A close up of a sign

Description automatically generated

REPORT

CLASSIFICATION OF IRIS FLOWER USING MACHINE LEARNING

A close up of a flower

Description automatically generated

NAME : KOMAL GINWAL

STEVENS CWID : 10449929

# *Table of Contents*

1. ***Data***
2. ***Libraries***
3. ***Summarize the dataset.***
4. ***Data Visualization***
5. ***Evaluation of Algorithm***
6. ***Result***
7. ***Conclusion***
8. ***References***

***Abstract***

***Classification of IRIS data set would be discovering patterns from examining petal and sepal size of the IRIS flower and how the prediction was made from analyzing the pattern to from the class of IRIS flower. In this paper we train the machine learning model with data and when unseen data is discovered the predictive model predicts the species using what it has been learnt from the trained data. To design and implement the Identification of Iris Flower species using machine learning using Python and the tool Scikit-Learn.***

***Execution***

***In a terminal or command window, navigate to the top-level project directory Iris Flower (that contains this README) and run one of the following commands:***

***ipython notebook report.ipynb***

***or***

***jupyter notebook report.ipynb***

***or if you have 'Jupyter Lab' installed***

***jupyter lab***

***This will open the Jupyter/iPython Notebook or Jupyterlab software and project file in your browser.***

## **1 . About Data**

|  |  |
| --- | --- |
| **Associated Task** | Classification |
| **Data Set Characteristics** | Multivariate |
| **Attribute Characteristics** | Real |
| **Number of Instances** | 150 |
| **Number of Attributes** | 4 |
| **Missing Values?** | **No** |

The Iris flower data set or Fisher's Iris data set is a multivariate data set introduced by the British statistician and biologist Ronald Fisher in his 1936 paper. T

The data set contains 3 classes of 50 instances each, total 150 instances, where each class refers to a type of Iris plant. One class is linearly separable from the other 2 and the latter are **not linearly separable** from each other.

**Predicting attribute:** Class of Iris plant.

**Attribute Information:** We have 4 features in this dataset and a target variable class.

* sepal length in cm.
* sepal width in cm.
* petal length in cm.
* petal width in cm.
* Class:
  + Iris Setosa
  + Iris Versicolour
  + Iris Virginica

**2. Import libraries**

A screenshot of a social media post

Description automatically generated

**2.1 Load Dataset**

Pandas is used to load the data.

A picture containing screenshot

Description automatically generated

**3. Summarize the Dataset**

In this step we are going to take a look at the data a few different ways:

* 1. Dimensions of the dataset.
  2. Peek at the data itself.
  3. Statistical summary of all attributes.
  4. Breakdown of the data by the class variable.

A close up of a logo

Description automatically generated

A close up of a piece of paper

Description automatically generated

## **Data Visualization**

1. Univariate plots to better understand each attribute.
2. Multivariate plots to better understand the relationships between attributes.

This gives us a much clearer idea of the distribution of the input attributes:

A screenshot of a social media post

Description automatically generated

A close up of a logo

Description automatically generated

Image. Box and Whisker Plots for Each Input Variable for the Iris Flowers Dataset

It looks like perhaps two of the input variables have a Gaussian distribution. This is useful to note as we can use algorithms that can exploit this assumption.

A close up of text on a white background

Description automatically generated

Image. Histogram Plots for Each Input Variable for the Iris Flowers Dataset

A close up of text on a white background

Description automatically generated

Image. Scatter Matrix Plot for Each Input Variable for the Iris Flowers Dataset

## **5. Evaluation of Algorithms**

5.1. Creation of validation dataset.

* 1. Test Harness
  2. Build multiple different models to predict species from flower measurements
  3. Select the best model.

### **5.1 Creation of Validation Dataset**

Statistical methods to estimate the accuracy of the models that we create on unseen data. We also want a more concrete estimate of the accuracy of the best model on unseen data by evaluating it on actual unseen data.

That is, we are going to hold back some data that the algorithms will not get to see and we will use this data to get a second and independent idea of how accurate the best model might actually be.

We will split the loaded dataset into two, 80% of which we will use to train, evaluate and select among our models, and 20% that we will hold back as a validation dataset.

### **5.2 Test Harness**

Stratified 10-fold cross validation to estimate model accuracy.

This will split our dataset into 10 parts, train on 9 and test on 1 and repeat for all combinations of train-test splits.

A screenshot of a cell phone

Description automatically generated

We are using the metric of ‘accuracy‘ to evaluate models. This is a ratio of the number of correctly predicted instances in divided by the total number of instances in the dataset multiplied by 100 to give a percentage (e.g. 95% accurate). We will be using the scoring variable when we run build and evaluate each model next.

**5.3. Build Model**

We'll evaluate these 6 algorithms:

* Logistic Regression (LR)
* Linear Discriminate Analysis (LDA)
* K-Nearest Neighbours (KNN)
* Classification and Regression Trees (CRT)
* Guassian Naive Bayes (GNN)
* Support Vector Machine (SVM)

This is a good mixture of simple linear (LR and LDA), nonlinear (KNN, CRT, NB and SVM) algorithms. We reset the random number seed before each run to ensure that the evaluation of each algorithm is performed using exactly the same data splits. It ensures the results are directly comparable.

**A screenshot of a social media post

Description automatically generated**

### **5.4 Select Best Model**

We now have 6 models and accuracy estimations for each. We need to compare the models to each other and select the most accurate.

Running the example above, we get the following raw results:

A screenshot of a social media post

Description automatically generated

A useful way to compare the samples of results for each algorithm is to create a box and whisker plot for each distribution and compare the distributions.

A screenshot of a social media post

Description automatically generated

Box and Whisker Plot Comparing Machine Learning Algorithms on the Iris Flowers Dataset

1. **RESULT:**

## **Make Predictions:**

We can make predictions on the validation dataset by fitting the entire model on training dataset

A screenshot of a cell phone

Description automatically generated

Now we want to get an idea of the accuracy of the model on our validation set.

This will give us an independent final check on the accuracy of the best model. It is valuable to keep a validation set just in case you made a slip during training, such as overfitting to the training set or a data leak. Both of these issues will result in an overly optimistic result.

* 1. **Evaluate Predictions**

A screenshot of a cell phone

Description automatically generated

The results in the previous section suggest that the SVM was perhaps the most accurate model. We will use this model as our final model.

We can evaluate the predictions by comparing them to the expected results in the validation set, then calculate classification accuracy,

We can see that the accuracy is 0.98 or about 98% on the hold out dataset of Supervised Vector Machine Algorithm. Finally, the classification report provides a breakdown of each class by precision, recall, f1-score and support showing excellent results (granted the validation dataset was small).

1. **Conclusion**

The primary goal of supervised learning is to build a model that “generalizes”. Here in this project we make predictions on unseen data which is the data not used to train the model hence the machine learning model built should accurately predicts the species of future flowers rather than accurately predicting the label of already trained data.

1. **References**

1. https://ipasj.org/IIJCS/Volume5Issue8/IIJCS-2017-08-18-18.pdf

2. https://elitedatascience.com/machine-learning-for-beginners#stock-prices

3.Diptam Dutta, Argha Roy, Kaustav Choudhury, “Training Aritificial Neural Network Using Particle Swarm Optimization Algorithm”, International Journal on Computer Science And Engineering(IJCSE), Volume 3, Issue 3, March 2013.