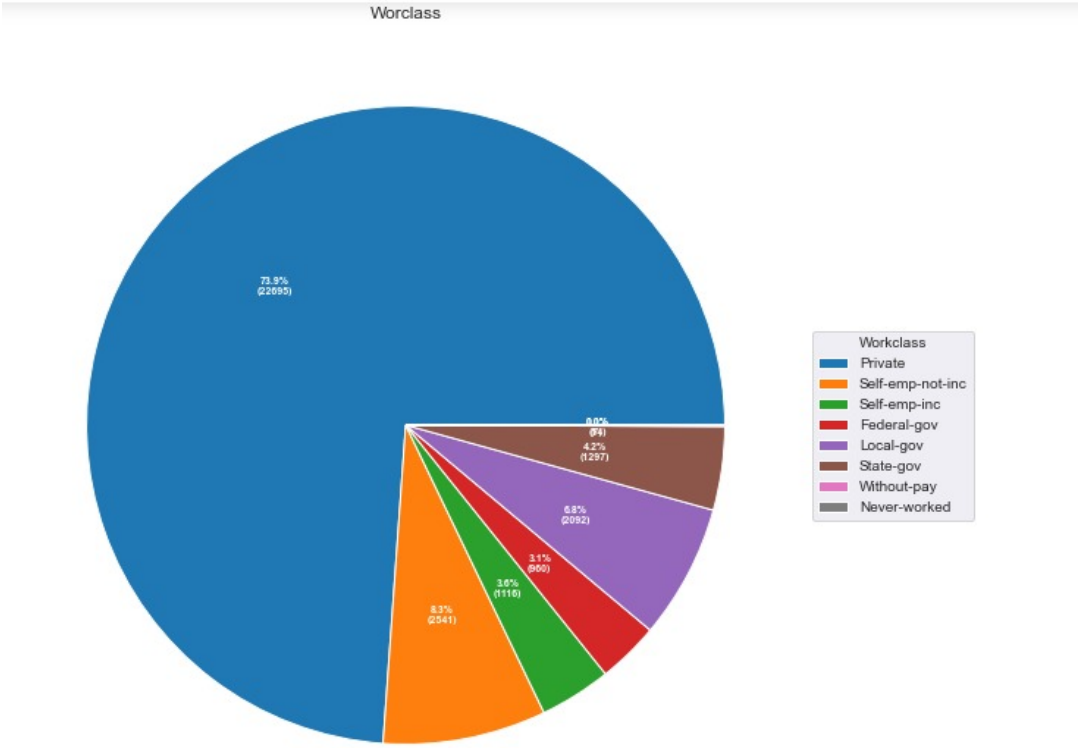

Figura 3. Distribuição de "Workclass"pelos individuos

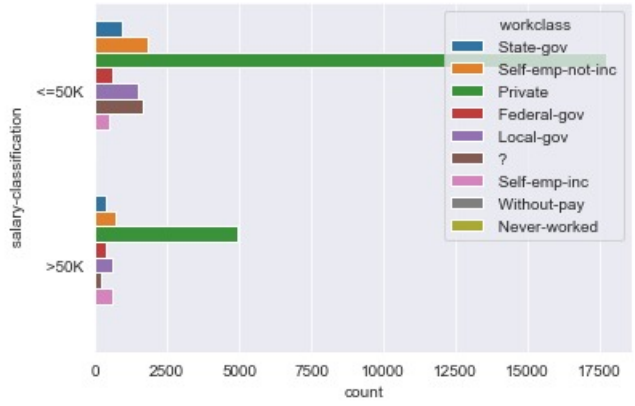

Figura 4. Relação entre "workclass"e a target variable "salary-classification"@

Nível salarial por peso da amostra

Para relacionar os atributos "fnlwgt"com o nível salarial construímos um histograma. Este atributo indica-nos o peso final determinado pela *Census Organization* e, tal como podemos ver, este valor varia entre 0 e 0.8, encontrando-se maioritariamente nos 0.2.

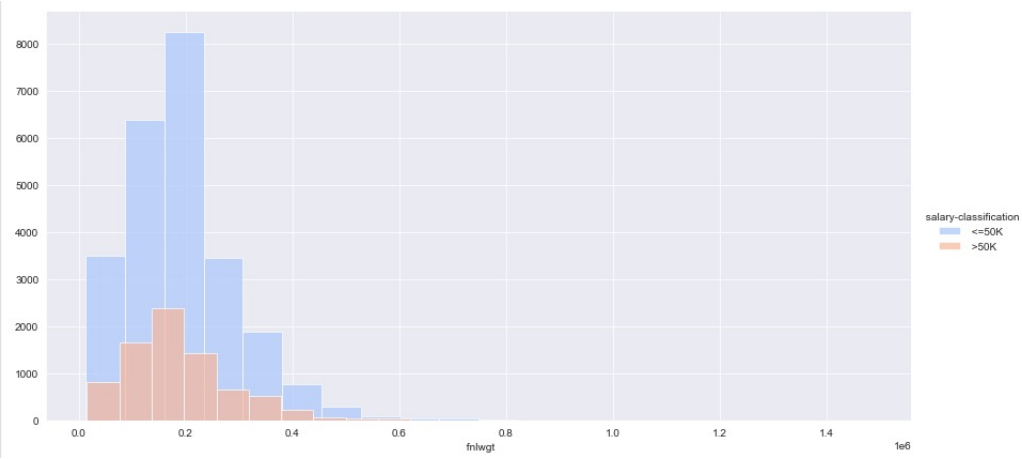

Figura 5. Relação entre "fnlwgt"e a target variable "salary-classification"@

Nível salarial por grau de escolaridade

Para analisarmos a relação entre os vários níveis de escolaridade e o nível salarial inicialmente fizemos uma observação dos dados de nível de educação em separado utilizando um gráfico circular onde observámos, por exemplo, que na contagem total de indivíduos a maioria tem como nível educacional a graduação da escola secundária ("HS-grad").

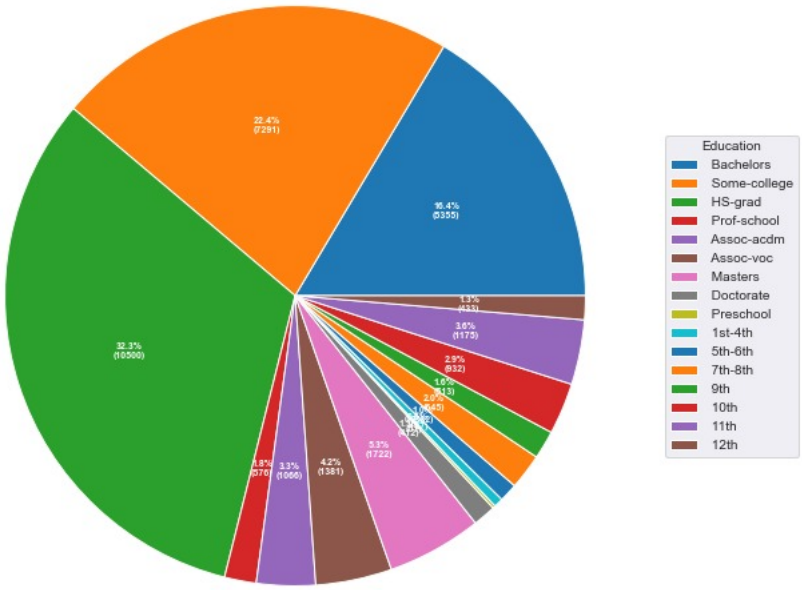

Figura 6. Nível salarial por grau de escolaridade

Nível salarial por grau de escolaridade em formato numérico

Para este dado construímos um histograma que agrupa os indivíduos por nível salarial e anos de educação onde podemos observar que, por exemplo, os indivíduos de nível salarial de " <=50K"na sua maioria fizeram nove anos de estudo e que por outro lado os indivíduos de nível salarial de " >50K"na sua maioria completaram 13 anos de estudos.

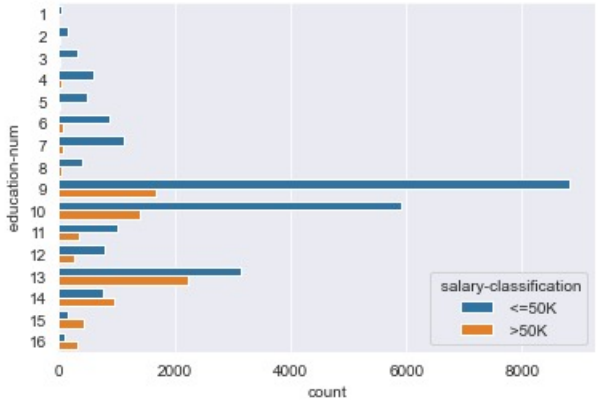

Figura 7. Nível salarial por grau de escolaridade em formato numérico

Nível salarial na relação entre o género e o estado civil

O género de um individuo e o seu estado civil pode influenciar o nível salarial deste. A partir de um histograma que relaciona estes mesmos atributos podes concluir que, por exemplo, a maior parte dos indivíduos do sexo masculino é casado pelo civil enquanto a maior parte dos indivíduos do sexo feminino não são casado.

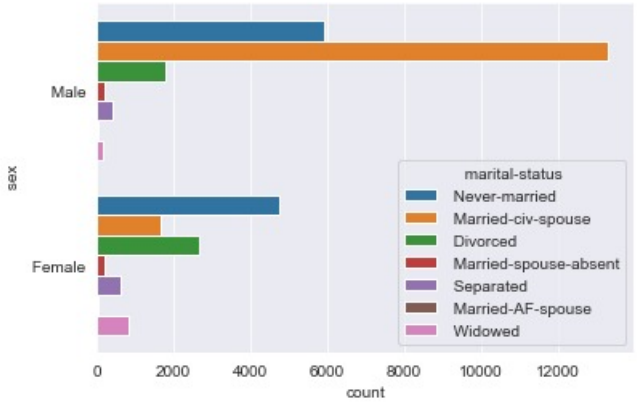

Figura 8. Nível salarial na relação entre o género e o estado civil

Nível salarial relacionado com ocupação

Para relacionar o nível salarial de um indivíduo com a ocupação deste, desenvolvemos um histograma onde podemos observar, por exemplo, que os indivíduos de nível salarial " <=50K"têm grande incidência em todos os campos de ocupação. Enquanto que aqueles com nível salarial " >50K"têm mais incidência em *exec managerial*.

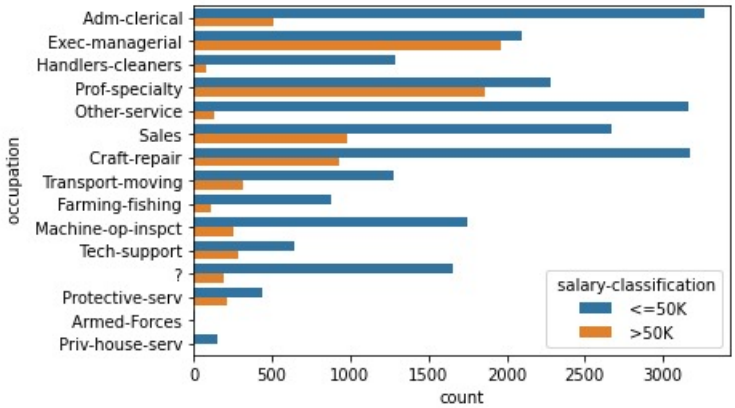

Figura 9. Relação entre o nível salarial e a ocupação

Nível salarial relacionado com a posição do indivíduo dentro do ambiente familiar em que está inserido

Para uma melhor análise do efeitos da variável "relationship"na vida de um individuo optámos por desenvolver um gráfico circular. A partir deste conseguimos percebe-se que a maior parte dos indivíduos em estudo estão na posição de marido numa relação.

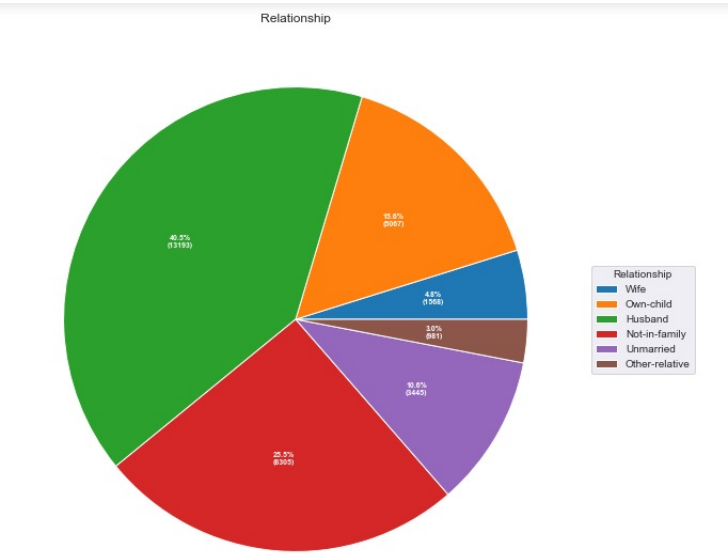

Figura 10. Nível salarial e relacionamento

Nível salarial relacionado com raça

Para perceber em que medida a raça de um individuo afecta o seu nível salarial procedemos a observação de um histograma com estes mesmos dados, onde conseguimos observar que a maior parte dos indivíduos tanto para o nível salarial de " <=50K"como para a nível salarial de " >50K"pertencem à raça branca. No entanto, podemos ainda observar, por exemplo, que para a raça "Amer-Indian-Eskimo", o número de indivíduos com um nível salarial " >50K"é praticamente nulo.

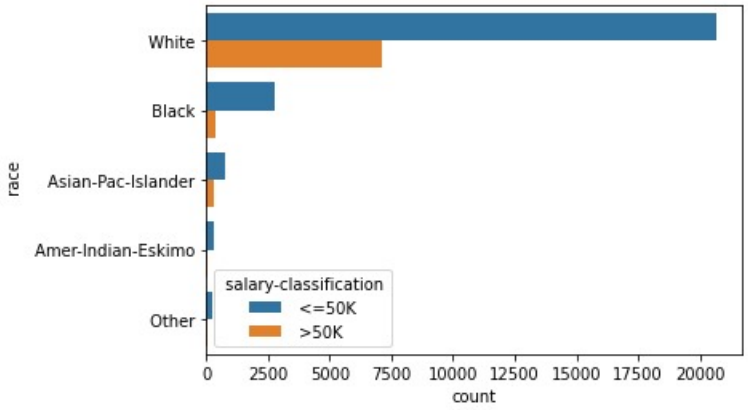

Figura 11. Nível salarial e raça

Nível salarial na relação entre ganho de capital e grau de escolaridade em formato numérico

Para observar a influência no nível salarial na relação entre os atributos "capital-gain"e "education-num", optámos por realizar um gráfico de pontos, onde podemos observar que para indivíduos de nível salarial " <=50K"para além de maioritariamente se encontrarem num grau de escolaridade baixa têm um baixa subida de ganho de capital.

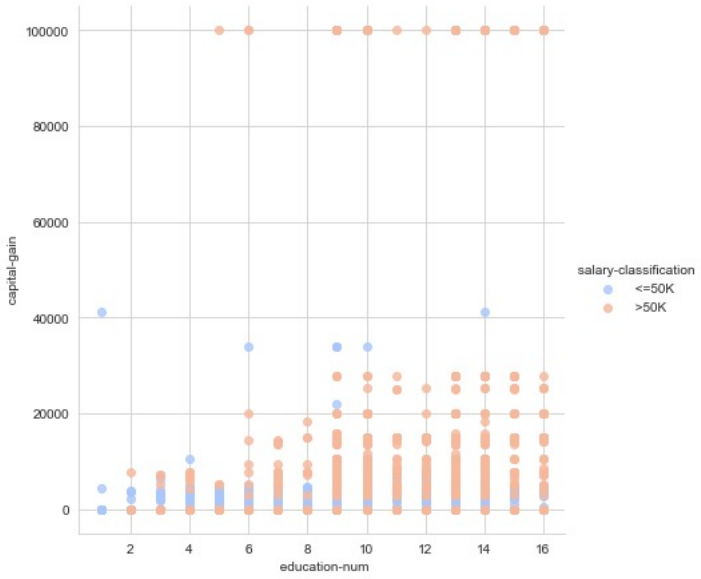

Figura 12. Nível salarial e ganho de capital

Nível salarial na relação entre perda de capital e grau de escolaridade em formato numérico

Em semelhança à análise anterior também aqui optámos por um gráfico de pontos para assim perceber a alteração do nível salarial na relação dos atributos "capital-loss"e "education-num". Desta forma, chegámos à conclusão que contrariamente ao ganho de capital, os indivíduos de salário " <=50K"apresentam uma maior subida na perda de capital.

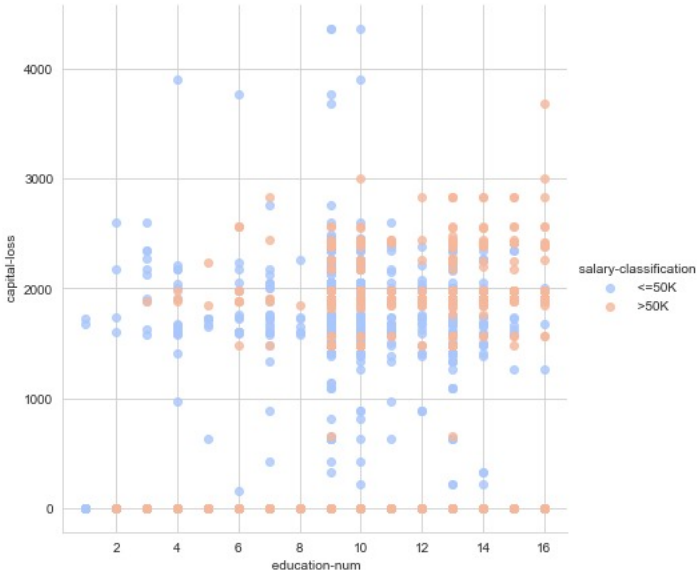

Figura 13. Nível salarial e perda de capital

Nível salarial relacionado com horas semanais de trabalho

Para melhor interpretar de que maneira o nível salarial de um indivíduo é afectado com a variação das horas semanais, procedemos à conceção de um histograma, onde pudemos observar que na fasquia de poucas horas de trabalho semanais são maioritariamente constituídas por indivíduos pertencentes ao nível salarial " <=50K". Com a subida de hora semanais de trabalho nota-se uma subida no número de indivíduos de nível salarial " >50K", no entanto o maior número de indivíduos continua a pertencer ao nível salarial " <=50K".

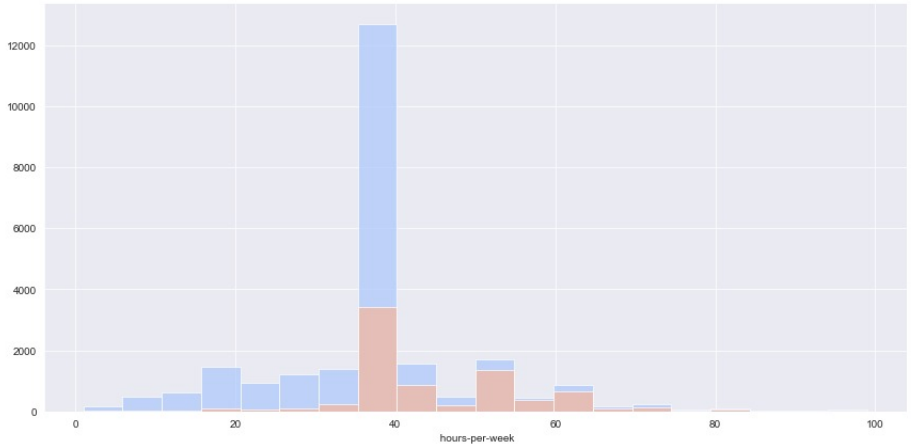

Figura 14. Nível salarial relacionado com horas semanais de trabalho

Nível salarial na relação com o país nativo

Para percebermos a influencia do país nativo no nível salarial de um individuo optámos por construir um mapa onde quanto mais intensa estiver a cor mais população tem essa zona.

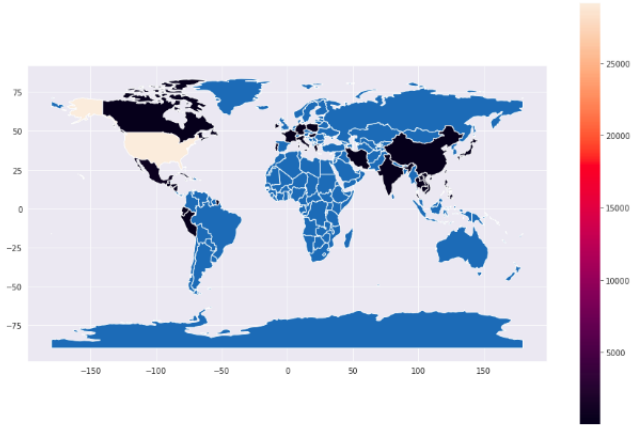

Figura 15. Nível salarial relacionado com país nativo

4 Data Preprocessing

De forma a adequar a informação contida no *dataset* aos modelos de extração de conhecimento que foram utilizados, foi necessário proceder ao seu pré-processamento.

4.1 Tratamento do Dataset

Inicialmente, analisamos o *dataset* onde percebemos que a maior parte dos dados de cada atributo estariam sob a forma de strings pelo que nos causaria problemas na execução do resto das tarefas. Então procedemos à substituição destas strings por inteiros, como está representado de seguida:

```
# label_encoder object knows how to understand word
    labels.label_encoder = preprocessing.LabelEncoder()

training[' workclass'] =
    label_encoder.fit_transform(training[' workclass'])
training[' education'] =
    label_encoder.fit_transform(training[' education'])
training[' marital-status'] =
    label_encoder.fit_transform(training[' marital-status'])
training[' occupation'] =
    label_encoder.fit_transform(training[' occupation'])
training[' relationship'] =
    label_encoder.fit_transform(training[' relationship'])
training[' race'] = label_encoder.fit_transform(training['
    race'])
training[' sex'] = label_encoder.fit_transform(training['
    sex'])
training[' native-country'] =
    label_encoder.fit_transform(training[' native-country'])
training[' salary-classification'] =
    label_encoder.fit_transform(training['
    salary-classification'])

#test
# label_encoder object knows how to understand word labels.
label_encoder = preprocessing.LabelEncoder()
# Encode labels in column 'species'.
test[' workclass']= label_encoder.fit_transform(test['
    workclass'])
test[' education']= label_encoder.fit_transform(test['
    education'])
test[' marital-status']= label_encoder.fit_transform(test['
    marital-status'])
test[' occupation']= label_encoder.fit_transform(test['
    occupation'])
test[' relationship']= label_encoder.fit_transform(test['
    relationship'])
test[' race']= label_encoder.fit_transform(test[' race'])
test[' sex']= label_encoder.fit_transform(test[' sex'])
test[' native-country']= label_encoder.fit_transform(test['
    native-country'])
```

```
test[' salary-classification']=
    label_encoder.fit_transform(test['
    salary-classification'])
```

Para além disso, precisamos ainda de criar duas variáveis para cada um dos *dataset*, uma "target"onde apenas estaria o "salary-classification"e o "data"com os restantes atributos.

4.2 Missing Values Analysis

De seguida, procedemos à análise do *dataset* no sentido de perceber a possibilidade de existência de valores em falta.

Figura 16. Verificação da existência de valores em falta

Ao observar o resultado obtido conclui-se que todos os campos de dados se encontram preenchidos.

4.3 Feature Selection

Existem 3 classes gerais de algoritmos de seleção de features :

- **Filter methods:**

  Neste método é aplicada uma medida estatística para atribuir uma pontuação a cada features. Assim, estas são classificadas pela pontuação e como tal selecionadas para serem mantidas ou não no conjunto de dados. Para aplicação deste algoritmo, optámos por utilizar a função *SelectKBest (Unvariate Selection*). Neste método optamos por usar *Chi-Squared* que é um teste estatístico para valores não negativos para selecionar as top k *features* que pretendemos. Usamos ainda outro método, o *VarianceThreshold*.

```
array([8.60061182e+03, 4.75081192e+01, 1.71147683e+05, 2.97942270e+02,
       2.40142178e+03, 1.12346982e+03, 5.04558854e+02, 3.65914312e+03,
       3.30313051e+01, 5.02439419e+02, 8.21924671e+07, 1.37214589e+06,
       6.47640900e+03, 1.36192560e+01])
```

Figura 17. SelectKBest

```
array([7.30000000e+01, 2.11975372e+00, 1.47242000e+06, 1.49784830e+01,
       6.61868663e+00, 2.26863420e+00, 1.40000000e+01, 2.58163360e+00,
       7.20448827e-01, 2.21369502e-01, 9.99990000e+04, 4.35600000e+03,
       9.80000000e+01, 4.10000000e+01])
```

Figura 18. VarianceThreshold

- **Wrapper methods:**

  Os métodos de *wrapper* consideram a seleção de um conjunto de recursos como um problema de pesquisa, onde diferentes combinações são preparadas, avaliadas e comparadas com outras combinações. Este processo de pesquisa pode ser metódico ou estocástico. O algoritmo que optamos por utilizar foi o *Recursive Feature Elimination* que cria modelos repetidamente e mantém o recurso que permite o melhor e pior desempenho em cada iteração sendo classificado os recursos tendo em conta a ordem da sua eliminação.

```
[ True  True False  True  True  True  True  True  True  True False False
  True False]
[1 1 5 1 1 1 1 1 1 1 4 3 1 2]

array([ 0,  1,  3,  4,  5,  6,  7,  8,  9, 12], dtype=int32)
```

Figura 19. Recursive Feature Elimination

- **Embedded methods**

Os métodos incorporados aprendem quais os recursos que contribuem para a precisão do modelo enquanto este é criado. Decidimos implementar o *Principal Component Analysis* que utiliza álgebra linear para transformar o conjunto de dados.

```
[9.95113633e-01 4.87183945e-03 1.44878129e-05 1.66472783e-08
 1.32821762e-08 5.46332990e-09 1.60533706e-09 1.42986911e-09
 4.49647372e-10 2.17725494e-10 1.75033388e-10 1.68427510e-10
 6.16270754e-11 1.23071689e-11]
```

Figura 20. Principal Component Analysis

```
array([0.16652223, 0.04498153, 0.16339744, 0.04158615, 0.08359418,
       0.06766071, 0.07706281, 0.08304814, 0.01443687, 0.02676704,
       0.09135605, 0.02883226, 0.09281332, 0.01794127])
```

Figura 21. Feature Importance

Ao observarmos todos os resultados para cada método concluímos que o mais adequado a utilizar seria o *SelectKBest*.Este remove todos os recursos, exceto os k com pontuação mais elevada. Desta forma, procedemos ao cálculo do K ótimo , ou seja, o k que proporciona melhores resultados a nível da *accuracy*.

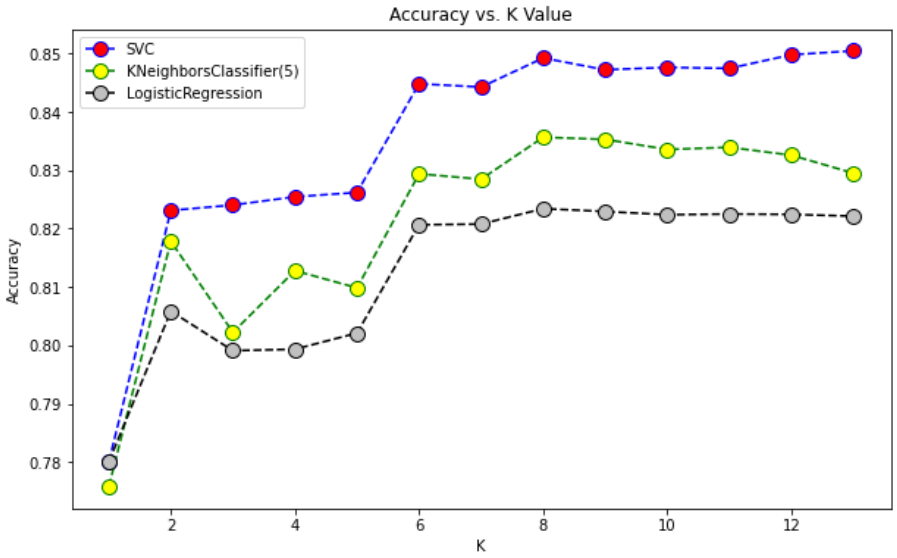

Figura 22. Pesquisa pelo KBest

Ao observar o gráfico conclui-se que o nível de *accuracy* está mais elevado ao aproximar-se de K=8, pelo que escolhemos este mesmo valor. Assim, apenas oito atributos do *dataset* são selecionados para permanecer no *dataset*. Para melhor compreender quais seriam esses oito atributo, optámos por calcular a *feature importance* de 8 atributos do *dataset* acompanhado do seu *status* e acompanhado ainda pelo nome de atributo

	Features	Score
10	capital-gain	8.219247e+07
11	capital-loss	1.372146e+06
2	fnlwgt	1.711477e+05
0	age	8.600612e+03
12	hours-per-week	6.476409e+03
7	relationship	3.659143e+03
4	education-num	2.401422e+03
5	marital-status	1.123470e+03

Figura 23. Feature importance para os melhores oito atributos

Como podemos observar os valores de *score* acompanham a matriz anteriormente calculada.Desta forma, os stributos escolhidos através de *feature selction* são "capital-gain", "capital-loss", "fnlwgt", "age", "hours-per-week", "relationship", "education-num", "marital-status".

4.4 Data Transformation

Seguidamente, procedemos à transformação dos dados, analisando e avaliando a aplicação de três diferentes técnicas, sendo elas :

- **Discretização**: Utilizada para reduzir o número de valores de um atributo contínuo, dividindo-o em intervalos.
- **Standardização**: Corresponde ao processo de aproximar os dados de uma distribuição normal.
- **Normalização**: Corresponde a escalar individualmente cada um dos registo, de forma a que estes passem a ter norma unitária.

De seguida, apresentam-se os resultados da aplicação das técnicas descritas. Estes foram obtidos através de *cross validation (5-fold)*, com o objetivo de obter várias métricas e não apenas uma, como se obteria se a aplicação das técnicas fosse avaliada através da precisão da previsão do *dataset* de teste. Além disto, foram testados diferentes modelos, sem modificações relevantes sobre os seus parâmetros por defeito. Como estamos perante um *dataset* desbalanceado, avaliamos também a métrica *AUROC*, que consiste na certeza com que a *accuracy* é dada.

Começamos por avaliar os resultados da aplicação da Standardização. Para tal, optamos por utilizar o *StandardScaler* e o *RobustScaler*.

	Standard Scaling		Robust Scaling	
	Accuracy	AUROC	Accuracy	AUROC
Logistic Regression	0.825 (+/- 0.002)	0.854	0.826 (+/- 0.003)	0.854
SVC	0.848 (+/- 0.008)	0.892	0.803 (+/- 0.003)	0.838
KNeighbors Classifier	0.815 (+/- 0.008)	0.790	0.841 (+/- 0.006)	0.817
GaussianNB	0.804 (+/- 0.010)	0.857	0.799(+/- 0.008)	0.855

Figura 24. Resultados obtidos com Standardização das caraterísticas

Analisando a figura, conseguimos concluir que os resultados obtidos pelo *StandardScaler* são ligeiramente melhores que os resultados obtidos pelo *RobustScaler*. Além disso, os valores de accuracy foram bastante elevados, sendo acompanhados de um valor de *AUROC* igualmente satisfatório, o que nos indica que os modelos não estão a ser tendenciosos.

Analisamos ainda os valores obtidos a partir da discretização dos dados, construindo uma tabela que inclui para modelo o seu valor de *accuracy* e de *AUROC*.

	Discretization	
	Accuracy	AUROC
Logistic Regression	0.759 (+/- 0.000)	0.507
SVC	0.759 (+/- 0.000)	0.499
KNeighbors Classifier	0.741 (+/- 0.003)	0.571
GaussianNB	0.764 (+/- 0.004)	0.757

Figura 25. Resultados obtidos com discretização das caraterísticas

Por último, procedemos à análise da aplicação da normalização. Tal como nos outros casos construímos uma tabela onde se encontram para cada modelo os valores de *accuracy* e *AUROC* decorrente da aplicação da normalização dos dados.

	Normalization	
	Accuracy	AUROC
Logistic Regression	0.759 (+/- 0.000)	0.581
SVC	0.773 (+/- 0.007)	0.713
KNeighbors Classifier	0.784 (+/- 0.002)	0.785
GaussianNB	0.771 (+/- 0.005)	0.740

Figura 26. Resultados obtidos com a normalização dos valores das caraterísticas

Após a análise destas três técnicas normalização, discretização e standardização concluímos que aquele que demonstrava melhor resultados de *accuracy* e *AUROC* seria a standardização, neste caso *StandardScaler*.

4.5 Dataset Balancing

O conjunto de dados de treino disponibilizado encontra-se desbalanceado, conforme se pode concluir com a figura :

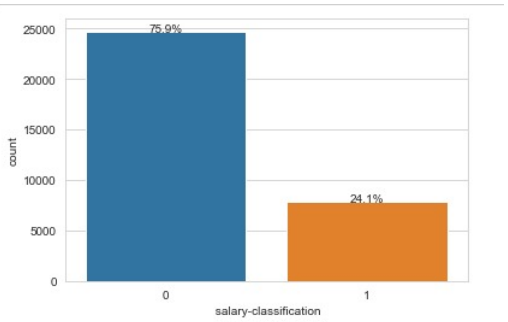

Figura 27. Número de registos por valor do atributo "salary-classification"

Como podemos observar o número de dados relativos ao nível salarial " <=50k"(75,9% dos dados) é muito superior ao número de dados no nível salarial " >=50K"(24,1% dos dados). Ora, se nada for feito em contrário, ao treinar um modelo classificador com este conjunto de dados, este tenderá a classificar todas as observações futuras como pertencendo à classe maioritária. No entanto, a *accuracy* deste classificador pode ser elevada (quando o conjunto de teste também é desbalanceado), o que poderá induzir em erro. Deste modo, convém identificar outras métricas que nos permitam avaliar se a performance de um classificador é adequada ao problema.

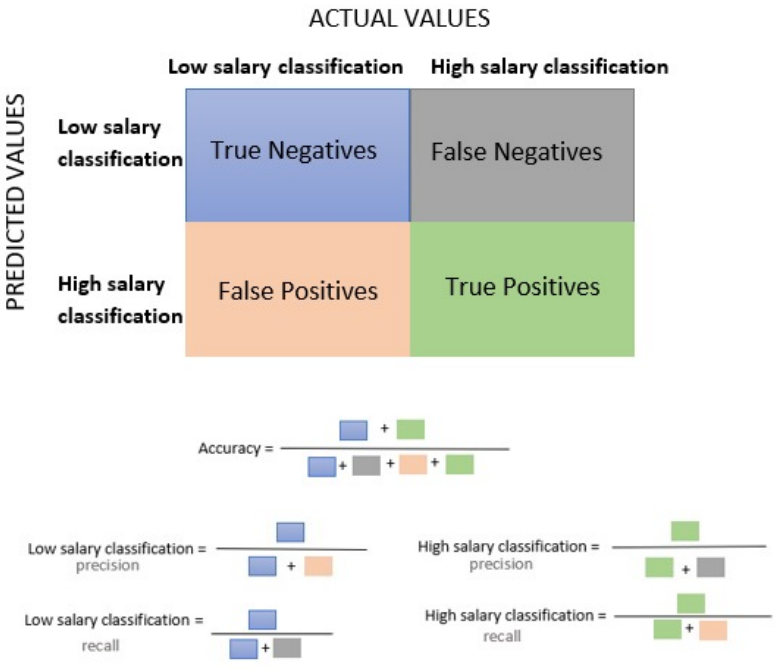

Figura 28. Matriz de confusão

Ao observar a figura podemos concluir que a métrica *recall* representa quantos registos de uma determinada classe conseguimos classificar corretamente, enquanto que a métrica precision representa a taxa de acerto das previsões relativas a uma classe. Ou seja, enquanto que a primeira quantifica a capacidade que o modelo apresenta para identificar uma classe, a segunda indica quão confiável são as previsões de uma classe.

Quando o modelo é tendencioso para a classe maioritária "salary-classification", a maioria das suas classificações será *true positive* ou *false positive*, quando prevê que um indivíduo tem um nível salarial <=50K, quando na verdade não o tem. Por outro lado, a medida *recall* para a classe minoritária será muito baixa, assim como a precisão com que classificamos estas observações. Desta forma, para se balancear o modelo é necessário obter melhores valores para estas métricas. É ainda de notar que é preferível prever que um indivíduo receba um nível salarial <=50K e assim o suceder, do que o contrário, ou seja, prever que um indivíduo recebe um nível salarial >50K e este pertencer ao nível <=50K. Em suma, pretendemos reduzir o número de falsos positivos, sem que isso aumente de forma significativa o número de falsos negativos.

Estratégias para balancear o conjunto de dados

As estratégias que optamos por utilizar para o balanceamento dos dados são os seguintes:

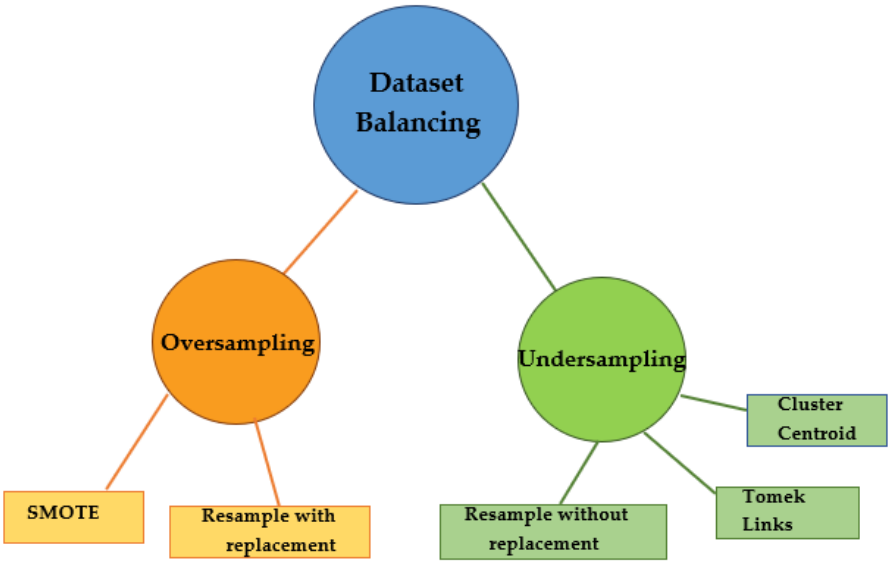

Figura 29. Procedimentos para balancear conjuntos de dados

O balanceamento de dados pode ser encarado a partir de duas abordagens distintas:

***oversampling* ->** igualar o número de observações da classe minoritária ao número de observações da classe maioritária.
***undersampling* ->** reduzir o número de observações da classe maioritária ao número de observações da classe minoritária

No contexto de *oversampling*, a forma mais simples para balancear um conjunto de dados consiste em, simplesmente, repetir múltiplas vezes as observações da classe minoritária (*resample with replacement*). Outra forma, *Synthetic Minority Oversampling Technique (SMOTE)*, passa por considerar pontos vizinhos e interpolar a sua informação de forma a que uma nova observação seja gerada.

No contexto de *undersampling*, podemos simplesmente selecionar aleatoriamente exemplos da classe maioritária (*resample without replacement*), ou então recorrer a técnicas como *Tomek Links* ou *Cluster Centroid*. Com *Cluster Centroid* são gerados centroides através da aplicação de algoritmos de *clustering (K-means)*. A técnica *Tomek Links*, ao contrário das restantes, não procura igualar o número de observações de ambas as classes. Um *Tomek link* é um par de observações muito próximas, mas que pertencem a classes opostas. Esta técnica procura identificar este tipo de ligações e remover a observação que pertença à classe maioritária. Desta forma, aumentamos a separação entre as duas classes promovendo um processo de classificação mais simples.

Depois da aplicação de todas estas estratégias optámos por construir uma tabela com todos os valores do resultado.

	Técnica	Accuracy	AUROC	'Low salary classification' (Recall, Precision)	'High salary classification' (Recall, Precision)
Logistic Regression	SMOTE-ENN	0.736	0.860	(0.800, 0.644)	(0.692, 0.644)
SVC	Tomek links	0.783	0.657	(0.990, 0.778)	(0.195, 0.778)
KNeighbors Classifier	SMOTE-ENN	0.973	0.982	(0.977, 0.959)	(0.971, 0.959)
GaussianNB	Tomek links	0.787	0.843	(0.954, 0.798)	(0.311, 0.798)

Figura 30. Resultados alcançados por técnicas de balanceamento

Como podemos ver, foram escolhidos para dois modelos a técnica *SMOTE-ENN* e para outros dois modelos a técnica que permitia melhor resultados seria a técnica *Tomek links*, pelo que optámos por aplicar nos modelos a técnica de balanceamento de dados *Tomek links*.

5 Model Validation

Após todo o processo de estudo dos dados fornecidos e após tratar do pré-processamento dos dados, tanto a nível da sua transformação como no que diz respeito à seleção das características mais importantes, procedemos à avaliação de modelos, tendo como objetivo obter aquele que possibilitasse atingir uma maior *accuracy* e por isso para cada modelo analisámos a sua respetiva matriz de confusão. Com o intuito de termos mais informação para analisar, optámos ainda por fazer também uma matriz relatório de classificação.

Ao longo das fases precedentes, fomos utilizando diferentes modelos para avaliar a eficácia das técnicas a serem estudadas, entre eles:

- Logistic regression;
- Support Vector Machines;
- K Means Clustering;
- K Nearest Neighbours;
- Gaussian Naive Bayes.

Para melhor perceber qual o modelo que apresentaria melhores resultados para o nosso conjunto de dados já tratados procedemos à aplicação de cada um destes modelos, bem como a construção de matrizes de confusão e de relatórios de classificação e assim avaliar cada modelo não só tendo em conta a sua *accuracy* mas tendo em conta também outros fatores como por exemplo *precision* e *recall*.

	precision	recall	f1-score	support
0	0.82	0.93	0.87	12435
1	0.58	0.32	0.41	3846
accuracy			0.79	16281
macro avg	0.70	0.62	0.64	16281
weighted avg	0.76	0.79	0.76	16281

Figura 31. Relatório de classificação do modelo Logistic Regression

```
[[11551   884]
 [ 2594  1252]]
```

Figura 32. Matriz de confusão do modelo Logistic Regression

	precision	recall	f1-score	support
0	0.79	1.00	0.88	12435
1	0.96	0.16	0.27	3846
accuracy			0.80	16281
macro avg	0.88	0.58	0.58	16281
weighted avg	0.83	0.80	0.74	16281

Figura 33. Relatório de classificação do modelo Support Vector Machines

```
[[12412    23]
 [ 3235   611]]
```

Figura 34. Matriz de confusão do modelo Support Vector Machines

	precision	recall	f1-score	support
0	0.76	0.74	0.75	12435
1	0.23	0.26	0.25	3846
accuracy			0.62	16281
macro avg	0.50	0.50	0.50	16281
weighted avg	0.64	0.62	0.63	16281

Figura 35. Relatório de classificação do modelo K Means Clustering

```
[[9158 3277]
 [2853  993]]
```

Figura 36. Matriz de confusão do modelo K Means Clustering

```
The optimal number of neighbors is 2
```

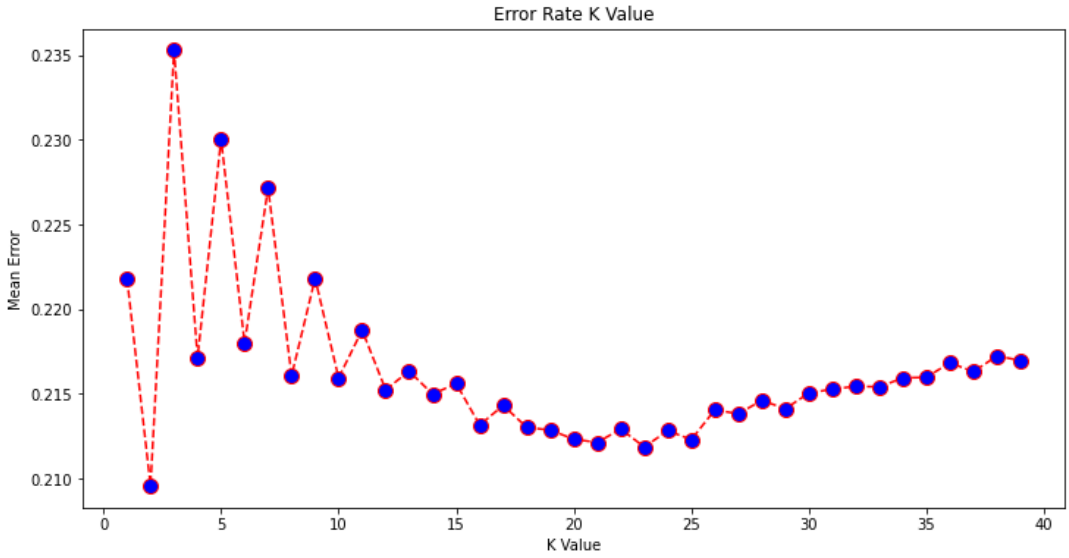

Figura 37. Número ótimo de neighbors (K)

Para que o modelo desse o melhor resultado possível para os dados fornecidos, optámos por calcular o melhor k, ou seja, o número ótimo de *neighbors*.

	precision	recall	f1-score	support
0	0.81	0.94	0.87	12435
1	0.59	0.27	0.37	3846
accuracy			0.78	16281
macro avg	0.70	0.61	0.62	16281
weighted avg	0.76	0.78	0.75	16281

Figura 38. Relatório de classificação do modelo K Nearest Neighbors

```
[[11730   705]
 [ 2813  1033]]
```

Figura 39. Matriz de confusão do modelo K Nearest Neighbors

	precision	recall	f1-score	support
0	0.82	0.95	0.88	12435
1	0.64	0.31	0.41	3846
accuracy			0.80	16281
macro avg	0.73	0.63	0.65	16281
weighted avg	0.77	0.80	0.77	16281

Figura 40. Relatório de classificação do modelo Gaussian Naive Bayes

```
[[11764   671]
 [ 2666  1180]]
```

Figura 41. Matriz de confusão do modelo Gaussian Naive Bayes

Ao observar a matriz de confusão e o relatório de classificação de cada modelo podemos perceber que :

	Accuracy	'Low salary classification' (Recall, Precision)	'High salary classification' (Recall, Precision)
Logistic Regression	0.79	(0.93, 0.82)	(0.33, 0.59)
Support Vector Machines	0.80	(1.00, 0.79)	(0.16, 0.96)
KMeans Clustering	0.62	(0.74, 0.76)	(0.26, 0.23)
KNeighbors Classifier (2)	0.78	(0.94, 0.81)	(0.27, 0.59)
GaussianNB	0.80	(0.95, 0.82)	(0.31, 0.64)

Figura 42. Tabela com dados de todos os modelos

A partir dos dados obtidos, podemos então concluir que, tendo em conta a *accuracy* mais elevada, o melhor modelo é o "Support Vector Machines"bem como o "GaussianNB", visto que ambos apresentam a mesma *accuracy*. Se para além desta métrica, também observarmos a *Recall* e a *Precision*, consideramos que o melhor modelo é o *GaussianNB*, uma vez que, apesar de, às vezes, os valores deste modelo serem mais baixos, são mais balanceados e como tal dão um resultado mais preciso e melhor.

6 Conclusão

A realização deste trabalho permitiu a análise detalhada das diferentes fases de desenvolvimento de um modelo para previsão sobre o conjunto de dados fornecido.

A tarefa que nos foi atribuída tornou-se mais difícil devido ao desbalanceamento dos dados, o que poderia levar a resultados tendenciosos na avaliação dos modelos, fornecendo um falso elevado valor de *accuracy*. Posto isto, deparámo-nos ainda com o desafio de aplicar várias técnicas ao nível da *Feature Selection* e da *Data Transformation*, para que no fim pudéssemos aplicar corretamente todos os modelos ao conjunto de dados e comparar a sua *accuracy*.

Podemos assim afirmar que ao longo do desenvolvimento do projeto foi notório que as fases iniciais são de extrema importância para a conceção do projeto uma vez que estas permitem uma análise cuidada do *dataset* e definir as melhores estratégias de tratamento do mesmo.

Referências

1. RAHEEL SHAIKH, *"Feature Selection Techniques in Machine Learning with Python"Medium, Towards Data Science, 28 Oct. 2018*,
`https://towardsdatascience.com/feature-selection`
Last accessed 27 Dez 2020.

SAGAR RAWALE., *Feature Selection Methods in Machine Learning.*,
`https://medium.com/feature-selection-methods`
Last accessed 27 Nov 2020

SCIKIT-LEARN, *Scikit-Learn 0.22 Documentation.*,
`scikitlearn.org/stable/modules/preprocessing.html.`
Last accessed 27 Nov 2020

KANGLE, *Resampling Strategies for Imbalanced Datasets*,
`www.kaggle.com/rafjaa/resampling-strategies-for-imbalanced-datasets.`
Last accessed 29 Nov 2020