MACHINE LEARNING MODEL ON HEART DISEASE DATASET

- AN ANALYTICAL STUDY -

TEAM MEMBERS:
RAUNAK SAHA FOUZDER(10900120170)
RUDRA NARAYAN ROY (10900120013)

TABLE OF CONTENTS

INTRODUCTION

- Cardiovascular diseases claim 17.9 million lives per year
- Countless patients swarm the limited available healthcare clinics which results in slow diagnosis and immense pressure on healthcare workers
- Need of an accurate and efficient way to diagnose heart diseases.
- An efficient machine learning model will be able to speedily and accurately diagnose the patient based on given data, hence saving time, effort and reduces pressure on health care workers.

PROBLEM DESCRIPTION (OUTLINE)

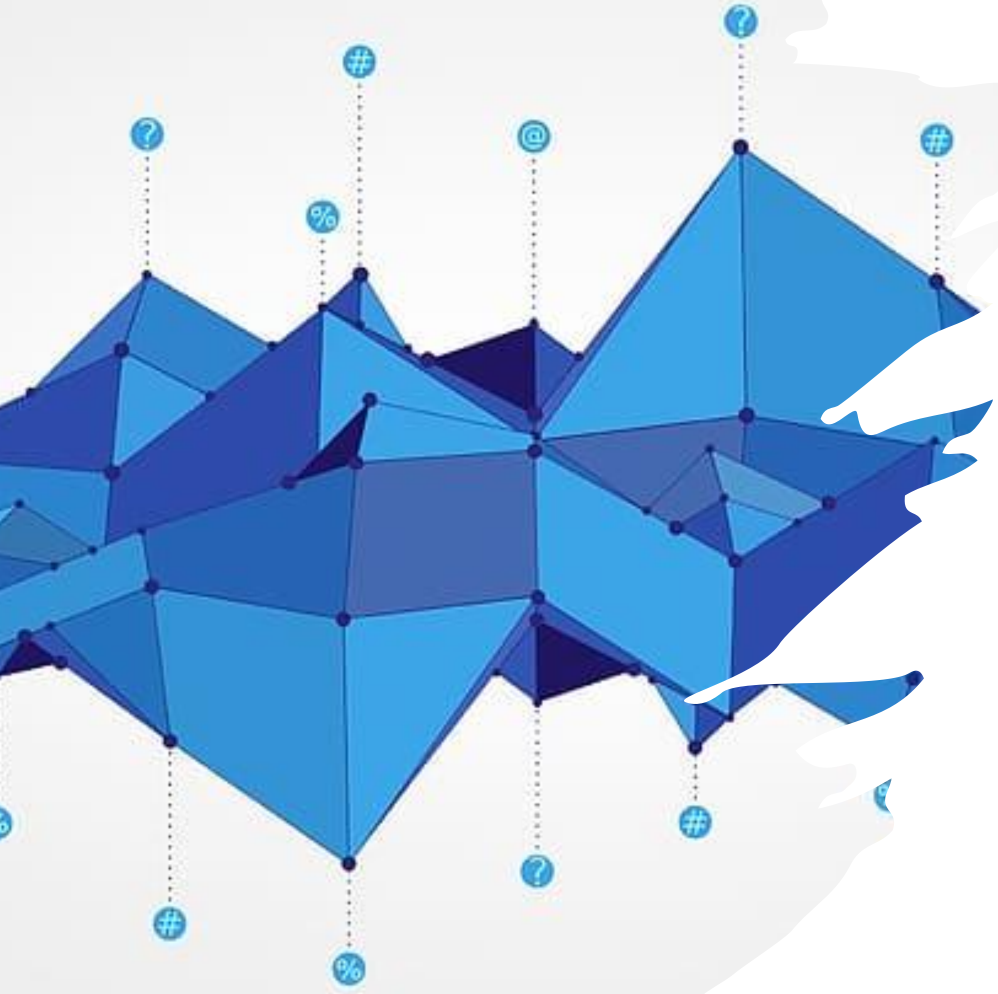
Increasing of Recall to reduce False Negatives(FN)

OUR APPROACH :- Clean the dataset of any unwanted data, split the available data for training and testing of the models, try various classical ML models and choose the best woking model.

Data Classification:- 1 (positive for heart disease) and 0 (negative for heart disease)

The dataset used in training and testing the model contains various parameters which are suspected to show effects in presence of a heart disease.

DATA DESCRIPTION



```
[3]: df=pd.read_csv("C:/Users/pratyasha das/Documents/PythonProjectsPD/DATASETS/heart_disease_data.csv")
df.head()
```

[3]:

	age	sex	chestpain	restbps	cholestrol	fastingbs	restecg	maxheartrate	exang	oldpeak	slope	ca	thalassemia	target	heart_disease
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1	0
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1	0
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1	0
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1	0
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1	0

Total columns=**15**

Total rows = 303

All **numeric** columns. No string columns. (**Discrete** columns= 10,Continuous columns=5)

DISCRETE COLUMNS :- sex, cp, fbs, recg, exang, slope, ca, thal(one **hot encoding**)

SEPARATOR = **“,”**

ANOMALY= **2 complimentary column(target,heart_disease)**

Header **present. Renaming needed**

Null **value percentage=0**

UPDATED DATA DESCRIPTION

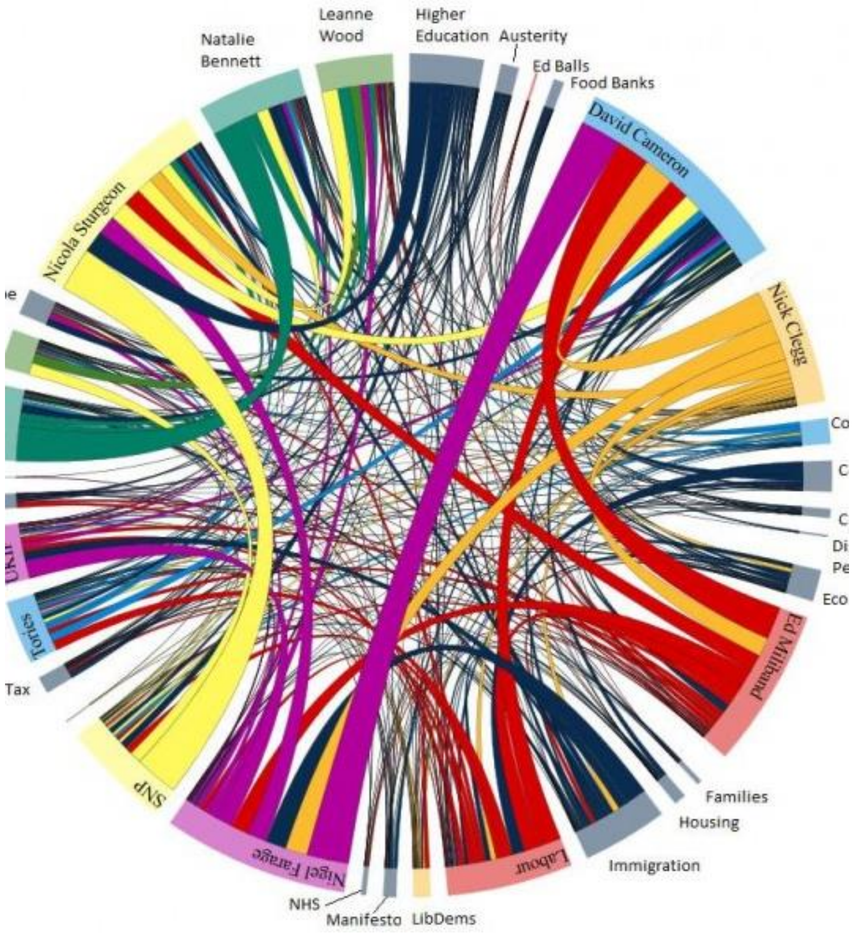
One Hot Encoded ***data description*** *: 303* ***rows X 23 columns***

Columns *removed in* ***improved model due to impurity:***
slope_2,ca_4,thal_*1,fbs****_1,recg_2***

Important Features : 17 columns

Columns included :- *Continuous(age****, rbps, chol, maxhr, oldpeak)****, Discrete(****sex_1, cp_1, cp_2, recg_1, exang_1, slope_1****,* ***slope_2, ca_1, ca_2, ca_3, thal_2, thal_3****)*

DATA CLEANING & MODEL BUILDING



- ***DATA CLEANING***

Removal of column(heart_disease) from dataset

- ***MODEL BUILDING***
  - **LOGISTIC REGRESSION**: Base Model(Training-89.78%,Test-85.71%) , Improved Model(Training-83.94%,Test-75%)
  - **DECISION TREE**: Base Model(Training-100%,Test-71.42%) , Improved Model(Training-90.51%,Test-78.57%)
  - **RANDOM FOREST**: Base Model(Training-100%,Test-78.57%) , Improved Model(Training-81.75%,Test-78.57%)
  - **KNN**: Base Model(Training-86.86%,Test-85.71%) , Improved Model(Training-88.32%,Test-82.14%)
- **NAIVE BAYES**: Base Model(Training :- 51.82%, Test 35.71%) , Improved Model(Training-83.94%,Test-75%)

FINAL MODEL

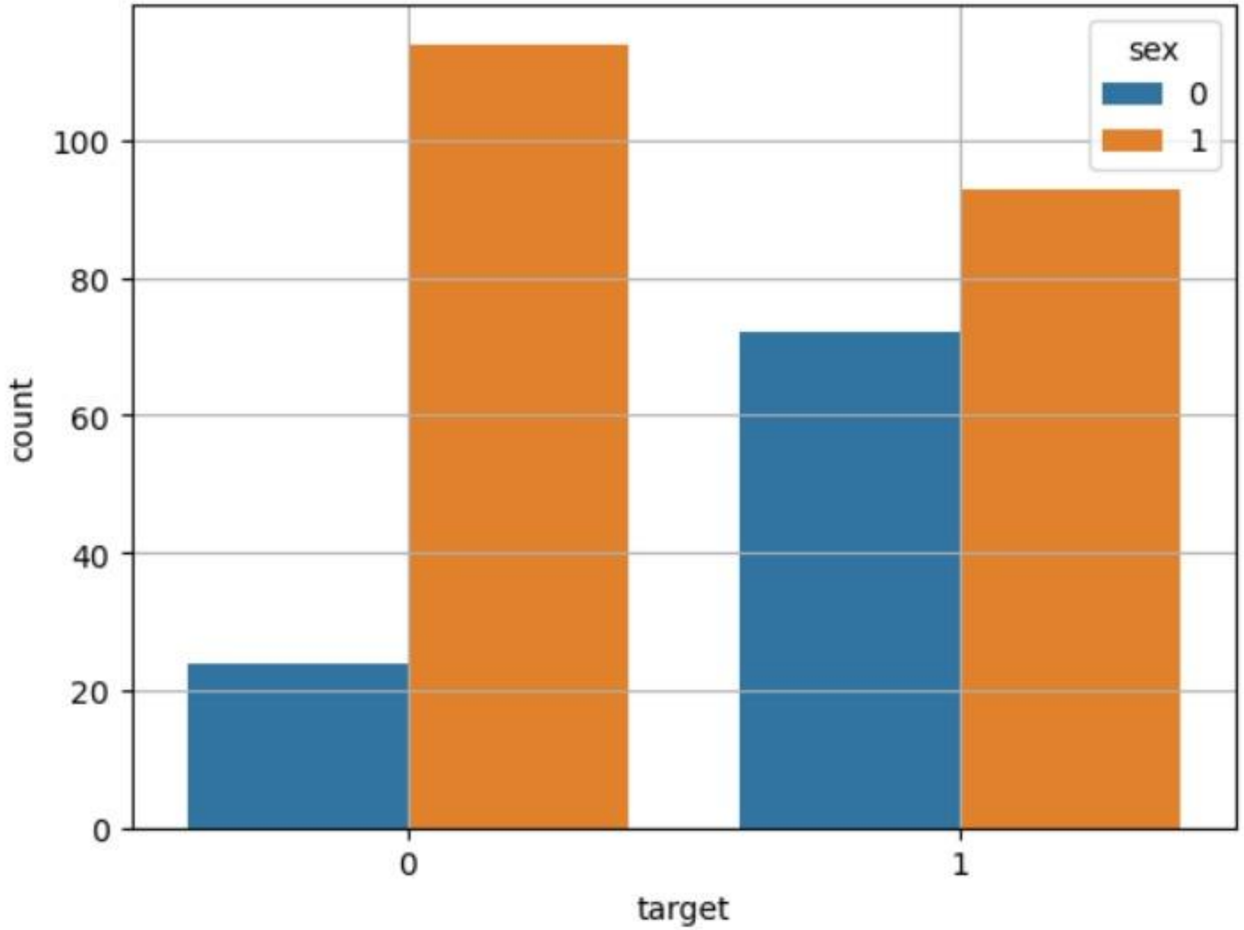
```
bestmodel=gobj.best_estimator_
predtest=bestmodel.predict(Xtest1)
predtrain=bestmodel.predict(Xtrain1)

print("TRAINING MERICS:")
printscores(ytrain,predtrain)
print("=================")
print("TEST METRICS:")
printscores(ytest,predtest)
```

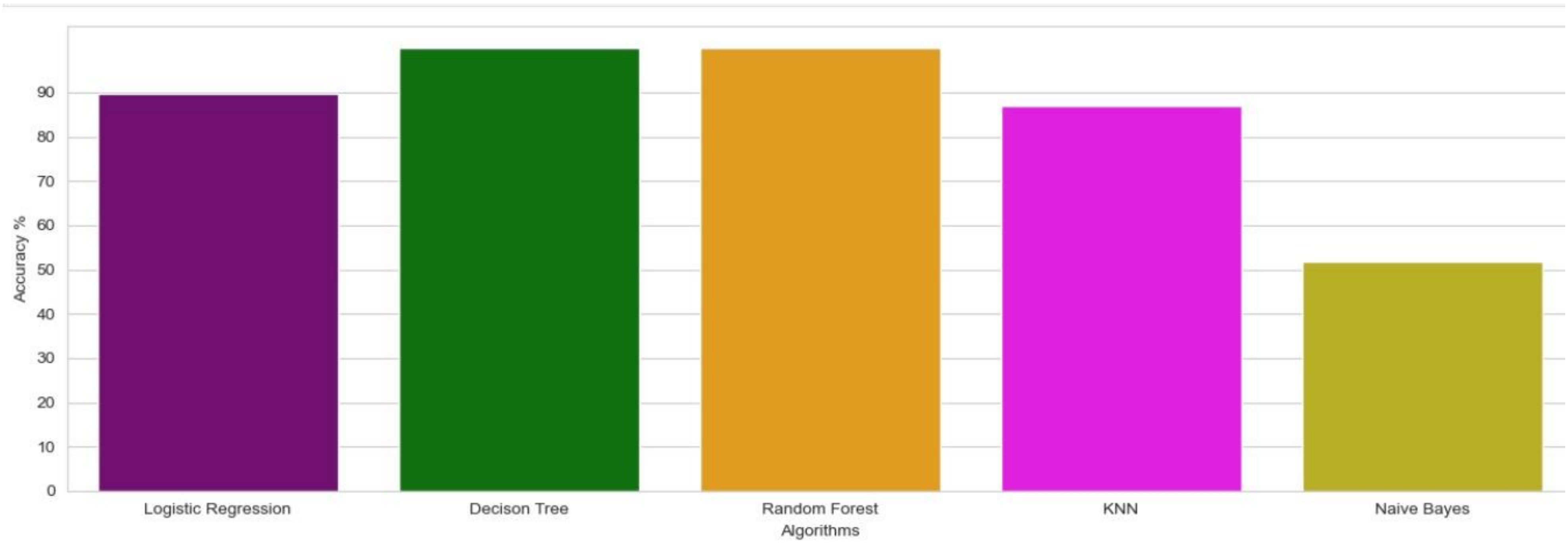
```
TRAINING MERICS:
accuracy : 77.431906614786 %
recall : 81.75182481751825 %
precision : 77.24137931034483 %
f1 : 79.43262411347517 %
AUC : 77.12591240875912 %
=================
TEST METRICS:
accuracy : 82.6086956521739 %
recall : 78.57142857142857 %
precision : 91.66666666666666 %
f1 : 84.61538461538461 %
AUC : 83.7301587301587 %
```

- **RANDOM FOREST MODEL**
- *Why?*
- Stable Model
- Satisfactory Recall in training and test values.
- Close f1 and accuracy for both training and test values.

EXPLORATORY DATA ANALYSIS



EXPLORATORY DATA ANALYSIS



FUTURE IMPROVEMENTS

- Development of more efficient algorithms that can handle big data and scale with increasing sample size
- Technique to make boosting algorithm robust to outlier would improve the performance
- Improvement in computational efficiency of boosting algorithm make them more accessible
- Automated parameter tuning
- Enhancing the interpretability of boosting models can help users gain insights into the decision-making process and build more trust in the models.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Prof. Piyali Chatterjee for her invaluable support and guidance throughout our machine learning project on classifying heart disease. The resources and mentorship provided by ma'am have played a crucial role in the successful completion of this endeavor. I would also like to thank my team member for his unwavering support and encouragement. Without their contributions, this project would not have been possible.