Assigment: Model Evaluation

DSA 8401: Applied Machine Learning

Master in Data Science and Analytics

Strathmore University

@iLabAfrica Centre

1 Goals

This week's activity focuses on practicing with metrics and model evaluation techniques. The specific objectives are:

- Analyze the different metrics
- Study the problem of imbalanced data
- Practice how to evaluate models

2 Assignment Description

In this activity, we will use the classic diabetes dataset (dataset description) and the same dataset but unbalanced (we have removed records with diagnosis 1) which you will find in the virtual classroom.

Optional: Use a different dataset in your expertise domain. Students can find a nice collection of datasets on the following page Dataset List.

Exercise 1. Steps to Evaluate a Model

Students will write a Jupyter notebook coding the following steps. The libraries needed for this assignment are

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import auc
from sklearn.metrics import roc_curve
from sklearn.metrics import roc_auc_score
from sklearn.metrics import confusion_matrix
```

- Upload to Pandas a local file that we have previously downloaded from the virtual classroom or optionally from the UCIrvine repository.

```
diabetesDF = pd.read_csv("/path_to_folder/diabetes.csv")
print(diabetesDF.head())
```

- Observe the dataset attributes and their statistical description.
- Split the dataset into training and test.
- Taking into account that the algorithm that we will use to classify diabetes cases will be logistic regression, two steps must be carried out. On the one hand, transform the data into numpy arrays and, on the other hand, normalize this data to improve the performance of the algorithm. Note also that we take the opportunity to differentiate the attributes of the dataframe in input data of the algorithm (trainData and testData - it would be the x used in the algorithm) and the value used to indicate whether the patient has diabetes or not (trainLabel and testLabel - would be the y):

```
trainLabel = np.asarray(dfTrain["Outcome"])
trainData = np.asarray(dfTrain.iloc[:,0:8])
testLabel = np.asarray(dfTest["Outcome"])
testData = np.asarray(dfTest.iloc[:,0:8])
means = np.mean(trainData, axis=0)
stds = np.std(trainData, axis=0)
trainData = (trainData - means)/stds
testData = (testData - means)/stds
```

- Create the model

```
model = LogisticRegression()
model.fit(trainData, trainLabel)
```

- Finally, evaluate the model with the test dataset. Display the confusion matrix and different metrics. Here you get the accuracy

```
accuracy = model.score(testData, testLabel)
print("accuracy = ", accuracy * 100, "%")
```

Question 1: Analyze the data in the dataframe and briefly present your most relevant conclusions.
Question 2: In which cases is it appropriate to use less than 20% of the data, and why?
Question 3: Check that the input data, trainData and testData, are normalized, indicating the Python code used for this. Why shouldn't we normalize the trainLabel and testLabel sets?
Question 4: Once the model has been evaluated, can you comment on what you think of the metrics used and the result obtained taking into account the nature of the dataframe considered?

Exercise 2. Metrics

In this second exercise, we will aim to correctly interpret the different metrics used to evaluate a model. First, we will show how to represent the ROC Curve for the case of logistic regression in Python. Recall that the ROC curve is the result of varying the detection threshold (classification). In our case, since the "Outcome" attribute is equal to 1 for a patient with a positive diabetes diagnosis, increasing the detection threshold means reducing both the detection probability and the false alarm probability.
Question 5: Comparing the ROC curve obtained with the random case (diagonal), what can you say about the selected classifier? Observing the trend of the curve, what type of threshold would you use (high or low)?
Question 6: Fill in the following table for different probability threshold values.

Treshold	Accuaracy	Sensitivity	Recall	F1	TPR	FPR
0.0						
0.2						
0.4						
0.6						
0.8						
1.0						

Question 7: What do you observe in the extreme cases when the threshold is 0 or 1?
Question 8: What do the metrics of Sensitivity and Recall contribute compared to Accuracy?
Question 9: If the diabetes diagnosis study were to be used to select individuals for a general screening through laboratory tests to find diabetics, which probability threshold value would you choose? Justify your answer.

Exercise 3. Imbalanced data

In the previous exercises, you might have observed that the diabetes dataset is imbalanced, and this has some impact on the evaluation metrics. Now we will use an even more imbalanced dataset. Specifically, you should consider the dataset diabetesDes.csv that you have in the virtual classroom. You are asked to follow the same steps as in Exercises 1 and 2 for this case.
Question 10: Comment on and explain the results obtained regarding the metrics observed when constructing the following table.

Optional exercise. Cross validation k-fold

Repeat the exercises 1 and 2 this time using cross validation.
Question 11: Analyze the behavior (mean and variance of the accuracy) of the different folds obtained for k values of 5, 10, and 15. What conclusions do you draw? Which value would you use to select hyperparameters or to estimate the performance of the classifier?

3 Hand in

The students will upload in the eLearning platform the Jupyter notebook. The notebook will contain the python code, the plots and especially personal responses to the previous questions. If there is no personal comment the mark will be F. Marks C,B and A will depend on the variety and quality of the personal comments.