Analysis of admission data using data mining algorithms

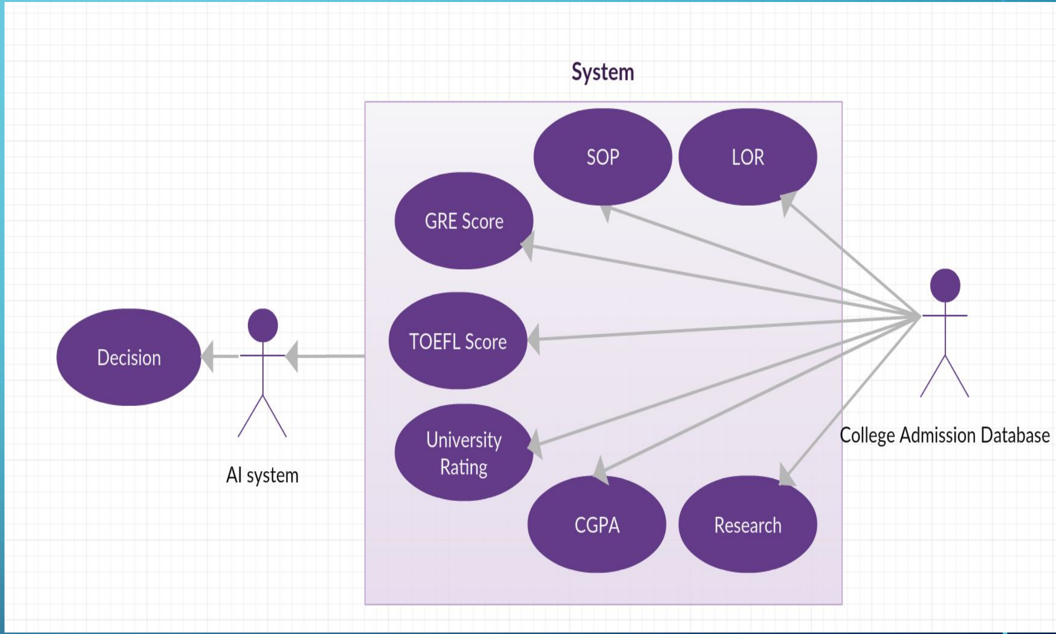
HIMANI GUPTA-95529

PRIYA PHAPALE - 94662

PRAGYA MOHARATHA-95474

Introduction

- College application process can be quite intimidating.
- Objective here is to analyze student information and predict decision on acceptance for the student.
- We are using **KNN, Naive Bayes, SVM** to predict decision on student's acceptance.
- This is a binary classification problem. **Decision says - 1** if the student will get an acceptance letter or **0** otherwise.



Sample Data and Statistics

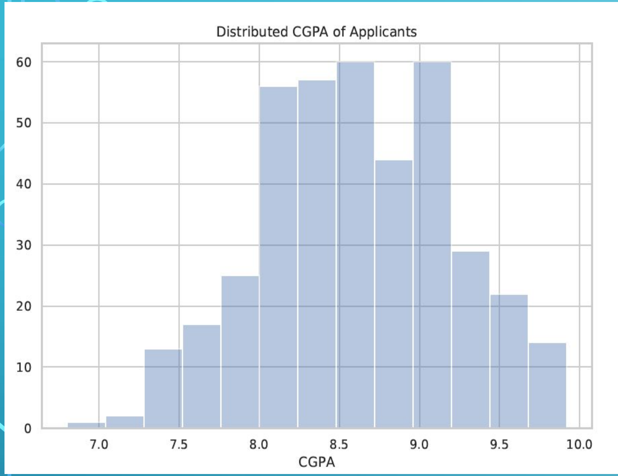
SampleData

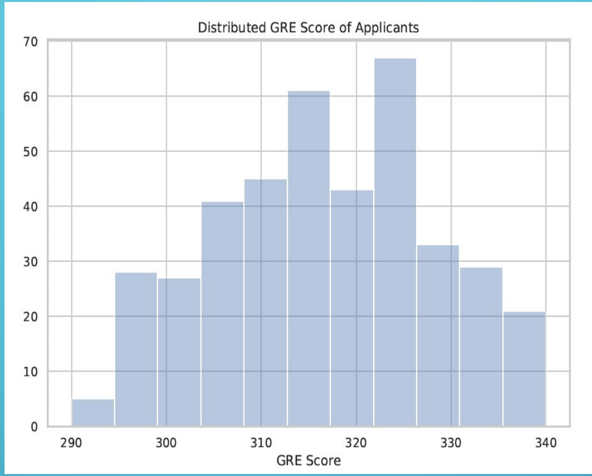
	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit	Decision
0	1	337	118	4	4.5	4.5	9.65	1	0.92	1
1	2	324	107	4	4	4.5	8.87	1	0.76	0
2	3	316	104	3	3	3.5	8	1	0.72	0
3	4	322	110	3	3.5	2.5	8.67	1	0.8	1
4	5	314	103	2	2	3	8.21	0	0.65	0
5	6	330	115	5	4.5	3	9.34	1	0.9	1
6	7	321	109	3	3	4	8.2	1	0.75	0

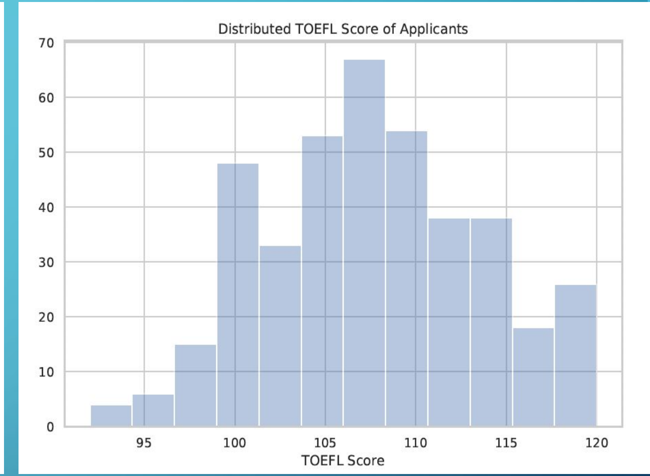
	count	mean	std	min	25%	50%	75%	max
Serial No.	400.0	200.500000	115.614301	1.00	100.75	200.50	300.2500	400.00
GRE Score	400.0	316.807500	11.473646	290.00	308.00	317.00	325.0000	340.00
TOEFL Score	400.0	107.410000	6.069514	92.00	103.00	107.00	112.0000	120.00
University Rating	400.0	3.087500	1.143728	1.00	2.00	3.00	4.0000	5.00
SOP	400.0	3.400000	1.006869	1.00	2.50	3.50	4.0000	5.00
LOR	400.0	3.452500	0.898478	1.00	3.00	3.50	4.0000	5.00
CGPA	400.0	8.598925	0.596317	6.80	8.17	8.61	9.0625	9.92
Research	400.0	0.547500	0.498362	0.00	0.00	1.00	1.0000	1.00
Chance of Admit	400.0	0.724350	0.142609	0.34	0.64	0.73	0.8300	0.97
Decision	400.0	0.320000	0.467060	0.00	0.00	0.00	1.0000	1.00

No missing values found.

Data Exploration







Features:

- GRE Score - Continuous
- TOEFL Score - Continuous
- University Rating - Ordinal Categorical
- SOP - Ordinal Categorical
- LOR - Ordinal Categorical
- CGPA - Continuous
- Research - Discrete

Target:

- **Decision** - Categorical

Chance of Admittance → Decision.

>= 80% is “Accepted”

< 80% is “Rejected”

IMPLEMENTATION

- **Importing libraries**

  - (Pandas, Numpy, matplotlib, sklearn)

- **Importing the dataset**
- **Exploratory data analysis**
- **Data preprocessing**

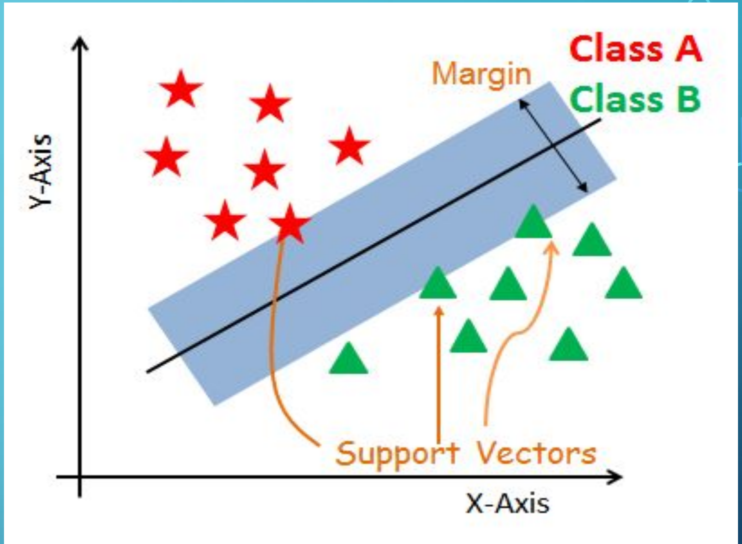
  - Divide the data into attributes and labels

  - Divide the data into training and testing sets.

- **Training the algorithm**
- **Making predictions**
- **Evaluating the algorithm**
- **Results**

SUPPORT VECTOR MACHINES (SVM)

- Supervised classification algorithm
- Finds most optimal decision boundary that maximizes the distance from the nearest data points of all the classes
- SVM kernels- Transforms low-dimensional input space and transforms it into a higher dimensional space



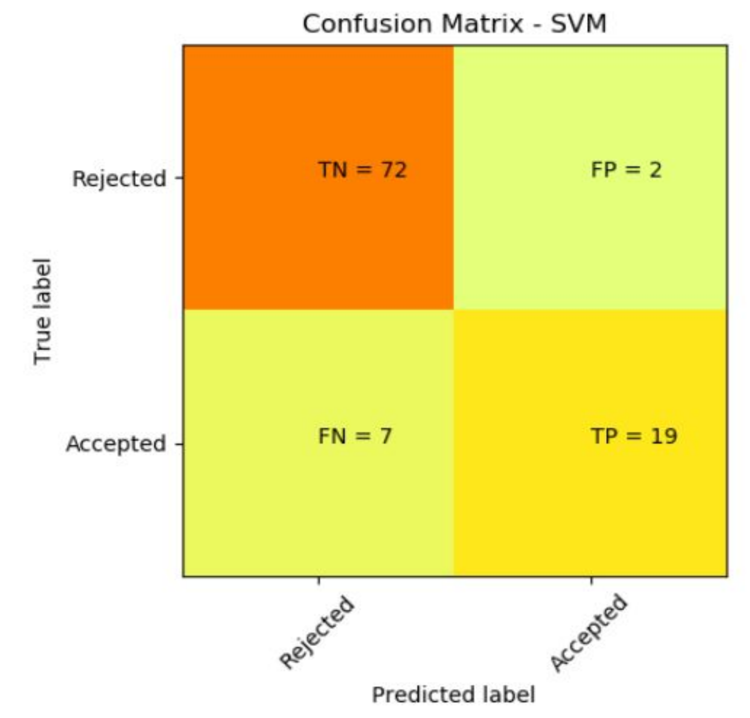
Linear Kernel	`K(x, xi) = sum(x * xi)`
Polynomial Kernel	`K(x,xi) = 1 + sum(x * xi)^d`
Radial Basis Function Kernel	`K(x,xi) = exp(-gamma * sum((x – xi^2))`

SVM Analysis of different kernels

```python
# Kernel - linear
svc = SVC(kernel='linear', probability=True)
svc.fit(X_train, y_train)
y_pred_linear = svc.predict(X_test)

# Kernel - Poly
svc = SVC(kernel='poly', probability=True)
svc.fit(X_train, y_train)
y_pred_poly = svc.predict(X_test)

# Kernel - rbf
svc = SVC(kernel='rbf', probability=True)
svc.fit(X_train, y_train)
y_pred_rbf= svc.predict(X_test)
```

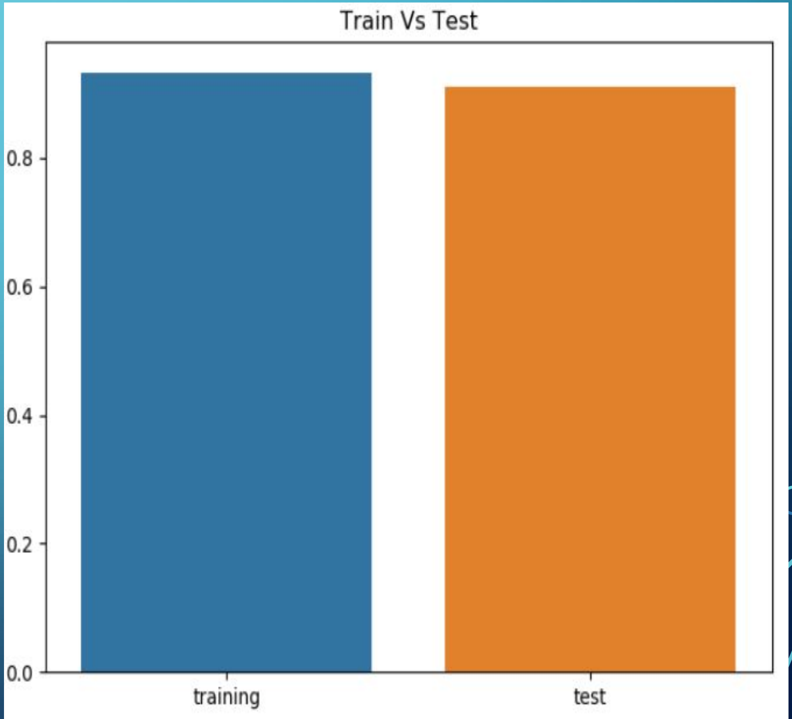
```
Linear:    0.91
Poly:    0.87
Radial:    0.93
------Linear-------
 [[72  2]
 [ 7 19]]
------Poly------
 [[74  0]
 [13 13]]
------RBF------
 [[72  2]
 [ 5 21]]
```

Confusion matrix



Accuracy

Training - 95%, Test - 93%



NAIVE BAYES ALGORITHM

- Naive Bayes is a statistical classification technique based on Bayes Theorem.
- It is one of the simplest supervised learning algorithms.
- Naive Bayes classifier assumes that the effect of a particular feature in a class is independent of other features.
- Even if these features are interdependent, these features are still considered independently. This assumption is called class conditional independence.
- This assumption simplifies computation, and that's why it is considered as naive.

01 CALCULATE PRIOR PROBABILITY FOR GIVEN CLASS LABELS

02 CALCULATE CONDITIONAL PROBABILITY WITH EACH ATTRIBUTE FOR EACH CLASS

03 MULTIPLY SAME CLASS CONDITIONAL PROBABILITY.

04 MULTIPLY PRIOR PROBABILITY WITH STEP 3 PROBABILITY.

05 SEE WHICH CLASS HAS HIGHER PROBABILITY, HIGHER PROBABILITY CLASS BELONGS TO GIVEN INPUT SET STEP.

Model Generation

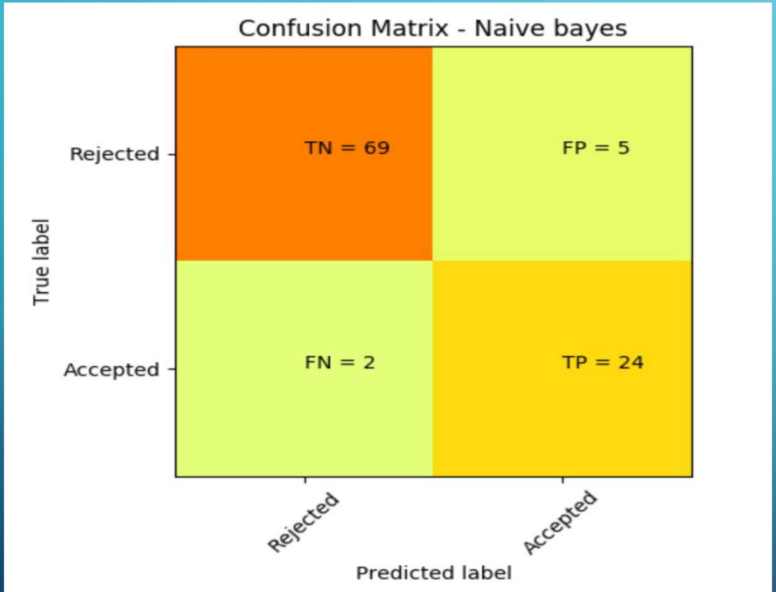
Gaussian Naive Bayes: It is used in classification and it assumes that features follow a normal distribution.

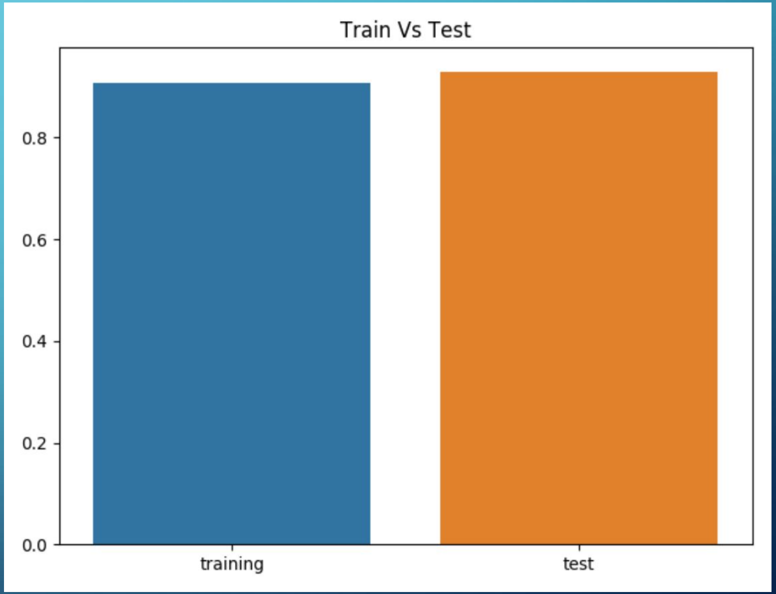
```
from sklearn.naive_bayes import GaussianNB

#Create a Gaussian Classifier
gnb = GaussianNB()

#Train the model using the training sets
gnb.fit(X_train, y_train)

#Predict the response for test dataset
y_pred = gnb.predict(X_test)
```

Confusion Matrix

Accuracy

Training - 90%, Test - 93%





Naive Bayes Algorithm

Pros:

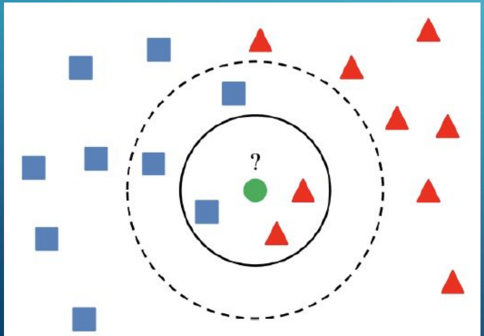
- It is not only a simple approach but also a fast and accurate method for prediction.
- Naive Bayes has very low computation cost.
- It can efficiently work on a large dataset.
- It is easy and fast to predict class of test data set.

Cons:

- One of the limitations of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

KNN Algorithm

- ❖ K nearest neighbor - supervised machine learning algorithm.
- ❖ calculates the distance of a new data point to all other training data points
- ❖ selects the K-nearest data points, where K can be any integer
- ❖ assigns the data point to the class to which the majority of the K data points belong.

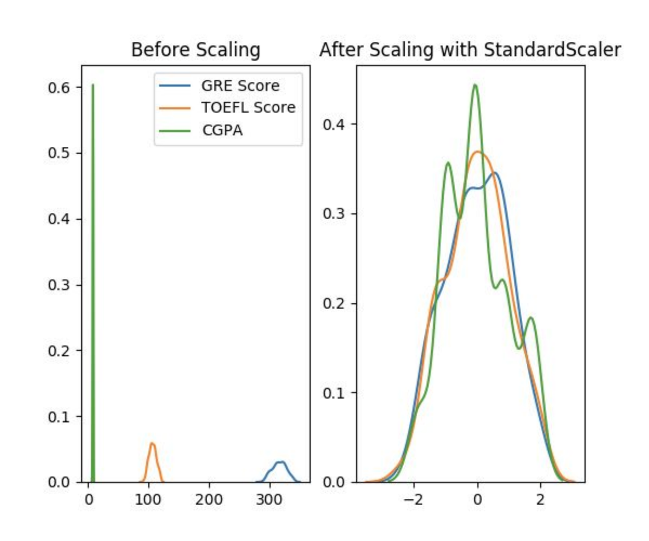


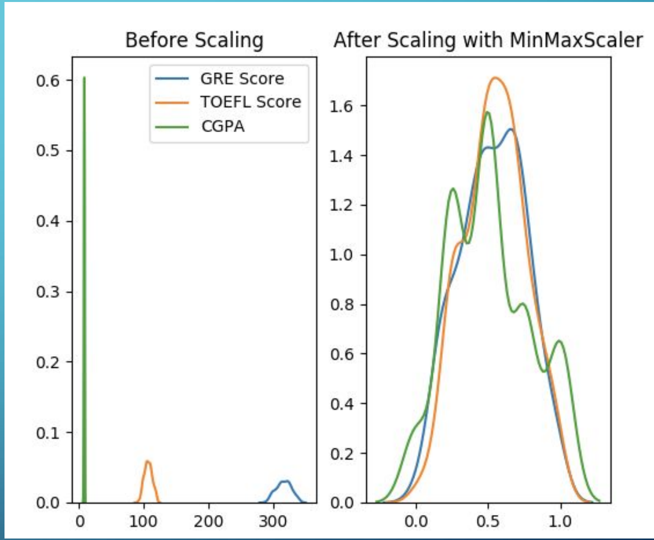

```
knn = neighbors.KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)
```

Scaling the data

- ❖ Dataset will contain features highly varying in magnitudes.
- ❖ Features with high magnitudes will weigh in a lot more in the distance calculations than features with low magnitudes
- ❖ KNN computes distance. Hence we need to bring all features to the same level of magnitudes.
- ❖ Common methods of scaling: MinMax Scaler, Standard Scaler, Normalization.

Scaling Data

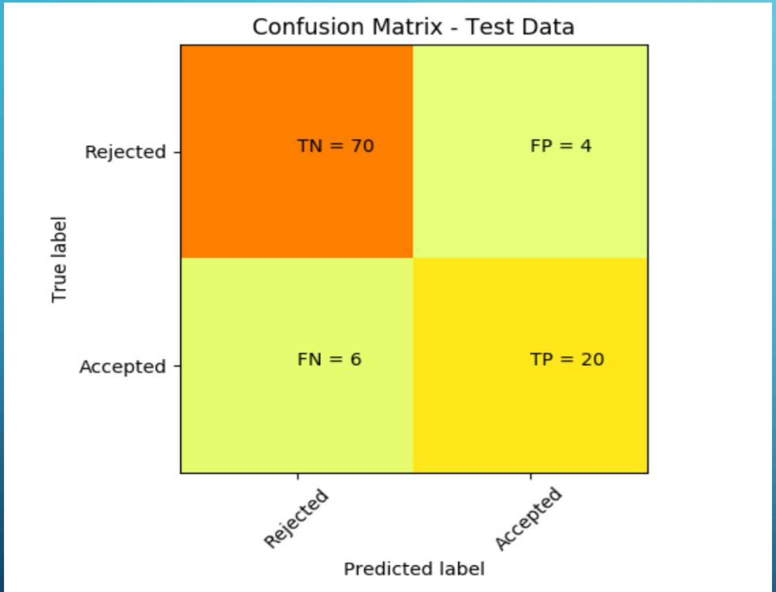




$X' = X - \overline{X} / \sigma$

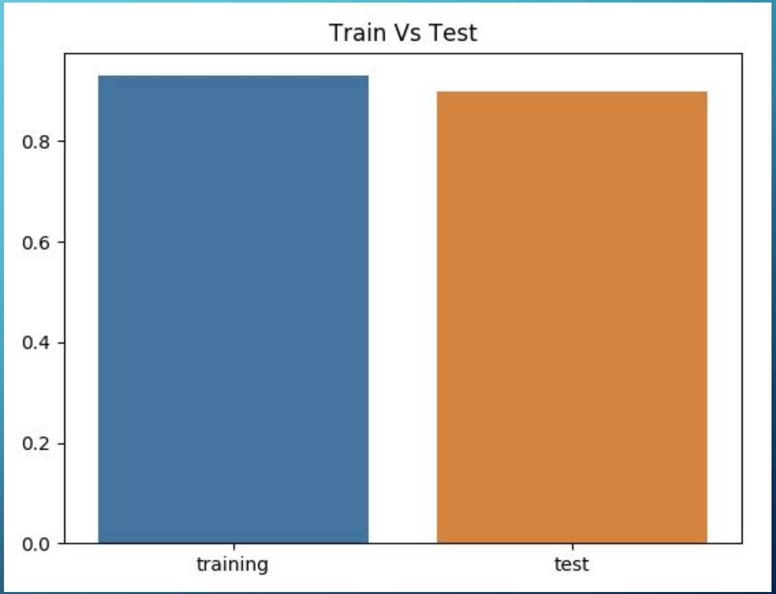
$X' = x - \min(x) / \max(x) - \min(x)$

Confusion Matrix



Accuracy

Training- 93%, Test - 90%



Classification Matrix - Report

		Predict		
		Positive	Negative	
Actual	Positive	20	6	26
	Negative	4	70	74
		24	76	100

True Positive Rate=
Sensitivity = TP / TP + FP
= 20/26 = **0.75**

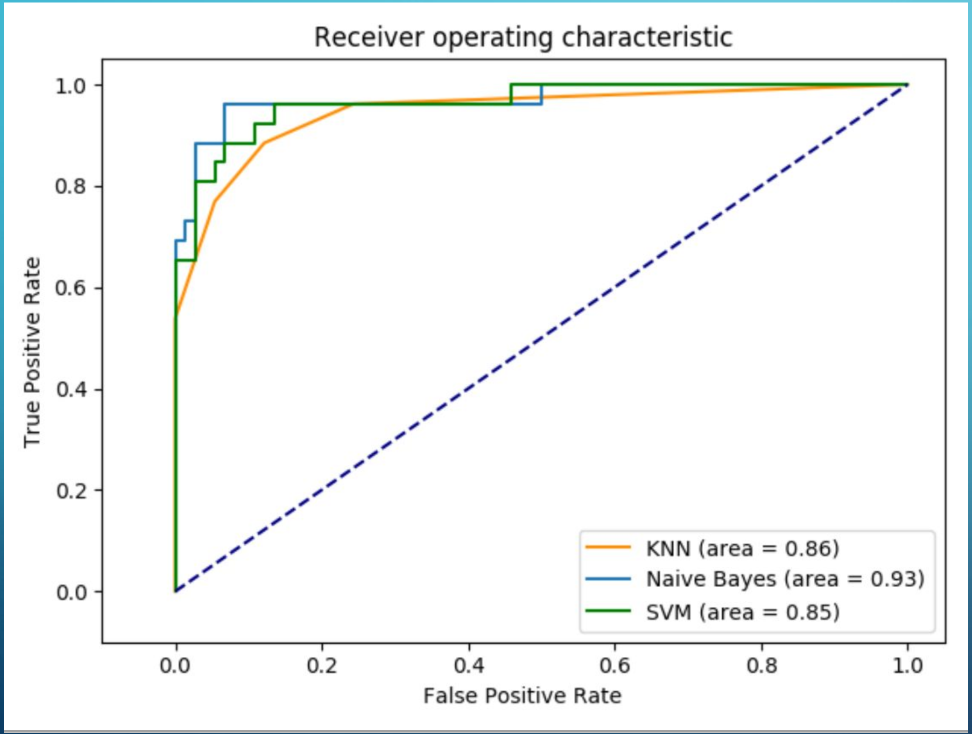
Specificity = TN / TN+FP
= 70/74 = **0.95**

False Positive rate = FP / FP+ TN
= 4/ 74 = **0.05**
= 1- Specificity

Accuracy = 70 + 20/ 100
= **0.9**

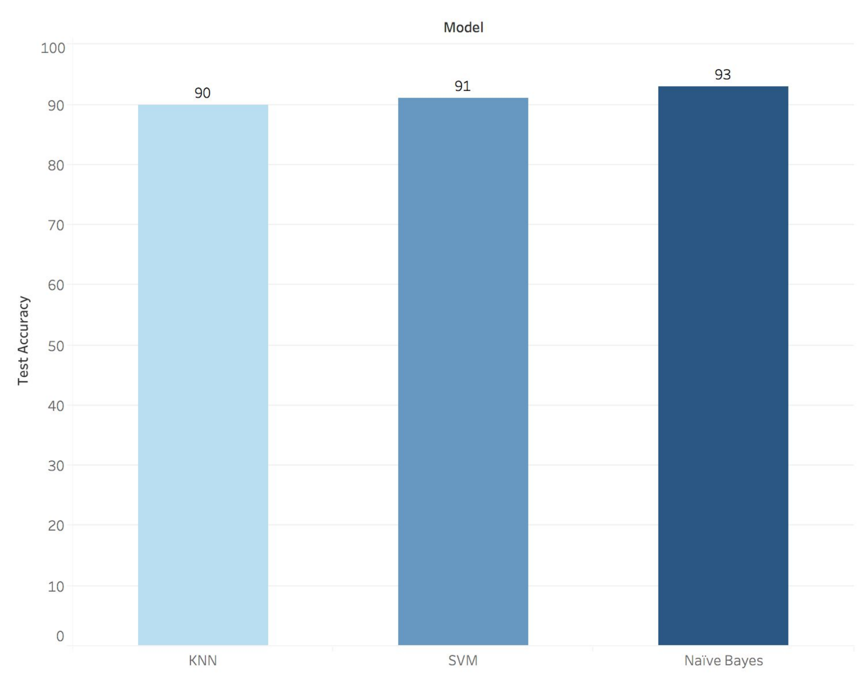
Positive Predictive Value = 20/24 = 0.83
Negative Predictive Value= 70/76 = 0.92

COMPARISON ANALYSIS



- As per the ROC chart we can see Naive Bayes behaves better than other models
- Most inclination towards 1.
- Highest ROC curve area

Test accuracy scores



Thank you..