Trabalho prático 2

Carolina Marques - PG42818
Constança Elias - PG42820
Maria Araújo - PG42844
Renata Ribeiro - A86271

Universidade do Minho
Departamento de Informática
www.di.uminho.pt

Abstract. (.....)

Keywords: Sklearn · Predictive analysis · Feature

1 Introdução

(...)

Para a resolução do problema vamos analisar, preparar e visualizar a distribuição do dataset, de modo a interpretar a sua relação com a variável a prever (i.e., salary-classification). Tendo em conta esta análise, o processo seguinte passa por desenvolver diferentes modelos de classificação e validar a sua performance. No final, seleccionamos justificando, o modelo que apresenta melhor performance de classificação.

1.1 Estrutura do relatório

Na secção 2, apresenta-se uma contextualização do problema proposto. Na secção 3, explicamos a análise e Tratamento de dados realizada, mostrando as metodologias utilizadas para a preparação dos dados. Em seguida, na secção 4, apresentamos os modelos preditivos que consideramos interessantes aplicar a este dataSet. Na secção 5, apresentamos uma breve validação dos modelos, bem como a explicação do modelo que presenta melhor performance de classificação. Este relatório termina com uma secção de conclusões que apresenta uma breve reflexão do trabalho realizado.

2 Contextualização

Com este trabalho, pretende-se encontrar uma solução para o problema, proposto no enunciado e que envolve a: *Preparação e análise de um dataset relativo às características de funcionários de múltiplas empresas, como forma de prever o nível salarial anual do um individuo.*

Para isso, vamos desenvolver um modelo de classificação utilizando o ambiente de desenvolvimento Python e aplicando as funcionalidades da biblioteca sklearn.

O dataSet fornecido, possui as seguintes *features*:

- **Age** - Idade do indivíduo (Valor inteiro positivo);
- **Workclass** - Situação de emprego de um indivíduo (Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked);
- **Fnlwgt** - Número de pessoas que o censo acredita que a entrada representa (Valor inteiro positivo);
- **Education** - Grau de escolaridade do indivíduo (Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool);
- **Fducation-num** - Nível de educação (Valor inteiro positivo);
- **Marital-status** - Estado civil do individuo (Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse);
- **Occupation** - Ocupação profissional do individuo (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** - Representa o que o individuo é em relação aos outros (Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried);
- **Race** - Raça do individuo (White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black);
- **Sex** - Género do individuo (Male, female);
- **Capital-gain** - Capital ganho por um individuo;
- **Capital-loss** - Capital perdido por um individuo;
- **Hours-per-week** - Número de horas de trabalho por individuo (Valor inteiro positivo).
- **Native-Country** - País de origem do indivíduo (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);

Atavés destes dados, pretende-se prever-se a feature:

- **Salary-classification** - indica se o funcionário ganha ou não mais de $50,000 anualmente.

3 Análise e Preparação dos dados

Inicialmente foi necessário analisar os dados de modo a efectuar um pré-processamento dos dados brutos para que na próxima etapas seja possível aplicar os modelos e extrair resultados conclusivos.

Com a consulta de algumas informações relativas ao dataset, e visualização de alguns gráficos, obtemos algumas informações:

- O dataset de treino tem aproximadamente o dobro do tamanha do de teste, é composto por 32561 linhas enquanto que o de teste possui 16281. Ambos tem 15 colunas.
- As features, *age, fnlwgt, education_num, capital_gain, capital_loss e hours_per_week* são numéricas. Por sua vez, *workclass, education, marital_satus, occupation, relationship, rece, sex, native_country e salary_classification* são variáveis categóricas.
- Nas variáveis numéricas observamos os seguintes dados estatísticos.

```
data.describe()
```

	age	fnlwgt	education_num	capital_gain	capital_loss	hours_per_week
count	32561.000000	3.256100e+04	32561.000000	32561.000000	32561.000000	32561.000000
mean	38.581647	1.897784e+05	10.080679	1077.648844	87.303830	40.437456
std	13.640433	1.055500e+05	2.572720	7385.292085	402.960219	12.347429
min	17.000000	1.228500e+04	1.000000	0.000000	0.000000	1.000000
25%	28.000000	1.178270e+05	9.000000	0.000000	0.000000	40.000000
50%	37.000000	1.783560e+05	10.000000	0.000000	0.000000	40.000000
75%	48.000000	2.370510e+05	12.000000	0.000000	0.000000	45.000000
max	90.000000	1.484705e+06	16.000000	99999.000000	4356.000000	99.000000

Fig. 1. Dados estatísticos obtidos no dataset de treino para as variáveis numéricas, nomeadamente valor mínimo e máximo

- Para a variável target (Salary_classification) o Dataset de treino encontra-se bastante desbalanceado. Como mostra a figura 2.

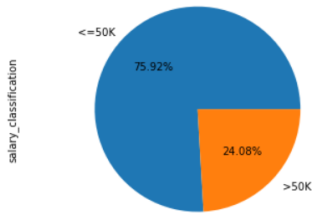

Fig. 2. O dataset de treino possui aproximadamente 76% de entradas que corresponde a indivíduos cujo o salário anual é inferior a 50,000$

– As variáveis, *native_country, workclass* e *occupation* encontram-se com valores em falta em (assinalado com "?").

```
test.isin([' ?']).sum(axis=0)

age                        0
workclass                963
fnlwgt                     0
education                  0
education_num              0
marital_status             0
occupation               966
relationship               0
race                       0
sex                        0
capital_gain               0
capital_loss               0
hours_per_week             0
native_country           274
salary_classification      0
dtype: int64
```

Fig. 3. Variáveis em falta no dataset de Teste.

```
data.isin([' ?']).sum(axis=0)

age                        0
workclass               1836
fnlwgt                     0
education                  0
education_num              0
marital_status             0
occupation              1843
relationship               0
race                       0
sex                        0
capital_gain               0
capital_loss               0
hours_per_week             0
native_country           583
salary_classification      0
dtype: int64
```

Fig. 4. Variáveis em falta no dataset de Treino.

Em relação ao processamento de dados, criamos variâncias do pré processamento de modo a procurar obter vários datasets e encontrar os modelos com os melhores valores possíveis.

O esquema 5, mostra com clareza os quatro datasets obtidos e o processamento de dados a que cada um foi sujeito.

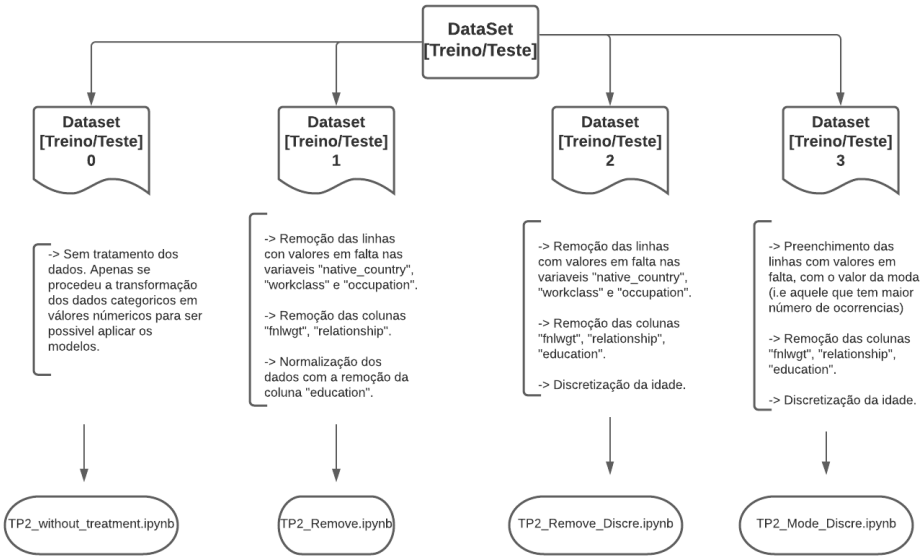

Fig. 5. Esquema representativo da limpeza a que o dataset foi sujeito.

Após análise do dataset inicial, verificou-se que o nome das variáveis se encontrava sem qualquer uniformidade contendo espaços no inicio que dificultavam o seu acesso. Utilizando a função replace, substitui-se os nomes das colunas retirando este espaço extra.

Para o Dataset 0, apenas se converteu as variáveis categóricas em numéricas (recorrendo a função map).

Para os restantes Datasets, a análise inicial dos dados foi semelhante. Tentou-se perceber quais as features que poderão apresentar uma maior relevância. Para isso, representamos graficamente as variáveis que considerados mais relevantes e analisamos e a sua relação com a variável target *salary_classification*. Além disso analisamos a correlação entre as features do dataset. Da análise dos dados anteriores, removemos nos datasets 1, 2 e 3, três features: *fnlwgt* (uma vez que corresponde a um dado estatísticos dos censos), *relationship* e *education* foi retirada por uma questão de normalização de dados pois a coluna *education-num* possui a mesma informação. As variáveis categóricas que sobraram, foram mapeadas para valores numéricos de modo a ser possível aplicar os modelos.

Para cada um dos dataset, estudou-se o seu comportamento recorrendo aos seguintes modelos:

COLOCAR EM CADA UM DELES UMA BREVE EXPLICAÇÃO DE COMO FORAM IMPLEMENTADOS

- **Regressão Logística;**
- **Naive Bayes;**

- **Decision Trees;**
- **K-Nearest Neighbour (KNN)**; Este modelo guarda todos os dados de treino para que, quando necessário, se executem comparações não necessitando de realizar aprendizagem.
- **Support Vector Machine (SVM);**

4 Construção e treino dos modelos preditivos

Nesta secção, discutimos os diferentes modelos utilizados e os resultados obtidos em cada um deles para os diferentes datasets. No enunciado deste projecto, foi sugerido que se use como métrica de avaliação da performance do modelo classificador a acurácia. Esta métrica indica, quanto o modelo acertou das previsões possíveis. Se um modelo tem uma acurácia de 70%, significa que acertou 7 das 10 previsões. Este valor é obtido, pelo calculo da razão entre o somatório das previsões corretas (verdadeiros positivos com verdadeiros negativos) sobre o somatório das previsões.

Para uma melhor avaliação das performances dos modelos, tivemos também em conta o valor do recall e precisão. O primeiro calcula quantos dos positivos reais o nosso modelo identifica, classificando-os como positivos verdadeiros. O segundo por sua vez, indica-nos quantos dos positivos previstos são efectivamente positivos verdadeiros.

Para o desenvolvimento dos modelos utilizou-se o ambiente de desenvolvimento Python/Sklearn.

4.1 DataSet 0

- **Regressão Logística**
  Aplicando este modelo ao dataset 0, que não teve qualquer tratamento de dados, observou-se uma accuracy de 80%, mas um elevado número de falsos negativos.

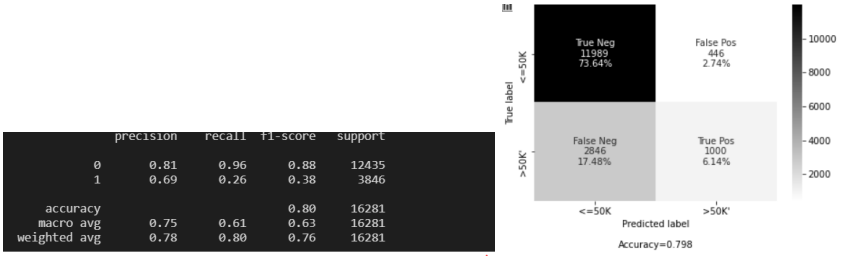

Fig. 6. Confusion Matrix e Classification report do modelo de regressão logística obtidas no dataset 0.

– **Naive Bayes**
Este modelo obteve uma acurácia de 80%.

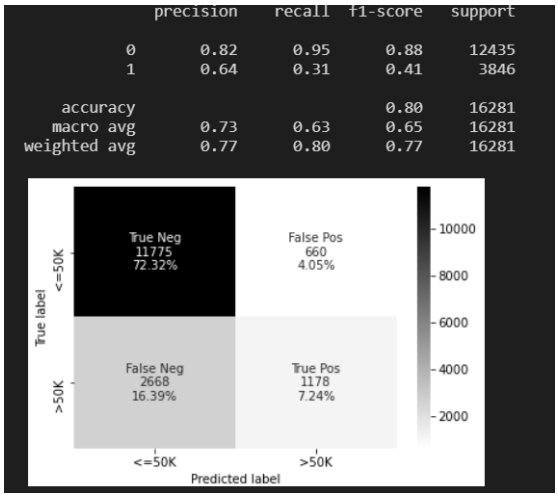

Fig. 7. Confusion Matrix e Classification report do modelo NaiveBayes obtidas no dataset 0.

– **K-Nearest Neighbour (KNN)**
Da análise da figura 8, conclui-se que o valor de K escolhido deve ser 20, visto que é o valor que nos garante melhores resultados sem overfitting.
Deste modo, obteve-se um accuracy de 80%, que não é um mau resultado, mas olhando para os resultados da confusion matrix podemos observar que o modelo não se está a comportar de acordo com o esperado apresentando um grande número de falsos negativos e falsos positivos

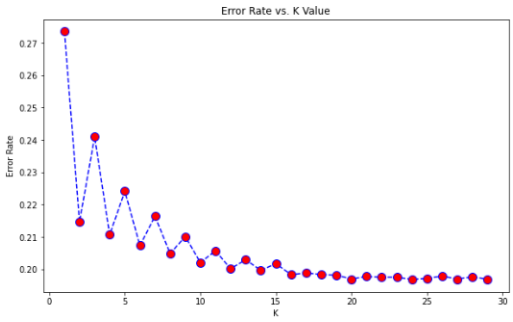

Fig. 8. Escolha do K no modelo KNN.

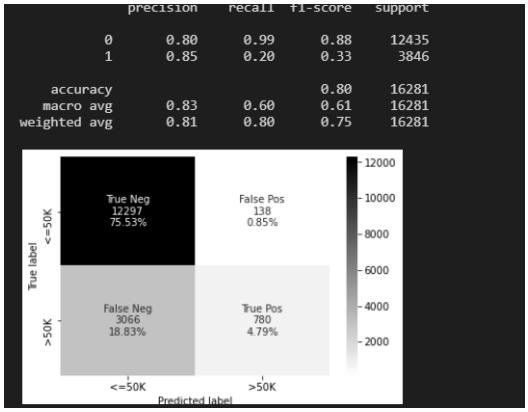

Fig. 9. Confusion Matrix e Classification report do modelo KNN, quando o k é 20, obtidas no dataset 0.

– **Support Vector Machine (SVM)**
A figura 10 apresenta os resultados obtidos com o modelo SVM. Que apresenta uma acuracia de 80%, a precisão também apresenta valores elevados o que significa que o modelo conseguiu prever quantos dos positivos previstos são efectivamente positivos verdadeiros. Alem disso, o modelos assinalou positivamente sempre que corresponde a um valor inferior a 50K, falhando 85% das vezes no outro caso. Isto mostra que o algoritmo aprendeu correctamente a prever os casos em que o salário é inferior a $50,000 mas não o conseguiu fazer no outro caso. Ou seja, aparenta estar overfitted e não tinha aprendido de todo.

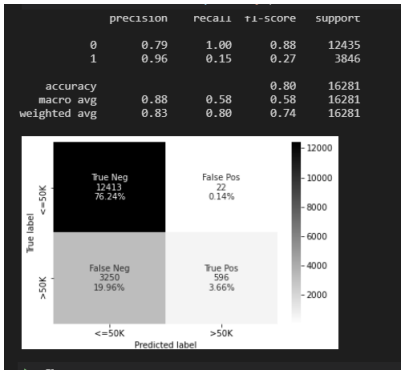

Fig. 10. Confusion Matrix e Classification report do modelo SVM obtidas no dataset 0.

– **Decision Trees**

Este algoritmo apresentou melhores resultados neste dataset. Conseguindo obter 84% de acurácia, valores de precisão de 76% e 86% e recall de 95% no caso do ganho inferior a 50k e 51% no outro caso.

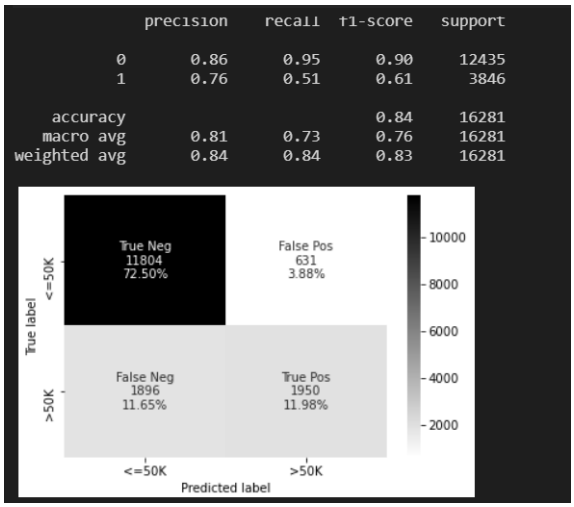

Fig. 11. Confusion Matrix e Classification report do modelo Decision Tree obtidas no dataset 0.

4.2 DataSet 1

- **Regressão Logística**
- **Naive Bayes**
- **Decision Trees**
- **K-Nearest Neighbour (KNN)**
- **Support Vector Machine (SVM**

4.3 DataSet 2

- **Regressão Logística**
- **Naive Bayes**
- **Decision Trees**
- **K-Nearest Neighbour (KNN)**
- **Support Vector Machine (SVM**

4.4 DataSet 3

- **Regressão Logística**

- **Naive Bayes**
- **Decision Trees**
- **K-Nearest Neighbour (KNN)**
- **Support Vector Machine (SVM**

5 Validação dos modelos

Tentar

para os FP e FN: tem mts mais caso de ¡=50k do que¿50k (dataset desbalanceado), o que isso faz é que o teu modelo fique mt bom para casos de ¡=50k mas fique mt maus para ¿50k. Por isso esse falsos positivos deve serem relacionados aos casos de ¿50k.

Para o melhor modelo, podemos aplicar tecnicas para corrigir esse desbalanceanto. oversampling em que pegas na parte pequena do dataset (neste caso ¿50k) e aumentas esse tamanho para ficar igual ao tamanho do ¡50k

ou entao, undersampling que é tornar a parte do ¡50k mais pequeno

mas existem funcoes em python que fazem esses algoritmos

tentamos retirar as tabelas capital gain e lost mas os valores obtidos pioraram.

6 Conclusão

References

1.