Departamento de Informática - IPV-ESTGV

Análise Inteligente de Dados

2020/2021

Trabalho Prático Final

Algoritmos Supervised Machine Learning para aprendizagem de modelos de classificação de dados

---

Autores:

Paulo Jorge Sousa
14743
estgv14743@alunos.estgv.ipv.pt

Hugo Loureiro
14660
estgv14660@alunos.estgv.ipv.pt

1 Objectivos do Trabalho Prático Final

O objectivo deste trabalho é desenvolver em Python, utilizando bibliotecas numpy, pandas e sklearn, um conjunto de programas que permita determinar a exactidão (**accuracy**) e a matriz de confusão (**confusion matrix**) de cada um dos modelos machine learning criados, utilizando os seguintes algoritmos:

- Naives Bayes
- Decision Tree
- Random Forest
- kNN

Estes algoritmos de aprendizagem devem ser aplicados a dois datasets, disponíveis no site da UCI, na construção de cada um dos modelos machine learning para classificação de dados:

- Census Income Data Set
- Congressional Voting Records Data Set

O programa deve garantir a preparação dos dados de forma a poderem ser utilizados na construção dos diferentes modelos pelos diferentes algoritmos. Verifique a evolução dos resultados quando utiliza diferentes percentagens do set de testes:

- 15%
- 30%
- 50%

2 Análise dos Data Set

Os valores de *accuracy* são extraídos da *confusion matrix*, tal como outras métricas de avaliação para classificação através das seguintes fórmulas apresentadas na seguinte imagem:

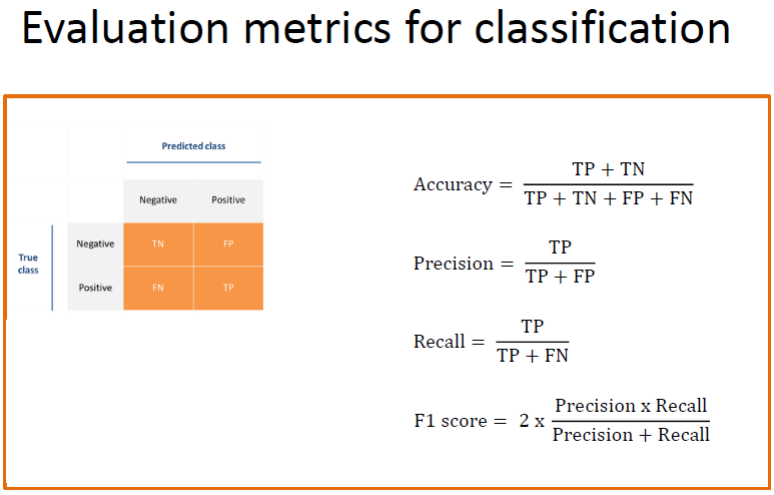

Figura 1: Evaluation metrics for Classification

Em muitos casos a ***Accuracy*** não é uma boa métrica da performance do nosso modelo, porque a distribuição das classes é desequilibrada, ou seja, uma classe é mais frequente que as outras.

Nestes casos, mesmo que prevamos todas as instâncias como a classe mais frequente, iríamos obter um valor bastante alto de *accuracy*. Isso significaria que o nosso modelo não estaria a aprender nada e que estaria apenas a prever para a classe mais frequente.

Então, precisamos de olhar para outra métrica específica de performance. E essa métrica será a ***Precision***.

Nos nossos data Sets em questão, a distribuição das classes é desequilibrada. então iremos também mostrar os valores de *precision*, para obtermos uma melhor análise dos dados dos nossos Data Sets.

De forma a obter a *precision*, críamos um excel para cada data set com todas as métricas de avaliação para classificação (accuracy, precision, recall e F1 score). Nestes ficheiros excel foram aplicadas as fórmulas da imagem acima de forma a confirmar o valor da accuracy e as restantes métricas, através das *confusion matrix* devolvidas pelos programas desenvolvidos. Os ficheiros excel encontram-se nas pastas com o nome de cada data set e com o seguinte nome: "**Evaluation metrics for classification - house votes-84**"para o Congressional Voting Records Data Set e "**Evaluation metrics for classification - adult-data**"para o Census Income Data Set.

As seguintes secções apresentam os valores de *accuracy* e *precision* dos nossos Data Sets, dependendo da percentagem do set de testes, combinação de encoderes e algoritmo de machine learning aplicado.

3 Congressional Voting Records Data Set

3.1 Relevant Information about the Data Set

This data set includes votes for each of the U.S. House of Representatives Congressmen on the 16 key votes identified by the CQA. The CQA lists nine different types of votes: voted for, paired for, and announced for (these three simplified to yea), voted against, paired against, and announced against (these three simplified to nay), voted present, voted present to avoid conflict of interest, and did not vote or otherwise make a position known (these three simplified to an unknown disposition).

NOTE: It is important to recognize that "?"in this database does not mean that the value of the attribute is unknown. It means simply, that the value is not "yea"or "nay"(see "Relevant Information"section above).

3.1.1 Attribute information

1. Class Name: 2 (democrat, republican)
2. handicapped-infants: 2 (y,n)
3. water-project-cost-sharing: 2 (y,n)
4. adoption-of-the-budget-resolution: 2 (y,n)
5. physician-fee-freeze: 2 (y,n)
6. el-salvador-aid: 2 (y,n)
7. religious-groups-in-schools: 2 (y,n)
8. anti-satellite-test-ban: 2 (y,n)
9. aid-to-nicaraguan-contras: 2 (y,n)
10. mx-missile: 2 (y,n)
11. immigration: 2 (y,n)
12. synfuels-corporation-cutback: 2 (y,n)
13. education-spending: 2 (y,n)
14. superfund-right-to-sue: 2 (y,n)
15. crime: 2 (y,n)
16. duty-free-exports: 2 (y,n)
17. export-administration-act-south-africa: 2 (y,n)

3.2 Programas em python

Os programas que desenvolvemos garantem a preparação dos dados de forma a poderem ser utilizados na construção dos diferentes modelos pelos diferentes algoritmos.

Neste caso, desenvolvemos 12 programas nos quais diferem a percentagem do set de testes e os encoders utilizados. Os programas devolvem-nos a ***accurracy*** e a ***Confusion Matrix***.

3.2.1 Accuracy

Os resultados de ***accuracy*** extraídos dos programas são os seguintes:

Accuracy - 15% test do set de testes	**Naive Bayes**	**Decision Tree**	**Random Forest**	**kNN**
Label Encoder	0.878787879	0.954545455	0.954545455	0.863636364
Label Encoder + Standard Scaler	0.893939394	0.954545455	0.954545455	0.878787879
Label Encoder + OneHotEncoder	0.954545455	0.954545455	0.954545455	0.909090909
Label Encoder + OneHotEncoder + Standard Scaler	0.954545455	0.954545455	0.954545455	0.893939394

Tabela 1: Accuracy values - 15%

Accuracy - 30% test do set de testes	**Naive Bayes**	**Decision Tree**	**Random Forest**	**kNN**
Label Encoder	0.900763359	0.954198473	0.954198473	0.885496183
Label Encoder + Standard Scaler	0.900763359	0.954198473	0.954198473	0.885496183
Label Encoder + OneHotEncoder	0.946564885	0.954198473	0.954198473	0.908396947
Label Encoder + OneHotEncoder + Standard Scaler	0.679389313	0.954198473	0.954198473	0.900763359

Tabela 2: Accuracy values - 30%

Accuracy - 50% test do set de testes	**Naive Bayes**	**Decision Tree**	**Random Forest**	**kNN**
Label Encoder	0.917431193	0.944954128	0.949541284	0.922018349
Label Encoder + Standard Scaler	0.922018349	0.944954128	0.949541284	0.922018349
Label Encoder + OneHotEncoder	0.963302752	0.963302752	0.958715596	0.926605505
Label Encoder + OneHotEncoder + Standard Scaler	0.614678899	0.963302752	0.958715596	0.922018349

Tabela 3: Accuracy values - 50%

3.2.2 Precision

Os valores de ***Precision*** são os seguintes:

Precision - 15% test do set de testes	**Naive Bayes**	**Decision Tree**	**Random Forest**	**kNN**
Label Encoder	0.75	0.904761905	0.904761905	0.72
Label Encoder + Standard Scaler	0.76	0.904761905	0.904761905	0.730769231
Label Encoder + OneHotEncoder	0.904761905	0.904761905	0.904761905	0.769230769
Label Encoder + OneHotEncoder + Standard Scaler	0.9047619055	0.904761905	0.904761905	0.740740741

Tabela 4: Precision values - 15%

Precision - 30% test do set de testes	**Naive Bayes**	**Decision Tree**	**Random Forest**	**kNN**
Label Encoder	0.795918367	0.928571429	0.928571429	0.775510204
Label Encoder + Standard Scaler	0.928571429	0.954198473	0.928571429	0.764705882
Label Encoder + OneHotEncoder	0.906976744	0.928571429	0.928571429	0.788461538
Label Encoder + OneHotEncoder + Standard Scaler	!!!	0.928571429	0.928571429	0.773584906

Tabela 5: Precision values - 30%

Precision - 50% test do set de testes	**Naive Bayes**	**Decision Tree**	**Random Forest**	**kNN**
Label Encoder	0.858695652	0.93902439	0.939759036	0.852631579
Label Encoder + Standard Scaler	0.868131868	0.93902439	0.939759036	0.852631579
Label Encoder + OneHotEncoder	0.941860465	0.963414634	0.941176471	0.854166667
Label Encoder + OneHotEncoder + Standard Scaler	!!!	0.963414634	0.941176471	0.852631579

Tabela 6: Precision values - 50%

Os valores das tabelas acima que estão marcados com !!! significa que não havia instâncias daquela classe com aquela percentagem do set para testes.

4 Census Income Data Set

4.1 Relevant Information about the Data Set

Este dataset foi extraido pelo Barry Becker do banco de dados do Censo de 1994, este conjunto de registo foi extraido seguindo determinadas condições ((AAGE>16) (AGI>100) (AFNLWGT>1) (HRSWK>0)), na tentativa de previsão se uma determinada pessoa ganha mais de 50.000 por ano.

4.1.1 Attribute information

1. age: continuous.
2. workclass: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
3. fnlwgt: continuous.
4. 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.
5. education-num: continuous.
6. marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
7. 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.
8. relationship: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
9. race: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
10. sex: Female, Male.
11. capital-gain: continuous.
12. capital-loss: continuous.
13. hours-per-week: continuous.
14. 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.
15. class: >50K, <=50K

4.2 Programas em python

Os programas que desenvolvemos garantem a preparação dos dados de forma a poderem ser utilizados na construção dos diferentes modelos pelos diferentes algoritmos.

Neste caso, desenvolvemos 12 programas nos quais diferem a percentagem do set de testes e os encoders utilizados. Os programas devolvem-nos a ***accurracy*** e a ***Confusion Matrix***.

4.2.1 Accuracy

Os resultados de ***accuracy*** extraídos dos programas são os seguintes:

Accuracy - 15% test do set de testes	**Naive Bayes**	**Decision Tree**	**Random Forest**	**kNN**
Label Encoder	0.82272262	0.813101331	0.843193449	0.777686796
Label Encoder + Standard Scaler	0.824360287	0.80962129	0.839713408	0.828659161
Label Encoder + OneHotEncoder	0.802047083	0.811873081	0.848515865	0.776663255
Label Encoder + OneHotEncoder + Standard Scaler	0.755987718	0.811054248	0.845035824	0.829273286

Tabela 7: Accuracy values - 15%

Accuracy - 30% test do set de testes	**Naive Bayes**	**Decision Tree**	**Random Forest**	**kNN**
Label Encoder	0.823421026	0.815641314	0.852492579	0.776230935
Label Encoder + Standard Scaler	0.825468318	0.808885249	0.849728734	0.835295322
Label Encoder + OneHotEncoder	0.805404852	0.816050773	0.849831098	0.775923841
Label Encoder + OneHotEncoder + Standard Scaler	0.758214761	0.81001126	0.847886171	0.834066946

Tabela 8: Accuracy values - 30%

Accuracy - 50% test do set de testes	**Naive Bayes**	**Decision Tree**	**Random Forest**	**kNN**
Label Encoder	0.81985136	0.812972176	0.850254898	0.772434126
Label Encoder + Standard Scaler	0.821816842	0.804864566	0.844174191	0.834653891
Label Encoder + OneHotEncoder	0.801854923	0.813524968	0.847183834	0.773048339
Label Encoder + OneHotEncoder + Standard Scaler	0.75689454	0.804127511	0.843130029	0.831337141

Tabela 9: Accuracy values - 50%

4.2.2 Precision

Os valores de ***Precision*** são os seguintes:

Precision - 15% test do set de testes	**Naive Bayes**	**Decision Tree**	**Random Forest**	**kNN**
Label Encoder	0.707379135	0.617721519	0.729525862	0.570478723
Label Encoder + Standard Scaler	0.709273183	0.609899329	0.718716578	0.660924751
Label Encoder + OneHotEncoder	0.631887456	0.615189873	0.740425532	0.567065073
Label Encoder + OneHotEncoder + Standard Scaler	!!!	0.612552301	0.72967265	0.665434381

Tabela 10: Precision values - 15%

Precision - 30% test do set de testes	**Naive Bayes**	**Decision Tree**	**Random Forest**	**kNN**
Label Encoder	0.699937225	0.619412516	0.74851592	0.557971014
Label Encoder + Standard Scaler	0.698729583	0.607468519	0.743199129	0.670903314
Label Encoder + OneHotEncoder	0.634245778	0.619450317	0.744401966	0.55687048
Label Encoder + OneHotEncoder + Standard Scaler	!!!	0.609051724	0.740924092	0.669876204

Tabela 11: Precision values - 30%

Precision - 50% test do set de testes	**Naive Bayes**	**Decision Tree**	**Random Forest**	**kNN**
Label Encoder	0.696888206	0.615016377	0.746433204	0.554129226
Label Encoder + Standard Scaler	0.696249536	0.602359109	0.737704918	0.676224944
Label Encoder + OneHotEncoder	0.627526132	0.616767984	0.744836775	0.556413556
Label Encoder + OneHotEncoder + Standard Scaler	!!!	0.600260756	0.739590444	0.668426904

Tabela 12: Precision values - 50%

Os valores das tabelas acima que estão marcados com !!! significa que não havia instâncias daquela classe com aquela percentagem do set para testes.

5 Conclusões

Como podemos ver com os dados acima representados e os dados dos ficheiros excel disponibilizados para cada data set, podemos concluir que os nossos modelos têm maior precisão (***precision***) a prever para a classe positiva (classe com mais instâncias) do que para a classe negativa (classe com menos instâncias).

- No **Census Income Data Set** temos como classe positiva «=**<=50K**", pois tem 24721 instâncias. Sendo que a classe negativa «=**>50K**"tem 7841 instâncias.
- No **Congressional Voting Records Data Set** temos como classe positiva «=**Democrats**", pois tem 267 instâncias. Sendo que a classe negativa, «=**Republicans**", tem 7841 instâncias.