AT - Lesson 82 - Project_Question Copy

March 21, 2023

0.0.1 Instructions

Goal of the Project This project is designed for you to practice and solve the activities that are based on the concepts covered in the following lessons:

1. Support Vector Machines - Introduction

0.0.2 Problem Statement

In this project, you are going to create a Support Vector Machine Classification model for classification of different species of Penguins.

Getting Started:

1. Click on START Project on the panel and follow the instructions given below.
2. Create a duplicate copy of the Colab file as described below.

- Click on the **File menu**. A new drop-down list will appear.
- Click on the **Save a copy in Drive** option. A duplicate copy will get created. It will open up in the new tab on your web browser.

3. After creating the duplicate copy of the notebook, please rename it in the **YYYY-MM-DD_StudentName_Project82** format.
4. Now, write your code in the prescribed code cells.

0.0.3 List of Activities

Activity 1: Loading and Analysing the Dataset

Activity 2: Data Visualization

Activity 3: Support Vector Classifier - Model Training

Activity 4: Model Prediction and Evaluation

Activity 1: Analysing the Dataset You are given with the Seaborn dataset on Penguins. This dataset consists of the following columns:

Field	Description
species	Categorical; states species of the Penguin
island	Categorical; states home island name for the Penguin in Antartica
bill_length_mm	Numeric; Length measured from the upper edge of the beak (bill) to the base of the skull or the first feathers in mm
bill_depth_mm	Numeric; Depth measure from the lower edge of the beak to the upper edge in mm
flipper_length_mm	Numeric; Length of the fin of the Penguin in mm
body_mass_g	Numeric; Body mass of the Penguin in grams.
sex	Categorical; Gender of the Penguin

Dataset Link: https://s3-student-datasets-bucket.whjr.online/whitehat-ds-datasets/penguin.csv

Dataset Credits: Python Seaborn Package

Citation

`Allison Marie Horst, Alison Presmanes Hill, & Kristen B Gorman. (2020). palmerpenguins: Palmer`

1. Load the dataset in a DataFrame

2. Print the first five rows of the dataset.

[ ]:
```
# Import the required modules and load the dataset

# Load the DataFrame

# Display the first five rows of the DataFrame
```

3. Print the information of the DataFrame.

[ ]:
```
# Print the dataset information
```

Q: Which are the object type (categorical) columns?

A:

---

4. Find the number of missing values in each column of the DataFrame

[ ]:
```
# Print the number of missing values in each column
```

Q: Are there any missing values?

A:

Q: Which columns have missing values?

A:

---

5. Drop the missing values from all the columns and verify the same

```
[ ]: # Drop the missing values and verify

     # Drop the NAN values

     # Verify the above by printing number of missing values in each column.
```

6. Print the number of occurences of each species in `species` column.

```
[ ]: # Display the number of occurrences of each species of Penguin in the 'species'␣
      ↪column.
```

Q: What are the different species of Penguin available in the column `species`?

A:

Q: What is the type of the column `species`?

A:

---

7. Add another column `Label` to the DataFrame to convert the non-numeric target column `species` into numeric. Print first five rows of DataFrame

```
[ ]: # Add numeric column 'label' to resemble non numeric column 'species


     # Print first five rows of the DataFrame
```

8. Print the number of occurences of each species in `label` column.

```
[ ]: # Display the number of occurrences of each species of Penguin in the 'label'␣
      ↪column.
```

Q: What are the different labels available in the column `label`?

A:

---

9. Convert the non-numeric columns `sex` into numeric.

```
[ ]: # Convert the non-numeric column 'sex' to numeric in the DataFrame

     # Print the number of occurance of each label in 'sex' column
```

```
# Convert the 'sex' column to numeric

# Print the number of occurance of each label in 'sex' column after converting

# Print the Datatype of teh 'sex' column
```

10. Convert the non-numeric columns `island` into numeric.

[ ]:
```
# Convert the non-numeric column 'island' to numeric in the DataFrame

# Print the number of occurance of each label in 'island' column

# Convert the 'island' column to numeric

# Print the number of occurance of each label in 'island' column after␣
↪converting

# Print the Datatype of the 'island' column
```

Hint: For conversion of non-numeric columns to numeric use the `map()` function

After this activity, the dataset should be loaded in the DataFrame and the required columns should be of numeric type.

Activity 2: Data Visualization In this activity, you have to create scatter plots for different features and each plot differentiate between the data points of different classes (Species of the Penguin).

1. Create a scatter plot between `bill_length_mm` and `bill_depth_mm`

[ ]:
```
# Create a scatter plot between 'bill_length_mm' and 'bill_depth_mm'
```

Q Write your interpretation about the output of the graph.

A

---

2. Create a scatter plot between `bill_length_mm` and `flipper_length_mm`.

[ ]:
```
# Create a scatter plot between 'bill_length_mm' and 'flipper_length_mm'
```

Q Write your interpretation about the output of the graph.

A

---

3. Create a scatter plot between `bill_depth_mm` and `flipper_length_mm`.

[ ]:
```
# Create a scatter plot between 'bill_depth_mm' and 'flipper_length_mm'
```

Q Write your interpretation about the output of the graph.

A

After this activity, the relation between the independent features of Penguins and their speicies should be recognised. Also, student can create more such Visualization for understanding the relation between rest of the columns

---

Activity 3: Train-Test Split We need to predict the value of the `label` variable, using other variables to predict the species of the Penguin. Thus, `label` is the dependent variable and `island`, `bill_length_mm`, `bill_depth_mm`, `flipper_length_mm`, `body_mass_g`, `sex` columns are the independent variables.

1. Split the dataset into the training set and test set such that the testing set contains 33% of the instances and the remaining instances will become the training set.

2. Set `random_state = 42`.

[ ]:
```
# Split the data into Training and Testing set

# Import all the libraries

# Create X and y variables

# Split the data into training and testing sets
```

After this activity, the features and target data should be splitted into training and testing data.

Activity 4: Support Vector Classifier - Model Training Implement Linear Support Vector Classification using `sklearn.svm` module in the following way:

1. Deploy the model by importing the `SVC` class and create an object of this class.
2. Call the `fit()` function on the Support Vector Classifier object and print the score using the `score()` function.

[ ]:
```
# Build a SVC model using the 'sklearn' module.


# 1. First, call the linear 'SVC' module and store it in a variable.

# 2. Call the 'fit()' function with 'x_train' and 'y_train' as inputs.

# 3. Call the 'score()' function with 'x_train' and 'y_train' as inputs to␣
↪check the accuracy score of the model.
```

Q What is the accuracy score?

A

After this activity, a SVC model object should be trained for multiclass classification.

Activity 5: Model Prediction and Evaluation In this activity, you will make predictions for training and testing set and evaluate the model

1. Predict the values for training set by calling the `predict()` function on the Logistic Regression object.

2. Print the distribution of the labels predicted in the predicted target series for the training features.

[ ]:
```
# Make predictions on the train dataset by using the 'predict()' function.

# Compute the predictions

# Print the occurrence of each type computed in the predictions.
```

Q: Are all the label values predicted for the training features data?

A:

---

3. Predict the values for testing set by calling the `predict()` function on the Logistic Regression object.

4. Print the distribution of the labels predicted in the predicted target series for the testing features.

[ ]:
```
#Make predictions on the test dataset by using the 'predict()' function.
# Compute the predictions


# Print the occurrence of each Penguin type computed in the predictions.
```

Q: Are all the labels predicted for the test features data?

A:

---

5. Display the confusion matrix for the test set:

[ ]:
```
# Print the confusion matrix for the actual and predicted data of the test set
```

Q Are there any False Positives or False Negatives?

A

---

6. Display the classification report for the test set:

```
[ ]: # Print the classification report for the actual and predicted data of the␣
     ↪testing set (if required)
```

Q What is the f1-score for all the labels?

A

After this activity, labels should be predicted for the target columns using test features set and the model should be evaluated for the same.

Write your interpretation of the results here.

- Interpretation 1:
- Interpretation 2:

0.0.4 Submitting the Project

1. After finishing the project, click on the **Share** button on the top right corner of the notebook. A new dialog box will appear.
2. In the dialog box, make sure that '**Anyone on the Internet with this link can view**' option is selected and then click on the **Copy link** button.
3. The link of the duplicate copy (named as **YYYY-MM-DD_StudentName_Project82**) of the notebook will get copied.
4. Go to your dashboard and click on the **My Projects** option.
5. Click on the **View Project** button for the project you want to submit.
6. Click on the **Submit Project Here** button.
7. Paste the link to the project file named as **YYYY-MM-DD_StudentName_Project82** in the URL box and then click on the **Submit** button.
