ISEP
IG.3510 - Machine Learning
September 23rd 2024

Tutorial course 2 : Supervised Learning : Classification
Part I

1 Part I : Exercises

Exercise 1. The table below provides a training data set containing six observations, three predictors, and one qualitative response variable.

Obs.	X_1	X_2	X_3	Y
1	0	3	0	Red
2	2	0	0	Red
3	0	1	3	Red
4	0	1	2	Green
5	-1	0	1	Green
6	1	1	1	Red

Suppose we wish to use this data set to make a prediction for Y when $X_1 = X_2 = X_3 = 0$ using K-nearest neighbors.

(a) Compute the Euclidean distance between each observation and the test point, $X_1 = X_2 = X_3 = 0$.

(b) What is our prediction with $K = 1$? Why ?

(c) What is our prediction with $K = 3$? Why ?

(d) If the Bayes decision boundary in this problem is highly non- linear, then would we expect the best value for K to be large or small ? Why ?

Exercise 2. Suppose we collect data for a group of students in a statistics class with variables $X_1 =$ hours studied, $X_2 =$ undergrad GPA, and $Y =$ receive an A. We fit a logistic regression and produce estimated coefficients, $\hat{\beta}_0 = -6$, $\hat{\beta}_1 = 0.05$ and $\hat{\beta}_2 = 1$.

(a) Estimate the probability that a student who studies for 40 h and has an undergrad GPA of 3.5 gets an A in the class.

(b) How many hours would the student in part (a) need to study to have a 50% chance of getting an A in the class ?

Exercise 3. Suppose that we take a data set, divide it into equally-sized training and test sets, and then try out two different classification procedures. First we use logistic regression and get an error rate of 20% on the training data and 30% on the test data. Next we use 1-nearest neighbors (i.e. $K = 1$) and get an average error rate (averaged over both test and training data sets) of 18%. Based on these results, which method should we prefer to use for classification of new observations ? Why ?

Exercise 4. This exercise is about odds.

(a) On average, what fraction of people with an odds of 0.37 of defaulting on their credit card payment will in fact default ?

(b) Suppose that an individual has a 16% chance of defaulting on her credit card payment. What are the odds that she will default ?

Exercise 5. Answer the following questions concerning differences between *LDA* and *QDA* :

(a) If the Bayes decision boundary is linear, do we expect *LDA* or *QDA* to perform better on the training set ? On the test set ?

(b) If the Bayes decision boundary is non-linear, do we expect *LDA* or *QDA* to perform better on the training set ? On the test set ?

(c) In general, as the sample size n increases, do we expect the test prediction accuracy of *QDA* to improve, decline, or be unchanged ? Why ?

(d) True or False : Even if the Bayes decision boundary for a given problem is linear, we will probably achieve a superior test error rate using *QDA* rather than *LDA* because *QDA* is flexible enough to model a linear decision boundary. Justify your answer.

Exercise 6. Suppose that we wish to predict whether a given stock will issue a dividend this year (*Yes* or *No*) based on X, last year's percent profit. We examine a large number of companies and discover that the mean value of X for companies that issued a dividend was 10, while the mean for those that didn't was 0. In addition, the variance of X for these two sets of companies was $\hat{\sigma}^2 = 36$. Finally, 80% of companies issued dividends. Assuming that X follows a normal distribution, predict the probability that a company will issue a dividend this year given that its percentage profit was $X = 4$ last year.
Hint : Recall that the density function for a normal random variable with parameters μ *and* σ^2 *is* $f(x) = \frac{1}{\sqrt{2\pi\sigma}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$. *You will need to use Bayes' theorem.*

2 Part II : Practical application graded !

2.1 The data set description

In this exercise you will use the *bankrupt* data set. The data was collected from the Taiwan Economic Journal from 1999 to 2009. Company bankruptcy was defined based on the business regulations of the Taiwan Stock Exchange. The purpose is to build a classifier to predict whether a given company goes bankrupt or not. Follow the instructions :

1. Use the function `read_csv()` from the *Pandas* library to read the file *bankrupt.txt*.

```
import pandas
bankrupt = pandas.read_csv("bankrupt.txt",sep=",")
```

2. [*graded question*] How many observations and variables are there ? Use the property `shape` of the dataframe.

3. [*graded question*] Use the command `bankrupt.describe()` to calculate descriptive statistics for all the variables. Comment on the results. Do you think it is necessary to standardize the variables before performing classification ?

4. [*graded question*] Describe the target variable. What is the number and percentage of companies that went bankrupt ? Is the data set balanced ? You can use the command `value_counts()`.

5. You will fit the classifiers studied in the lecture using the *train* data and evaluate the quality of your models using the *test* data. That is why the whole data set was split into two subsets. To this end, you will find in moodle 4 files `x_train.csv`, `x_test.csv`, `y_train.csv` and `y_test.csv`. Only 12 features were selected from the original data set. The features were renamed according

Orinal name	new name
ROA(C) before interest and depreciation before interest	ROAC
ROA(A) before interest and % after tax	ROAA
ROA(B) before interest and depreciation after tax	ROAB
Tax rate (A)	TRA
Total Asset Growth Rate	TAGR
Debt ratio %	DR
Working Capital to Total Assets	WKTA
Cash/Total Assets	CTA
Current Liability to Assets	CLA
CFO to Assets	CFOA
Current Liability to Current Assets	CLCA
Net Income to Total Assets	NITA
Bankrupt ?	Bankrupt

to the following table.

Import the four data sets.
Warning : The first column contains the row labels. You will have to use `read_csv()` function with the `index_col` parameter set to 0 to import the data properly.

[graded question] How many observations are in each data set ? Is the distribution of the classes of the target variable similar in both data sets ?

In order to have the same data set for all the classification methods, you are going to standardize the data set. You will need to use `StandardScaler()` function from the `preprocessing` library of `sklearn`. Create a new dataframe of the standardized variables contained in `x_train` and in `x_test`. Verify that the new dataframes have the same length as the original ones.

2.2 Logistic regression

In order to fit logistic regression, you will need the function `glm()` which is part of the `formula` submodule belonging to the `statsmodels` module, a Python module that provides classes and functions for the estimation of many different statistical models.

```
import statsmodels.api as sm
import statsmodels.formula.api as smf
```

1. *[graded question]* Perform logistic regression with `Bankrupt` as the response and all the variables in the x_train data set as features. You will need to set the argument as follows `family=sm.families.Binomial()` to tell `Python` to run logistic regression. Previously, do not forget to merge the `x_train` and the `y_train` dataframes, to this end you can use the function `concat()` of the *Pandas* library as follows :
   `bankrupt_train=pandas.concat([x_train,y_train],axis=1)`.

   ```
   formula = 'Bankrupt ~ ROAC + ROAA +.... complete the formula'
   model = smf.glm(formula = formula, data=bankrupt_train, family=sm.families.Binomial())
   logreg = model.fit()
   ```

Use the summary() method to print the results, print(logreg.summary()).

Do any of the predictors appear to be statistically significant ? If so, which ones ? Interpret the coefficient estimates of the significant features.

2. [*graded question*] The command print(logreg.fittedvalues) allows to print the estimated probability $\hat{p}(Y = Bankrupt|\mathbf{X})$ (where $\mathbf{X}$ is the set of explanatory variables). That is, the probability for the company to go bankrupt given $\mathbf{X}$.
   In order to predict the target variable, you can convert these probabilities into classes labels as follows :

   ```
   yhat_logreg_probs = logreg.fittedvalues
   yhat = [1 if x > 0.5 else 0 for x in yhat_logreg_probs]
   ```

   where yhat_logreg_probs is the vector of estimated probabilities and yhat is a list containing the predicted values $\hat{y}$.

   You can compute the confusion matrix and obtain a report of performance metrics in order to assess the quality of fit of the classifier by using the confusion_matrix(), classification_report functions from the sklearn.metrics library. For instance, to calculate the confusion matrix between the vector of predicted values yhat and the real values y_train run[1] :

   ```
   from sklearn.metrics import confusion_matrix, classification_report
   print(confusion_matrix(yhat, y_train))
   print(classification_report(yhat,y_train,digits=3))
   ```

   Compute the confusion matrix and the classification report for your logistic regression model. Interpret the metrics.

3. [*graded question*] The performance indicators calculated in the previous question do not give any idea about how the model will perform in unknown data as they are calculated using the same data that was used to fit the model. Calculate the performance measures described in the previous question using the test set. You will have to use the predict() function in order to predict the target variable for the test set. For instance, in order to obtain the predicted probabilities run the command logreg.predict(x_test). Interpret the results. Do you think is it appropriate to consider mainly the overall accuracy in an imbalanced data set ? If not, which metrics are more relevant ?

2.3 K-Nearest Neighbors

You can build a K-Nearest Neighbors model using the KNeighborsClassifier() function, which is part of the neighbors submodule of (sklearn). For instance, to train KNN with $K = 1$ classifier in training sets y_train and x_train and then make predictions with the test sets y_test and x_test run the following commands :

```
from sklearn.neighbors import KNeighborsClassifier
import numpy as np
neigh = KNeighborsClassifier(n_neighbors=1)
neigh.fit(x_train,np.ravel(y_train))
yhat_knn=neigh.predict(x_test)
```

[1] According to the sklearn help, the first input to give to the function confusion_matrix() is the vector of real values and the second one is the vector of predictions. However, in this lab we invert the order in order to follow what is given in the lecture, that is, the estimates in rows and the ground-truth in columns.

1. [*graded question*] Fit a *KNN* classifier for the following values of K ranging from 1 to 20 and choose the model for which the the balanced accuracy (arithmetic mean of the recall per class.) for the test set is the highest. You will need to import the `balanced_accuracy_score()` function from the `sklearn.metrics` library. What value of K would you choose ? For this value of K calculate the performance indicators, confusion matrix and classification report, on the test set. Interpret the results.

2.4 Discriminant Analysis

The `LinearDiscriminantAnalysis()` and the `QuadraticDiscriminantAnalysis()` functions from the class `sklearn.discriminant_analysis` allow to perform Linear and Quadratic discriminant Analysis respectively. For instance, to perform LDA on the train set and predict the target variable on the test set you can run the following code.

```
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
lda = LinearDiscriminantAnalysis()
model_lda = lda.fit(x_train,np.ravel(y_train))
yhat_lda=model_lda.predict(x_test)
```

In addition, the commands :

```
print(model_lda.priors_)
print(model_lda.means_)
```

allow to know the prior probabilities, $\hat{\pi}_0$ and $\hat{\pi}_1$, and the group means for each predictor in the order they were given.

1. [*graded question*] Interpret the prior probabilities as well as the group means.
2. [*graded question*] Calculate the confusion matrix and the classification report on the test set for the model obtained with the *LDA* classifier. Discuss about the obtained metrics (specificity, sensitivity,precision, f1-score, etc.).
3. [*graded question*] Perform *QDA* on the training data. Calculate the confusion matrix and the classification report on the test set for the *QDA* classifier.

2.5 ROC (Receiver operating characteristic) curve

The functions `roc_curve()` and `auc()` from the `sklearn.metrics` class allow to build a ROC curve and calculate the `auc` (aire under the curve). Previously you will need to compute the estimated probabilities $\hat{p}(Y = 1 l X)$. For instance, for the LDA model you can obtain them with the following code :

```
lda_scores = lda.predict_proba(x_test)[:,1]
```

1. [*graded question*] Now, you can run the following code to perform preliminary calculations to plot de ROC curve :

   ```
   from sklearn.metrics import roc_curve, auc
   fpr, tpr, thresholds = roc_curve(y_test, lda_scores)
   aire = auc(fpr, tpr)
   ```

   In the above code, interpret the outputs of the `roc_curve()` function, namely `fpr` and `tpr`.
2. Finally, to plot the ROC curve use the `plt()` function from the `matplotlib.pyplot` class.

```
import matplotlib.pyplot as plt
plt.figure()
plt.plot(fpr, tpr, color='orange', lw=2, label='ROC curve (auc = %0.3f)' % aire)
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='—')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC curve for LDA classifier')
plt.legend(loc="lower right")
plt.show()
```

[graded question] Calculate the `auc` and plot the ROC curve for the 4 models LDA, QDA, logistic regression and *KNN* with the chosen value of K. According to the ROC curve which model would you choose ?

3. *[graded question]* By looking at the ROC curve of each model, which value of classification threshold would you choose to deal with this imbalanced data set and get a good compromise for the two classes ? Conclude about the advantages and disadvantages of dealing with this imbalanced data set.

3 References

— UCI Machine Learning Repository (2020) "*Taiwanese Bankruptcy Prediction*". `https://doi.org/10.24432/C5004D`. Consulted on September 15th, 2023.

— James, Gareth ; Witten, Daniela ; Hastie, Trevor and Tibshirani, Robert. "An Introduction to Statistical Learning with Applications in R", 2nd edition, New York : "Springer texts in statistics", 2021. Site web : `https://hastie.su.domains/ISLR2/ISLRv2_website.pdf`.

— Hastie, Trevor ; Tibshirani, Robert and Friedman, Jerome (2009). "The Elements of Statistical Learning (Data Mining, Inference, and Prediction), 2nd edition". New York : "Springer texts in statistics". Site web : `http://statweb.stanford.edu/~tibs/ElemStatLearn/`