

IRIS Data Classification Model

Contents

Table of Contents

[Objectives 3](file:////Users/apple/Desktop/Company%20-%20Do%20Not%20Remove/group-i/IRIS%20Data%20Classification%20Model.docx#_Toc3736813)

[Descriptive Summary 4](file:////Users/apple/Desktop/Company%20-%20Do%20Not%20Remove/group-i/IRIS%20Data%20Classification%20Model.docx#_Toc3736814)

[Subtask - a 4](file:////Users/apple/Desktop/Company%20-%20Do%20Not%20Remove/group-i/IRIS%20Data%20Classification%20Model.docx#_Toc3736815)

[Problem Statement : 4](#_Toc3736816)

[Understanding Data: 4](#_Toc3736817)

[Records 5](#_Toc3736818)

[Statistics 6](#_Toc3736819)

[AML Studio Notebook Summary 7](#_Toc3736820)

[**Frequency distribution** 9](#_Toc3736821)

[AML Studio Steps 9](file:////Users/apple/Desktop/Company%20-%20Do%20Not%20Remove/group-i/IRIS%20Data%20Classification%20Model.docx#_Toc3736822)

[& 9](file:////Users/apple/Desktop/Company%20-%20Do%20Not%20Remove/group-i/IRIS%20Data%20Classification%20Model.docx#_Toc3736823)

[Visualisation . 9](file:////Users/apple/Desktop/Company%20-%20Do%20Not%20Remove/group-i/IRIS%20Data%20Classification%20Model.docx#_Toc3736824)

[Subtasks – b,c 9](file:////Users/apple/Desktop/Company%20-%20Do%20Not%20Remove/group-i/IRIS%20Data%20Classification%20Model.docx#_Toc3736825)

[Dataset 10](#_Toc3736826)

[Split Data 14](#_Toc3736827)

[Model Selection 15](#_Toc3736828)

[Score Model 17](#_Toc3736829)

[Evaluate Model 17](#_Toc3736830)

[Visualizing Output : 17](#_Toc3736831)

[**Confusion Matrix** 18](#_Toc3736832)

[Whole Scheme & Each Step 19](file:////Users/apple/Desktop/Company%20-%20Do%20Not%20Remove/group-i/IRIS%20Data%20Classification%20Model.docx#_Toc3736833)

[**Model Selection** 20](#_Toc3736834)

[**Webservice:** 20](#_Toc3736835)

[API Key 21](file:////Users/apple/Desktop/Company%20-%20Do%20Not%20Remove/group-i/IRIS%20Data%20Classification%20Model.docx#_Toc3736836)

[**Creating the Deployment :** 21](#_Toc3736837)

[**API KEY :** 21](#_Toc3736838)

[**Secondary Key :** 22](#_Toc3736839)

[**Request Response :** 22](#_Toc3736840)

[**Batch Response :** 22](#_Toc3736841)

# Objectives

1. Use the dataset attached in the Task Email for IRIS
2. Choose two different classification methods.
3. Compare the performance of these models against each other in the same scheme, which one is performing better?
4. The report related to this part should include:
   1. Descriptive summary of the data
   2. Steps taken during the process of implementing the project and why you decided to take these steps
   3. Visualized output of each step.
   4. Include output of whole scheme + each step
   5. Produce an API Key or your project.

# Descriptive Summary

# Subtask - a

### Problem Statement :

Fisher's paper is a classic in the field and is referenced frequently to this day. It is required to design and develop a model that can predict the type of Iris Plant based on the features provided.  
  
Predicted attribute: Class of iris plant. 

Attribute Information:  
1. sepal length in cm   
2. sepal width in cm   
3. petal length in cm   
4. petal width in cm   
5. class:   
-- Iris Setosa   
-- Iris Versicolour   
-- Iris Virginica

We will be using Random Forest & Logistic Regression models for comparison to evaluate a better model out of both.

### Understanding Data:

The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other.

The dataset is considered from the attached text file provided in the mail of tasks as below:

A screenshot of a cell phone

Description automatically generated

Note: Overall understanding on the data shows that we could find 1 row of missing values and no Skewness in the dataset. Below are few of the approaches implemented to confirm the statement:

1. Created a project, added the dataset into the project and visualized: This shows we have 151 \* 5 features in this dataset.

### Records

A screenshot of a cell phone

Description automatically generated

1. Descriptive analysis overall: We have analyzed the overall statistics on this dataset using Summarize Data component which helps in providing detailed analysis about each feature.

### Statistics

Before Preprocessing :

A screenshot of a cell phone

Description automatically generated

After Preprocessing: Removing empty record:

A screenshot of a cell phone

Description automatically generated

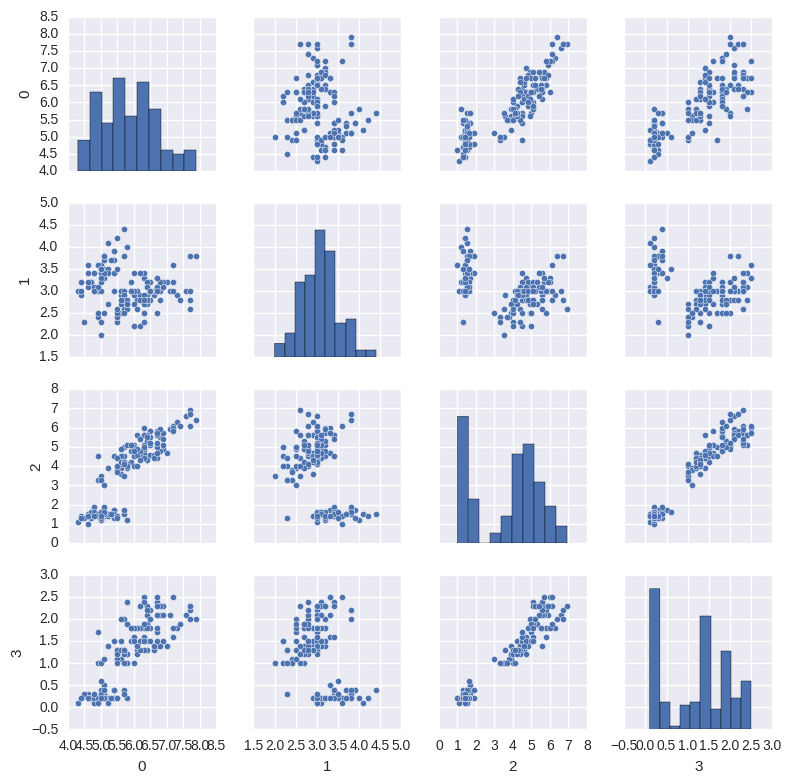
Note: The above information shows that all the variables are not skewed and has normalized data as both the mean and median are almost similar.

1. As AML studio doesn’t by default provide much visualization, we went ahead with creating many visualizations in understanding the data better using Notebook creation inside the AML Studio.
2. This helps in understanding the details of each feature and its correlation in much better way.

### AML Studio Notebook Summary

A screenshot of a social media post

Description automatically generated



The above image provides an understanding on how the dataset features are correlated. This clearly shows that all the features are closely correlated.

1. As all the features are correlated, all are considered for the model creation.
2. Frequency distribution of all the features gives a better understanding on how the features are distributed with respect to each feature. Below image explain it in more detail:

### **Frequency distribution**

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

# AML Studio Steps

# &

# Visualisation .

# Subtasks – b,c

1. Create a Project and append the dataset from the sample datasets provided by microsoft.

### Dataset

A screenshot of a cell phone

Description automatically generated

Once done with adding the dataset , create a Python Notebook with Python 3 for the dataset to analysis and visualizations. This can be implemented by clicking on the dataset and Python 3 Notebook as below :

A screenshot of a cell phone

Description automatically generated

This helps in creating understanding the descriptive statistics and summary with better visualizations.

1. As the column names are missing its recommended to include the column names through Edit Metadata Component .

A screenshot of a cell phone

Description automatically generated

1. It is also recommended to convert the target feature to be converted to catergorical so that the tree classification can easily navigate the branches .

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

1. In the descriptive statistics we found that one of the Null record has been created . We use Clean Missing Data Component to remove the missing row.

A screenshot of a social media post

Description automatically generated

1. Its recommended to summarize the data to understand if the data is clean enough to continue for model creation.

A screenshot of a cell phone

Description automatically generated

1. Once the data is found to be clean , its required to split the data into training and testing data with 75% of Training and 25% of Testing Data.

### Split Data

A close up of a womans face

Description automatically generated

Note : While splitting the data the most important aspect is to consider selecting target variable as Stratified Column which helps in creating a balanced training and testing datasets while splitting

A screenshot of a cell phone

Description automatically generated

1. As this project requires to use two classification algorithms, Random Forest (Decision Tress - Bagging) component & Multiclass Logistic Regression is included into the project.

### Model Selection

A close up of a map

Description automatically generated

Considering the Hypertuning Parameter components and its results below are the indices we considered for the both models.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

1. Considering Classicfication Models Components and Train Dataset from split data as Input sources to Train a Model , Train Model component will be integrated.

A close up of a map

Description automatically generated

Train models takes the target feature to get trained upon.

A screenshot of a social media post

Description automatically generated

1. Inorder to score the dataset , Score Model component takes the Test Data to get the results on trained model

### Score Model

A close up of a map

Description automatically generated

Once the models are scored this can be evaluated using Evaluate Model Component where both the models are mapped to single Evaluate Model Compoenent.

### Evaluate Model

A screenshot of a cell phone

Description automatically generated

### Visualizing Output :

Confusion Matrix gives a clear clarify & accuracy on the model . This clearly confirms that Random Forest is giving 97% of overall Accuracy when compared to Logistic Regression which has 94%

A screenshot of a cell phone

Description automatically generated

### **Confusion Matrix**

Confusion Matrix provides a clear understanding on dataset which has Virginica closely mapped to Versicolour in both the models.

A screenshot of a social media post

Description automatically generated

# Whole Scheme & Each Step

Subtask - d

whole scheme + each step

The whole scheme design looks as below :

A close up of a map

Description automatically generated

### **Model Selection**

As Random Forest gives better results than the Logistic Regression Model , Its recommended to continue only with Random Forest Model .

### **Webservice:**

A webservice Predictive Experiment will be generated based on Random Forest. After generating the Predictive Experiment as below :

A screenshot of a cell phone

Description automatically generated

# API Key

Subtask - e

### **Creating the Deployment :**

For creating the deployment profile , It is required to run the Predictive Experiment and then deploy it .

A close up of a logo

Description automatically generated

Once the deployment is completed the AML Studio generates an API Key which can be consumed through any of the services and can be testing using C#, Python …

A screenshot of a social media post

Description automatically generated

### **API KEY :**

KTUOvY42uVJyTxrw12xeigk9N/4skuALKOsuyA3Ni5r+qVY/q4044r1Cy/MzYm7hgmsBrfgp8XBBcobu1tsB5A==

### **Secondary Key :**

jyy2WIm/RyYEXPebMKa8fcTfaNmwkFWJ8ykdNXLuyIr8efo8yQaHgLnUQQkf7LcyorvECYj5x0eSJ4C8reu/jQ==

### **Request Response :**

<https://ussouthcentral.services.azureml.net/workspaces/0791fbe1c5254bca93cbfb90611752c5/services/4408d24d1fbc4fe9b69119d889fa7865/execute?api-version=2.0&format=swagger>

### **Batch Response :**

<https://ussouthcentral.services.azureml.net/workspaces/0791fbe1c5254bca93cbfb90611752c5/services/4408d24d1fbc4fe9b69119d889fa7865/jobs?api-version=2.0>