



دانشکده مهندسی کامپیوتر و فناوری اطلاعات

Kasra Khalafi: 9531306

Final Project

Data Mining

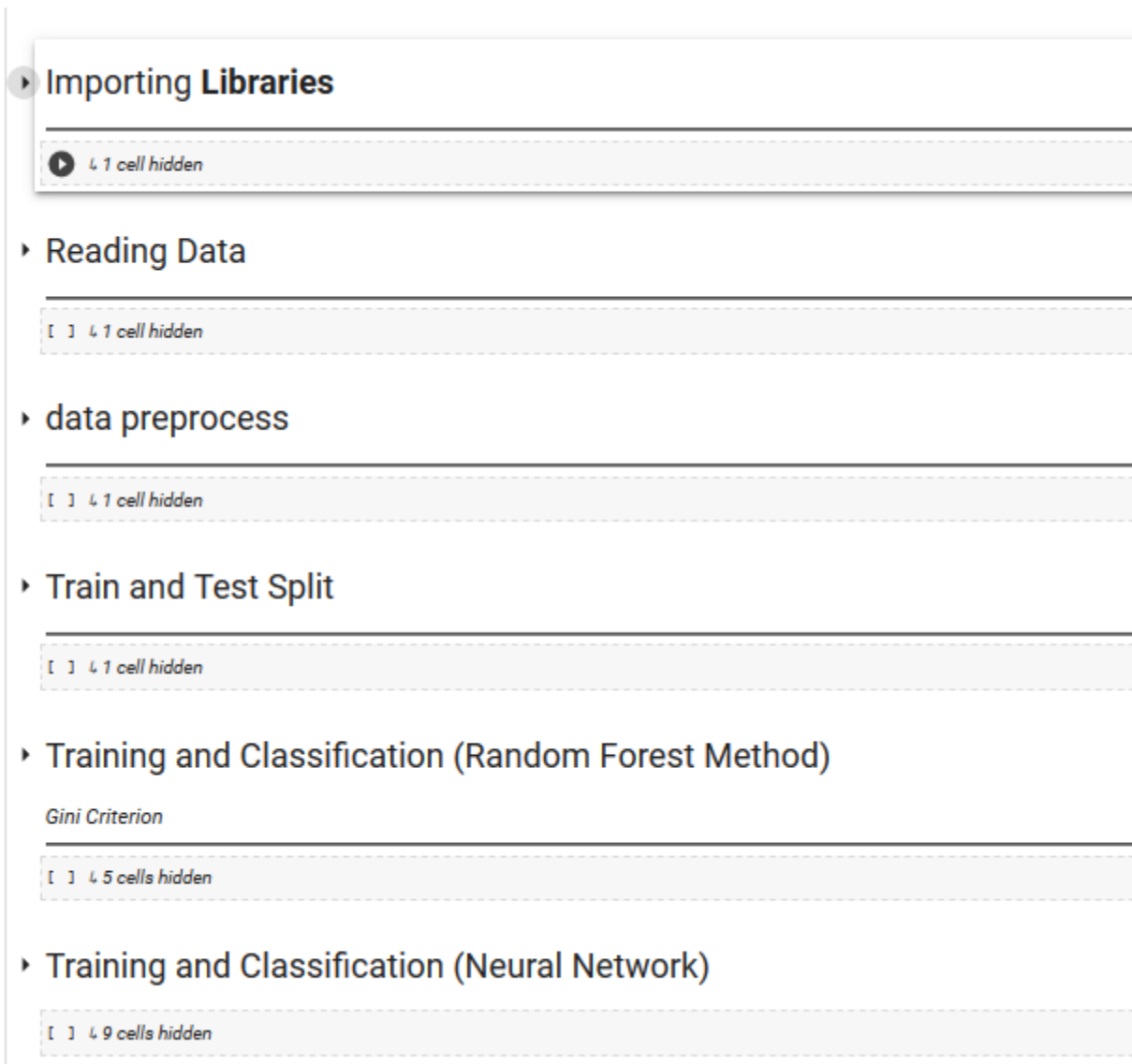
Dr.Ehsan Nazerfard

Winter Term 2019

In final project, there was a data from hotel with different features (32 features) and there existed about 120.000 data and rows.

For processing data and also checking cancelation 2 method as mentioned in question is used. First one is one of the ensemble methods, Random Forest. The second method is Neural Network (NN) which was implemented with help of Keras.

As an overlook, steps are divided as below:



Firstly, needed libraries were added then Data was read by Pandas library. This input data was preprocessed.

For preprocessing part, first of all, all data types in each column were changed to numbers so that we are working now with numbers instead of combination of numbers and strings. Also date was not

separated and it was all considered as one data, which is not correct because it contains year, month and day which could be useful information in training part.

In Train and Test Split, train and test parts were separated from each other by shuffling input data to inhibit any further problems in training and to cease overfitting. 20 percent of data was chosen for test and the remaining 80 percent was selected for training, because test is just for making sure that our model works fine or not and main part of our endeavor is happening for training part.

Random Forest:

With help of SKLearn, random forest was implemented as code attached.

First, criterion was chosen as gini. F1-score obtained is 99 percent.

Other details are illustrated below:

```
▶ clf = RandomForestClassifier(criterion='gini',max_depth=2)
   clf = clf.fit(x_train, y_train)
```

```
[98] yPredict = clf.predict(x_test)
      print(classification_report(y_test, yPredict))
```

	precision	recall	f1-score	support
0	0.98	1.00	0.99	15057
1	1.00	0.97	0.99	8821
accuracy			0.99	23878
macro avg	0.99	0.99	0.99	23878
weighted avg	0.99	0.99	0.99	23878

First, criterion was choosed as entropy. F1-score obtained is 98 percent.

Other details are illustrated below:

```
[100] clf = RandomForestClassifier(criterion='entropy',max_depth=2)
      clf = clf.fit(x_train, y_train)
```

```
▶ yPredict = clf.predict(x_test)
  print(classification_report(y_test, yPredict))
```

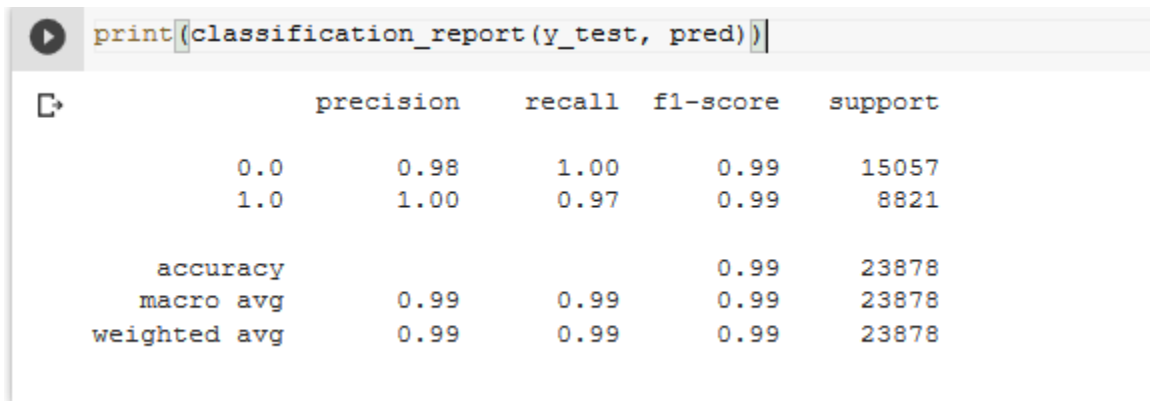
	precision	recall	f1-score	support
0	0.97	1.00	0.99	15057
1	1.00	0.95	0.98	8821
accuracy			0.98	23878
macro avg	0.99	0.98	0.98	23878
weighted avg	0.98	0.98	0.98	23878

Neural Network:

With help of Keras, a 3 layer neural network was implemented as code attached.

F1-score obtained is 99 percent.

Other details are illustrated below:



The image shows a Jupyter Notebook cell with a code icon on the left and a code editor containing the command `print(classification_report(y_test, pred))`. Below the code editor, the output of the command is displayed as a table. The table has five columns: 'precision', 'recall', 'f1-score', and 'support'. The first two rows show results for classes 0.0 and 1.0. The next three rows show aggregated metrics: 'accuracy', 'macro avg', and 'weighted avg'.

	precision	recall	f1-score	support
0.0	0.98	1.00	0.99	15057
1.0	1.00	0.97	0.99	8821
accuracy			0.99	23878
macro avg	0.99	0.99	0.99	23878
weighted avg	0.99	0.99	0.99	23878