Iris Flower Dataset

Manish Mohan Kamble

Manishkamble7547@gmail.com

July 2021

1.Application Type

This is a classification project ,since the variable to be predicted is categorical.

The goal here is to model the probabilities of class membership, conditioned on the flower fetures.

2.About Data Set

The data source is the file iris_flowers.csv.it contains the data for this example in comma separated values(csv) format. The number of columns is 5,and the number of rows is 150

The variables are

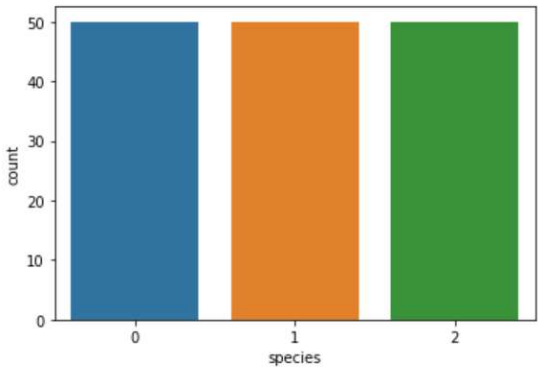
- Sepal_length : Sepal length, in centimeters, used as input
- Sepal_width : Sepal width , in centimeters ,used as input
- Petal_width : Petal length , in centimeters , used as input
- Petal_width : Petal width , in centimeters , used as input
- Species : iris Setosa , Versicolor , or Virginica used as a target

Species is well-distributed , since there is the same number of virginica , setosa, and versicolor samples.

Neural networks work with numbers. In this regard , the categorical variable “class” is transformed into three numerical variables as follows

- Iris_setosa : 1 0 0
- Iris_versicolor : 0 1 0
- Iris_virginica : 0 0 1

```
plt.figure()
sns.countplot(data = df, x='species')
plt.show()
```



3. EDA & Preprocessing

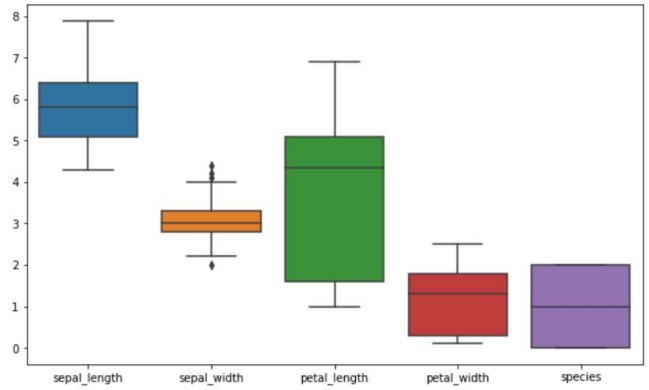
Shape of the data is (150, 5) and dataset is very clean and has no null values present. Describe function shows statistical summary of data. Then we plot heatmap for correlation of data. species is categorical variable so we transformed it into numerical variable using Label Encoder.

We visualize continuous variables by using distplot and also skew for distribution .most of the data is normally distribution. We also use boxplot and pair plot for better understanding.

There are clear separations shown , especially for pairs of features having ‘variance’.

Then we use iloc for seprating labels and feature and split the data into training set and test set using train test split.after that we scale data using standardscaler.

```
[ ] # Distribution of data using Boxplot
    plt.figure(figsize=(10,6))
    sns.boxplot(data=df)
    plt.show()
```



4 . Neural Network

The next step is to choose a 1 hidden layer neural network .for multi-class classification .

The neural network must have four inputs since the data set has four input variable (sepal length ,sepal width , petal length ,and petal width).

And we use 1 perceotron layers :

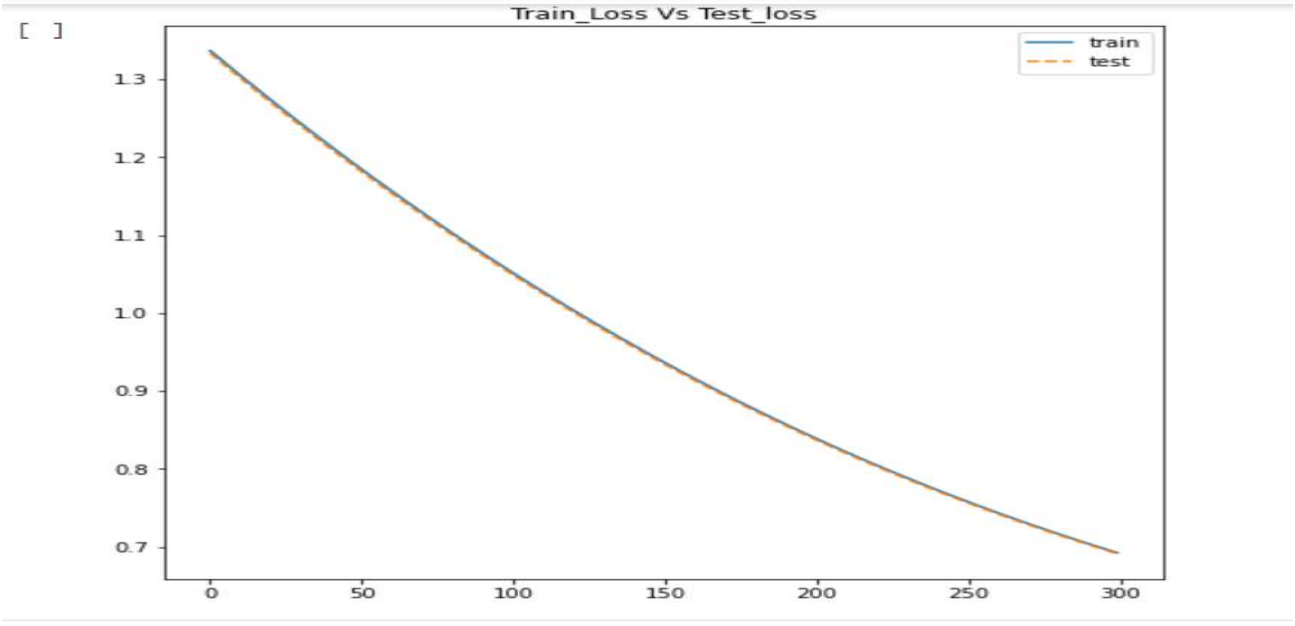
The layer has 3 neuron ,4 inputs and a softmax activation function.

The neural network has three output since the target variable contains 3 classes (setosa , versicolor , and virginica).

For compiling we use adam optimizer loss is sparse_categorical_crossentropy and accuracy for metrics then we fit our model with 300 epochs and 150 batch size.

The following chart shows how the training and testing loss decrease with epochs during the training process.

```
[ ]  fig , ax = plt.subplots(figsize=(8,8))
     plt.title('Train_Loss Vs Test_loss')
     plt.plot(model_history.history['loss'], label='train')
     plt.plot(model_history.history['val_loss'],label='test',linestyle='--')
     plt.legend()
     plt.show()
```

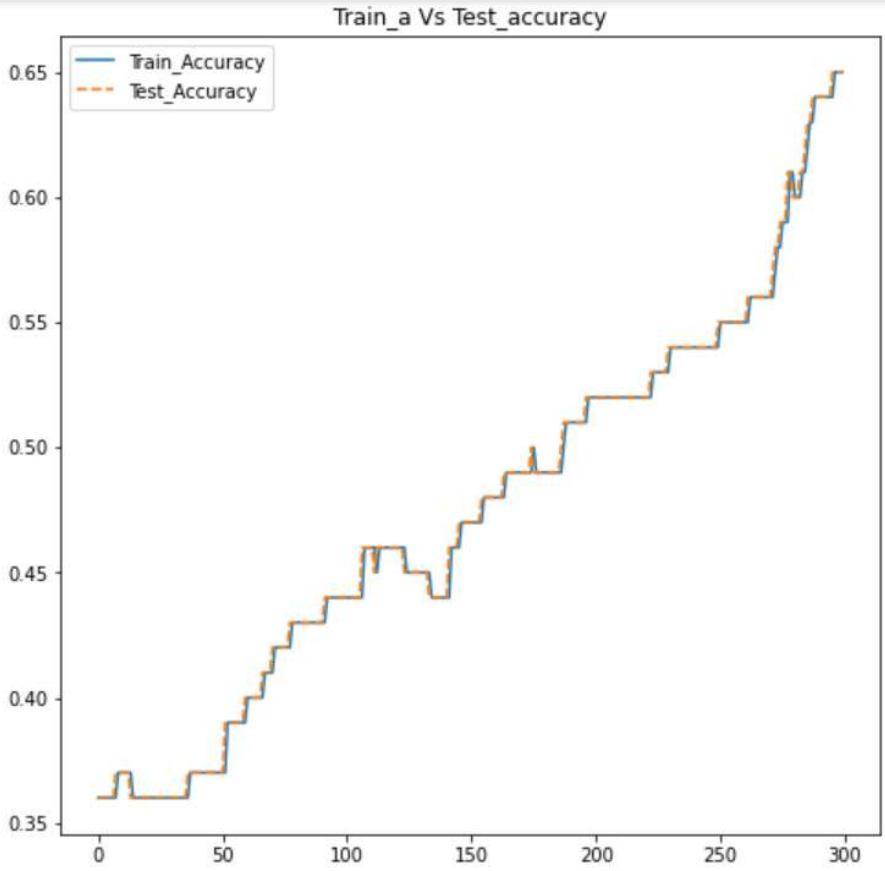


This chart shows how the training and testing accuracy increase with epochs during the training process.

```
[ ]  fig , ax = plt.subplots(figsize=(8,8))
     plt.title('Train_a Vs Test_accuracy')
     plt.plot(model_history.history['accuracy'], label='Train_Accuracy')
     plt.plot(model_history.history['val_accuracy'],label='Test_Accuracy',linestyle='--')
     plt.legend()
```

+ Code + Text

[ ]



Then we get y_pred values from X_test and y_pred contain array so to return the indices of maximum values along an axis we use argmax function.

We create Classification report to get evaluation metrics .

Following image shows classification report

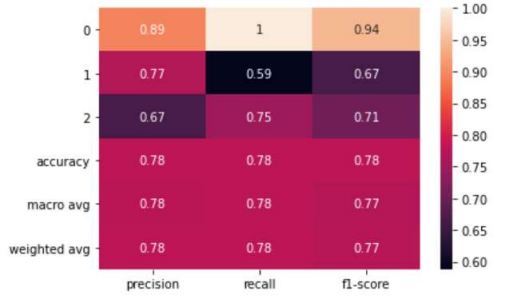
And we also plot that classification report and confusion metrics using heatmap for better understanding.

```
[ ]  print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	0.89	1.00	0.94	17
1	0.77	0.59	0.67	17
2	0.67	0.75	0.71	16
accuracy			0.78	50
macro avg	0.78	0.78	0.77	50
weighted avg	0.78	0.78	0.77	50

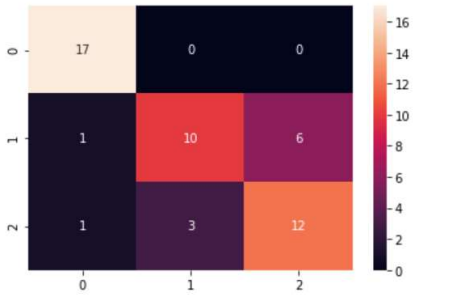
```
[ ] #.iloc[:-1,:] to exclude support
    sns.heatmap(pd.DataFrame(clf_report).iloc[:-1,:].T,annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9bb17b4b50>
```




```
[ ] mat1 = confusion_matrix(y_test,y_pred)
    sns.heatmap(mat1,annot=True,fmt='d')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9ba6
```



5 . Conclusion

Here we build a ANN model with 100 epochs and visualize train vs test loss, train and test accuracy,classification report and confusion metrics .So we can see that the model achieved an estimated classification accuracy of about 78%.

6.References

- UCI Machine Learning Repository. Iris Flower Dataset | Kaggle
- Fisher, R.A. "The use of multiple measurements in taxonomic problems" Annual Eugenics, 7, Part II, 179-188 (1936); also in "Contributions to Mathematical Statistics" (John Wiley, NY, 1950).