

CS 5720 Neural Network Deep Learning

ICP-5

CRN	23441
NAME	MANOJ BALA
EMAIL	mxb40210@ucmo.edu
STUDENT_ID	700754021

GitHub Repository:

<https://github.com/mxb40210/700754021-NeuralNetworkDeepLearning>

Assignment 4:

<https://github.com/mxb40210/700754021-NeuralNetworkDeepLearning/tree/main/assignments/assignment5>

1. Question 1

The screenshot displays a Jupyter Notebook environment with a file explorer on the left showing a project named '700754021-NeuralNetworkDeepLearning' containing files 'question_1.py' and 'question_2.py'. The main editor shows a Python script for implementing a linear SVM classifier using scikit-learn. The script reads a 'glass.csv' dataset, splits it into training and testing sets, trains a LinearSVC model, and predicts on the test set. The output console on the right shows the execution results, including a Naive Bayes Classifier accuracy of 46.15 and a detailed classification report for the SVM model.

```
1 """
2 2. Implement linear SVM method using scikit library
3 Use the same dataset above
4 Use train_test_split to create training and testing part
5 Evaluate the model on test part using score and classification_report
6 """
7
8 import pandas as pd
9 from sklearn.model_selection import train_test_split
10 from sklearn.metrics import classification_report
11 from sklearn.svm import SVC, LinearSVC
12
13 # Read glass csv
14 data_path = 'data/glass.csv'
15 data_df = pd.read_csv(data_path)
16
17 # Create dataframes for X (features), y (Target)
18 X = data_df.drop(columns=['Type'])
19 y = data_df['Type']
20
21 # Split the data into train and test sets using test_train_split
22 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
23
24 # Initialize the Linear SVM
25 svm_classifier = LinearSVC()
26
27 # Train
28 svm_classifier.fit(X_train, y_train)
29
30 # Predict
31 y_predictions = svm_classifier.predict(X_test)
```

Naive Bayes Classifier - Accuracy: 46.15
Naive Bayes Classifier - Classification Report:

	precision	recall	f1-score	support
1	0.39	0.86	0.54	21
2	0.50	0.12	0.19	26
3	0.00	0.00	0.00	7
5	0.00	0.00	0.00	2
6	0.67	1.00	0.80	2
7	0.88	1.00	0.93	7
accuracy			0.46	65
macro avg	0.41	0.50	0.41	65
weighted avg	0.44	0.46	0.37	65

Process finished with exit code 0

2. Question 2

The screenshot shows a Jupyter Notebook interface with a file explorer on the left, a code editor in the center, and a console output on the right. The code in the notebook implements a Naive Bayes classifier using the GaussianNB model from sklearn. It reads a CSV file named 'glass.csv', splits the data into training and testing sets, and evaluates the model's performance. The output in the console shows the accuracy of the Naive Bayes classifier and a classification report for a Linear SVM classifier, which is noted to have a better accuracy due to non-linearity in the data.

```
1 #####
2 1. Implement Naive Bayes method using scikit-learn library
3 Use dataset available with name glass
4 Use train_test_split to create training and testing part
5 Evaluate the model on test part using score and classification_report
6 #####
7
8 import pandas as pd
9 from sklearn.model_selection import train_test_split
10 from sklearn.naive_bayes import GaussianNB
11 from sklearn.metrics import classification_report
12
13 # Read glass csv
14 data_path = 'data/glass.csv'
15 data_df = pd.read_csv(data_path)
16
17 # Create dataframes for X (features), y (Target)
18 X = data_df.drop(columns=['Type'])
19 y = data_df['Type'] # Target
20
21 # Split the data into train and test sets using test_train_split
22 X_train, X_test, y_train, y_test = train_test_split(*arrays: X, y, test_size=0.3)
23
24 # Initialize the Naive Bayes classifier
25 naive_bayes_classifier = GaussianNB()
26
27 # Train
28 naive_bayes_classifier.fit(X_train, y_train)
29
30 # Predict
31 y_predictions = naive_bayes_classifier.predict(X_test)
32
33
```

Run: question_2

If this would cause problems for you, please provide us feedback at <https://github.com/pandas-dev/pandas/issues/54466>

```
import pandas as pd
/Users/rajkumar/Documents/Manoj/2_Academics/UCM/3_Spring-2024/NNDL/Assignments/78075
warnings.warn(
/Users/rajkumar/Documents/Manoj/2_Academics/UCM/3_Spring-2024/NNDL/Assignments/78075
warnings.warn(
/Users/rajkumar/Documents/Manoj/2_Academics/UCM/3_Spring-2024/NNDL/Assignments/78075
warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/Users/rajkumar/Documents/Manoj/2_Academics/UCM/3_Spring-2024/NNDL/Assignments/78075
warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/Users/rajkumar/Documents/Manoj/2_Academics/UCM/3_Spring-2024/NNDL/Assignments/78075
warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
Linear SVM Classifier - Accuracy: 49.23
Linear SVM Classifier - Classification Report:
precision recall f1-score support
1 0.40 0.10 0.15 21
2 0.45 0.92 0.61 26
3 0.00 0.00 0.00 7
5 0.00 0.00 0.00 2
6 0.00 0.00 0.00 2
7 0.86 0.86 0.86 7
accuracy 0.49 65
macro avg 0.28 0.31 0.27 65
weighted avg 0.40 0.49 0.39 65
SVM has better accuracy than Naive Bayes due to non-linearity in the data
Process finished with exit code 0
```

18:35 LF UTF-8 4 spaces Python 3.12 (700754021-NeuralNetworkDeepLearning) ICP-5