INDUSTRIAL INTERNSHIP

**WEEKLY PERFORMANCE REPORT (WPR)**

# Student Name : Aman Srivastava

**Head-Coordinator Name : vatshal goshwami**

**Organization : Cureya Hours Worked : Monday 5 , Tuesday 5**

**Wednesday 5 , Thursday 5 , Friday 5 , Saturday 5 .**

On Monday a new task was given by team leader to Apply the Naive Bayes Classifier Algorithm on Iris database And predict the types of flower based on measurements.

**Monday**:-I started reseraching all the topic related to naive bayes classifier.

Here is the some glance:

Introduction to naive Byes algorithm:

In machine learning, Naïve Bayes classification is a straightforward and powerful algorithm for the classification task.

Naive Bayes classifiers are a collection of classification algorithms based on Bayes’ Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

Naïve Bayes models are also known as simple Bayes or independent Bayes. All these names refer to the application of Bayes’ theorem in the classifier’s decision rule. Naïve Bayes classifier applies the Bayes’ theorem in practice. This classifier brings the power of Bayes’ theorem to machine learning.

It uses Bayes theorem of probability for prediction of unknown class.

Bayes’ Theorem finds the probability of an event occurring given the probability of another event that has already occurred. Bayes’ theorem is stated mathematically as the following equation:

P(A|B) = {P(B|A) P(A)} / { P(B)}

where A and B are events and P(B) ? 0.

Basically, we are trying to find probability of event A, given the event B is true. Event B is also termed as evidence.

P(A) is the priori of A (the prior probability, i.e. Probability of event before evidence is seen). The evidence is an attribute value of an unknown instance(here, it is event B).

P(A|B) is a posteriori probability of B, i.e. probability of event after evidence is seen.

Now, with regards to our dataset, we can apply Bayes’ theorem in following way:

P(y|X) = {P(X|y) P(y)} /{P(X)}

where, y is class variable and X is a dependent feature vector (of size n) where:

X = (x\_1,x\_2,x\_3,.....,x\_n)

**Tuesday:**

**Naive Bayes algorithm intuition:-**

Naïve Bayes Classifier uses the Bayes’ theorem to predict membership probabilities for each class such as the probability that given record or data point belongs to a particular class. The class with the highest probability is considered as the most likely class. This is also known as the Maximum A Posteriori (MAP).

The MAP for a hypothesis with 2 events A and B is

MAP (A)

= max (P (A | B))

= max (P (B | A) \* P (A))/P (B)

= max (P (B | A) \* P (A))

Here, P (B) is evidence probability. It is used to normalize the result. It remains the same, So, removing it would not affect the result.

Naïve Bayes Classifier assumes that all the features are unrelated to each other. Presence or absence of a feature does not influence the presence or absence of any other feature.

In real world datasets, we test a hypothesis given multiple evidence on features. So, the calculations become quite complicated. To simplify the work, the feature independence approach is used to uncouple multiple evidence and treat each as an independent one.

**Types of Naive Bayes algorithm:-**

There are 3 types of Naïve Bayes algorithm. The 3 types are listed below:-

Gaussian Naïve Bayes

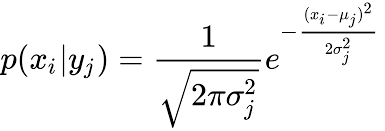
Multinomial Naïve Bayes

Bernoulli Naïve Bayes

These 3 types of algorithm are explained below.

Gaussian Naïve Bayes algorithm

When we have continuous attribute values, we made an assumption that the values associated with each class are distributed according to Gaussian or Normal distribution. For example, suppose the training data contains a continuous attribute x. We first segment the data by the class, and then compute the mean and variance of x in each class. Let µi be the mean of the values and let σi be the variance of the values associated with the ith class. Suppose we have some observation value xi . Then, the probability distribution of xi given a class can be computed by the following equation –



## **Multinomial Naïve Bayes algorithm**

With a Multinomial Naïve Bayes model, samples (feature vectors) represent the frequencies with which certain events have been generated by a multinomial (p1, . . . ,pn) where pi is the probability that event i occurs. Multinomial Naïve Bayes algorithm is preferred to use on data that is multinomially distributed. It is one of the standard algorithms which is used in text categorization classification.

## **Bernoulli Naïve Bayes algorithm**

In the multivariate Bernoulli event model, features are independent boolean variables (binary variables) describing inputs. Just like the multinomial model, this model is also popular for document classification tasks where binary term occurrence features are used rather than term frequencies.

**Wednesday:**

# ****Applications of Naive Bayes algorithm:****

Naïve Bayes is one of the most straightforward and fast classification algorithm. It is very well suited for large volume of data. It is successfully used in various applications such as :

1. Spam filtering
2. Text classification
3. Sentiment analysis
4. Recommender systems

# ****Import libraries****

from sklearn.naive\_bayes import GaussianNB

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import confusion\_matrix

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

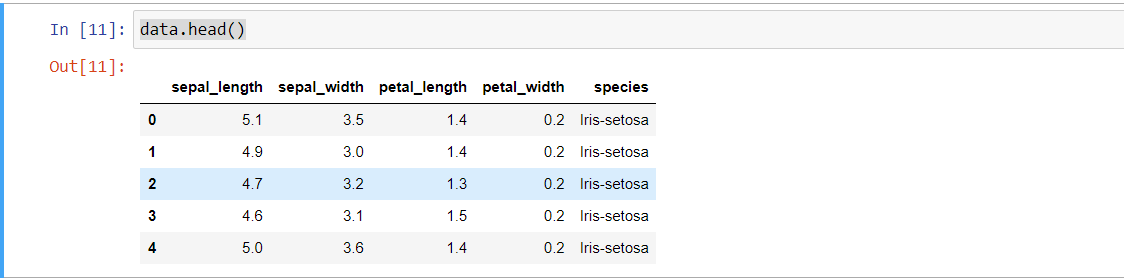
import seaborn as sns; sns.set()

**Import Dataset:**

data = pd.read\_csv('IRIS.csv')

**View top 5 dataset:**

data.head()



**Create the X, Y, Training and Test**

xdata = data.drop('species', axis=1)

ydata = data.loc[:, 'species']

**Init the Gaussian Classifier**

model = GaussianNB()

**Train the model**

model.fit(xdata, ydata)

**Predict Output**

pred = model.predict(xdata)

**Thursday:**

# ****Confusion matrix:****

A confusion matrix is a tool for summarizing the performance of a classification algorithm. A confusion matrix will give us a clear picture of classification model performance and the types of errors produced by the model. It gives us a summary of correct and incorrect predictions broken down by each category. The summary is represented in a tabular form.

Two types of outcomes are possible while evaluating a classification model performance.

These two outcomes are summarized and plot in a confusion matrix given below.

# Predict Output # Plot Confusion Matrix

mat = confusion\_matrix(pred, ydata)

names = np.unique(pred)

sns.heatmap(mat,square=True,annot=True,fmt='d',cbar=False,xticklabels=names, yticklabels=names)

plt.xlabel('Truth')

plt.ylabel('Predicted')

pred=model.predict(xdata)

**Friday:**

**Gaussian Naive Bayes model accuracy(in %):**

from sklearn.datasets import load\_iris

iris = load\_iris()

# store the feature matrix (X) and response vector (y)

X = iris.data

y = iris.target

# splitting X and y into training and testing sets

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.4, random\_state=1)

# training the model on training set

from sklearn.naive\_bayes import GaussianNB

gnb = GaussianNB()

gnb.fit(X\_train, y\_train)

# making predictions on the testing set

y\_pred = gnb.predict(X\_test)

# comparing actual response values (y\_test) with predicted response values (y\_pred)

from sklearn import metrics

print("Gaussian Naive Bayes model accuracy(in %):", metrics.accuracy\_score(y\_test, y\_pred)\*100)

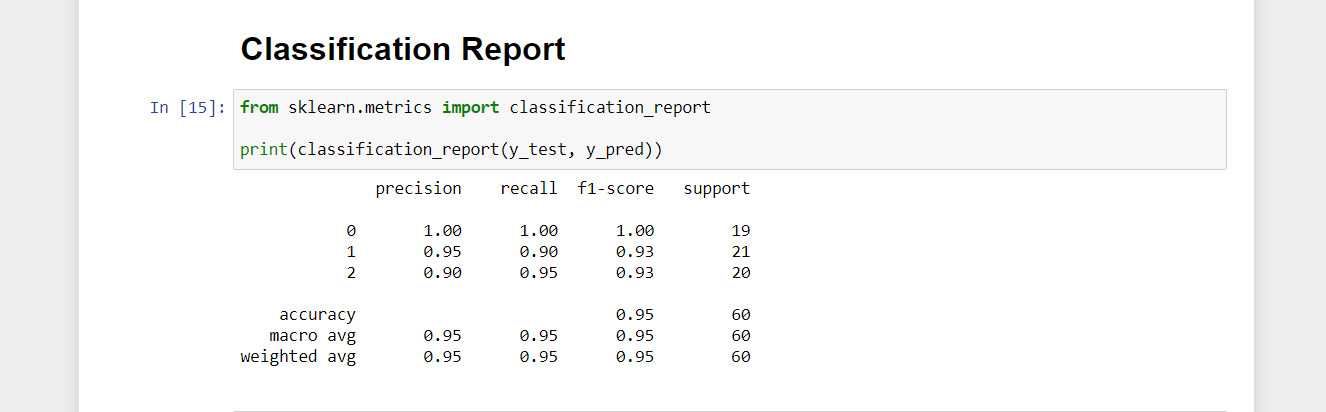
**Classification Report:**

**Classification report** is another way to evaluate the classification model performance. It displays the **precision**, **recall**, **f1** and **support** scores for the model

We can print a classification report as follows:-

from sklearn.metrics import classification\_report

print(classification\_report(y\_test, y\_pred))



**Calculate class probabilities**

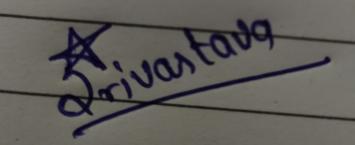
y\_pred\_prob = gnb.predict\_proba(X\_test)[0:10]

y\_pred\_prob

# **Results and conclusion**

1. In this project, I build a Gaussian Naïve Bayes Classifier model to predict types of flower based on measurements. The model yields a very good performance as indicated by the model accuracy which was found to be 95 %.
2. I have compared the model accuracy score which is 95% with null accuracy score which is 0.7582. So, we can conclude that our Gaussian Naïve Bayes classifier model is doing a very good job in predicting the class labels

So, we can conclude that our classifier does a very good job in predicting types of flower based on measurements

Student Signature  Date 16/07/2021 Head- coordinator Signature \_Date

Instructions: After the completed report has been signed by both the student and Head-coordinator, the head-coordinator shall scan the form to a pdf format and email it to the Director-1 ([bpmishra435@gmail.com](mailto:bpmishra435@gmail.com)) of the company. Specific problems, concerns or suggestions from either the student/ head-coordinator should be emailed separately to the C.E.O.([info@cureya.in](mailto:info@cureya.in)) of the company.