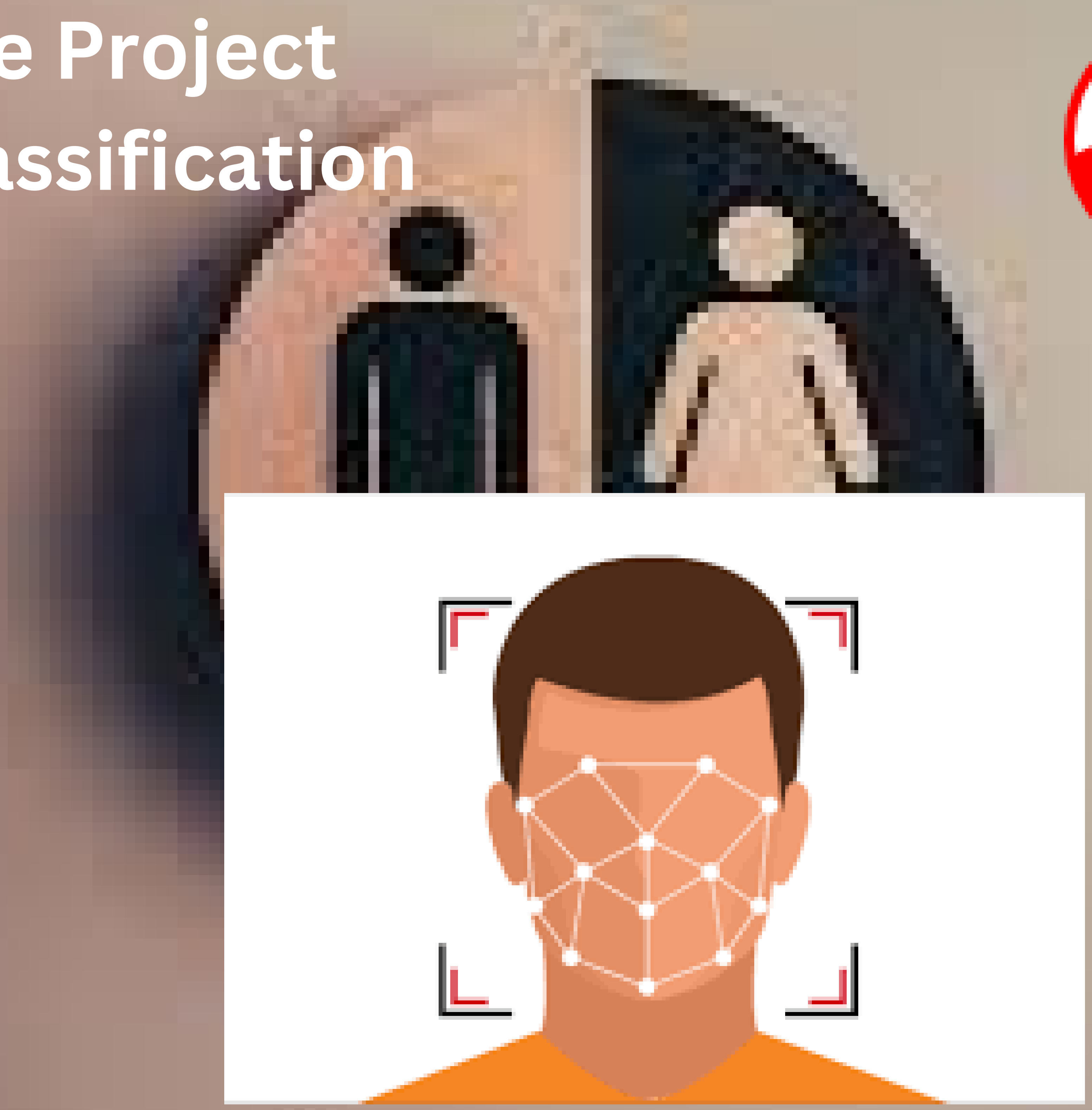
Bhavan's Vivekananda College
Sainikpuri
Autonomous College

Capstone Project
Gender Classification



Group-6
Team Members

1.N.Bhavana Reddy
2.UudhhayKiirran
3.Jashwanth
4.SaiPrasanna

Bhavan's Vivekananda College of Science, Humanities & Commerce ,   BSC.Honors Data Science

Introduction

DataSet-Gender Classification

- A survey was conducted on 5002 participants by researches on their different facial features and determine the relation between the gender and facial features.

- Objective: To predict ‘Gender’ using the different features of a person in the dataset

Technical contents:

- Data importing
- Data exploration
- Data Preprocessing
- Data Modelling

Data Importing

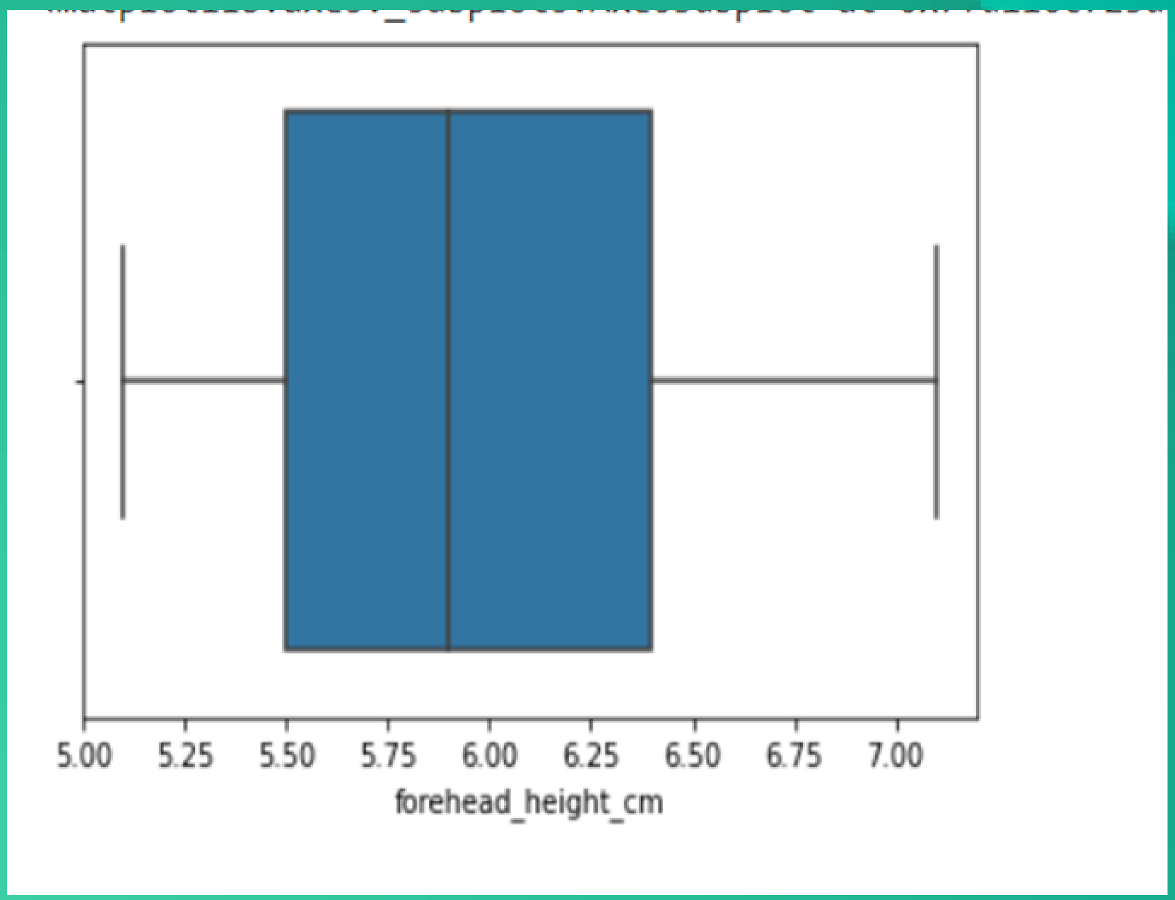
	long_hair	forehead_width_cm	forehead_height_cm	nose_wide	nose_long	lips_thin	distance_nose_to_lip_long	gender
0	1	11.8	6.1	1	0	1	1	Male
1	0	14.0	5.4	0	0	1	0	Female
2	0	11.8	6.3	1	1	1	1	Male
3	0	14.4	6.1	0	1	1	1	Male
4	1	13.5	5.9	0	0	0	0	Female

- The data contains 8 columns of which 7 are independent variable columns and 'Gender' is the target variable.

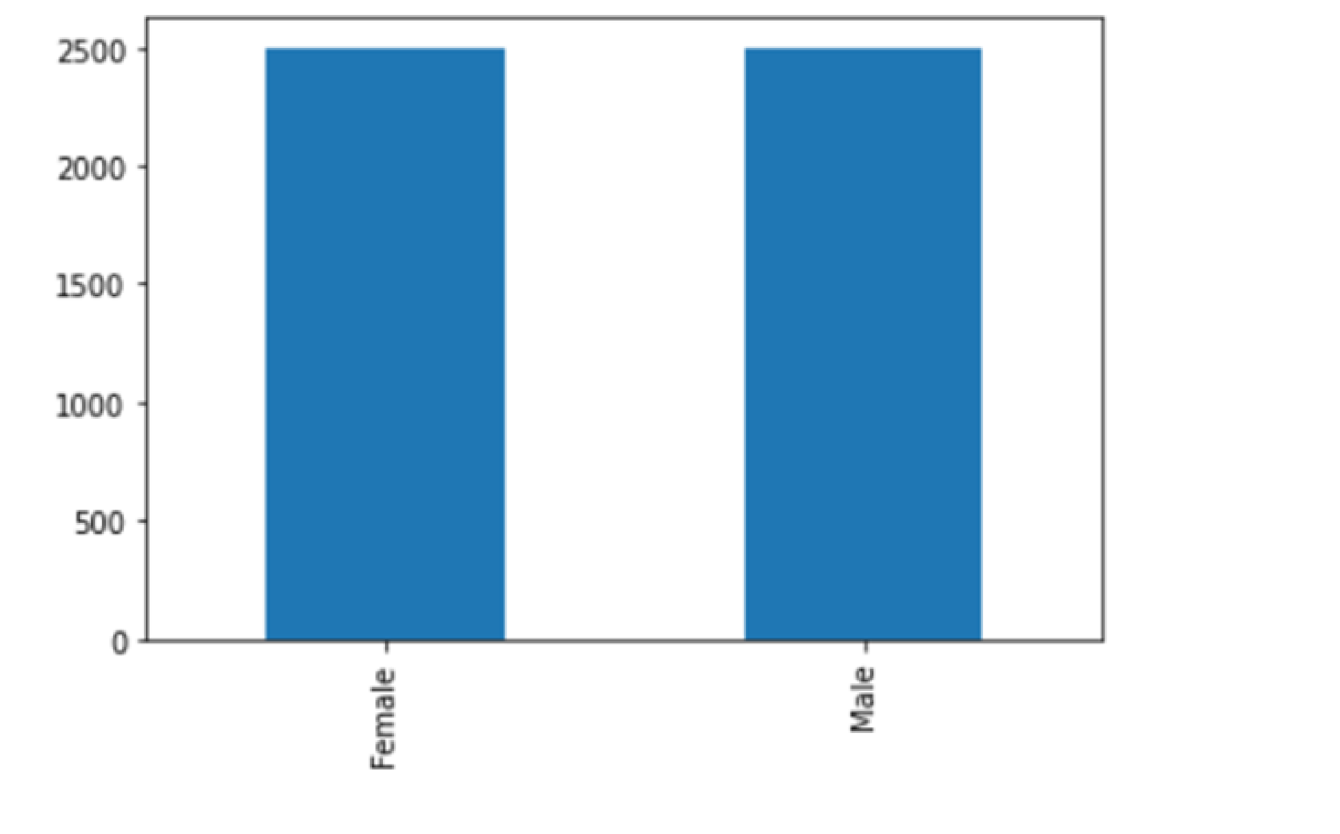
Variable description

- forehead_height,forehead_width are continous values and the other 5 independent columns have binary values.
- Long_hair-0-- no long hair,1--long hair.
- nose_wide-0--not wide,1--nose wide
- nose_long-0--not long,1--nose is long
- lips_thin-0--thin lips,1--not thin lips
- distance from nose to lip long-0--short distance,1--long distance

Data Cleansing

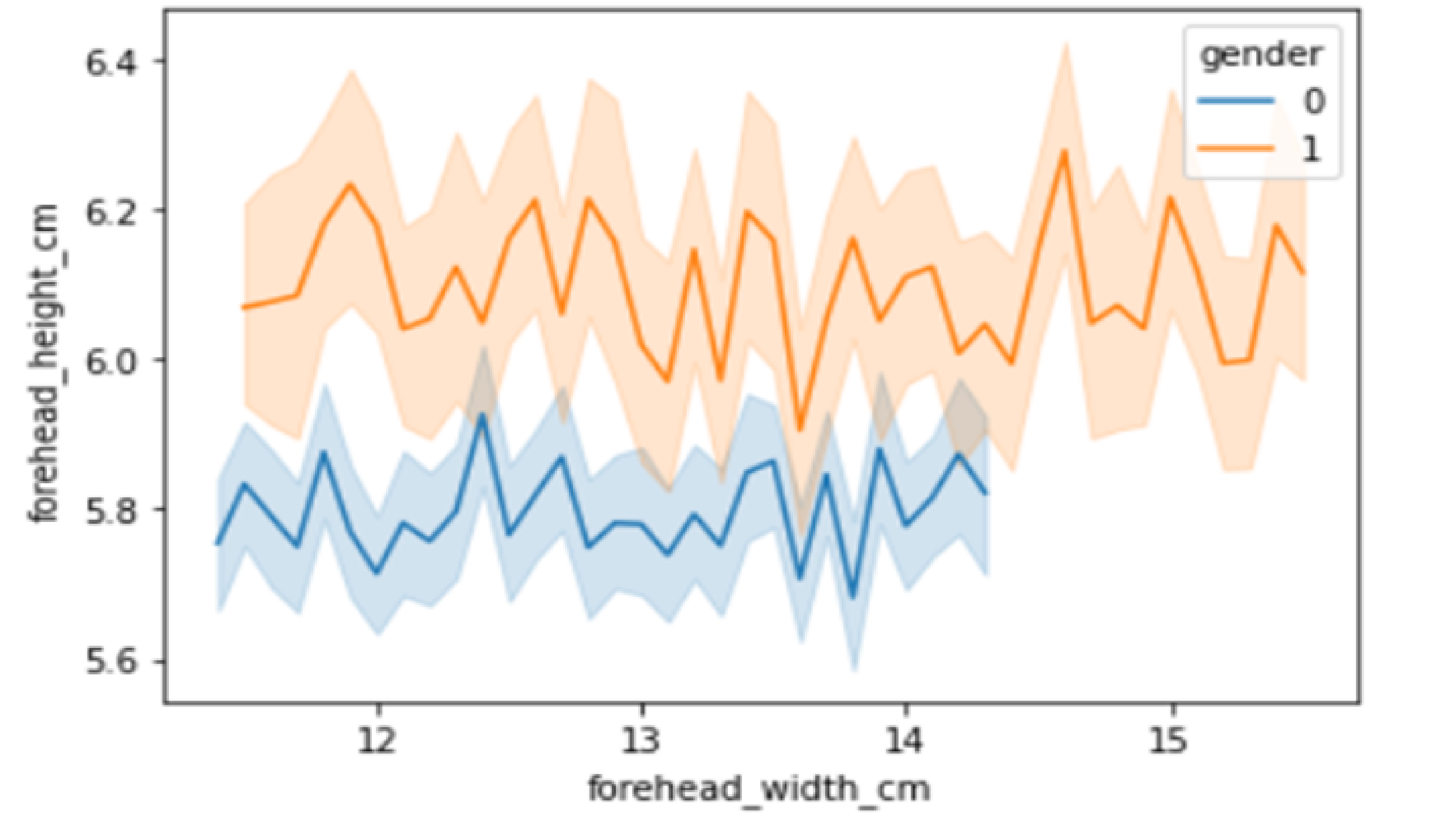


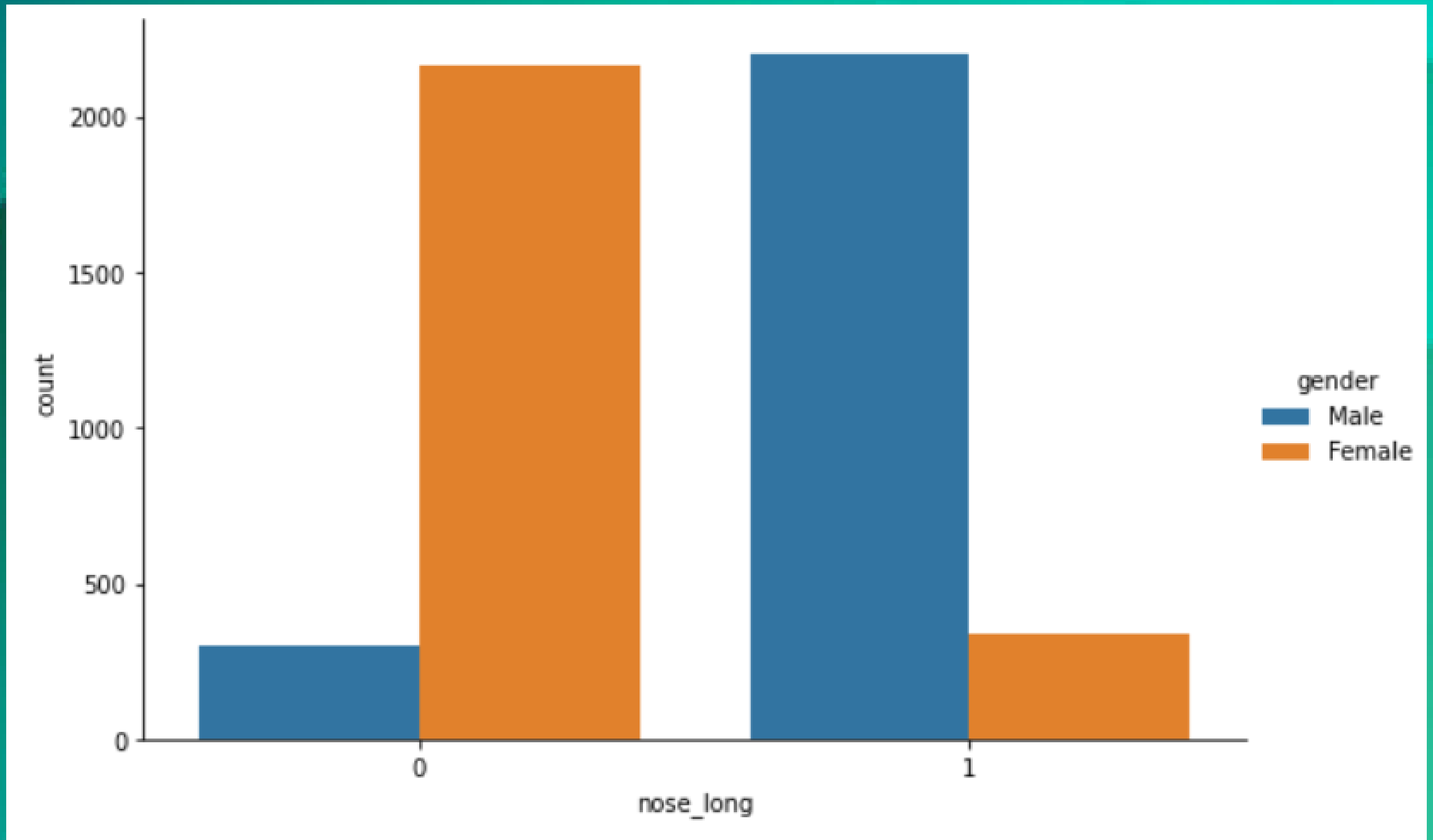
Exploratory Data Analysis

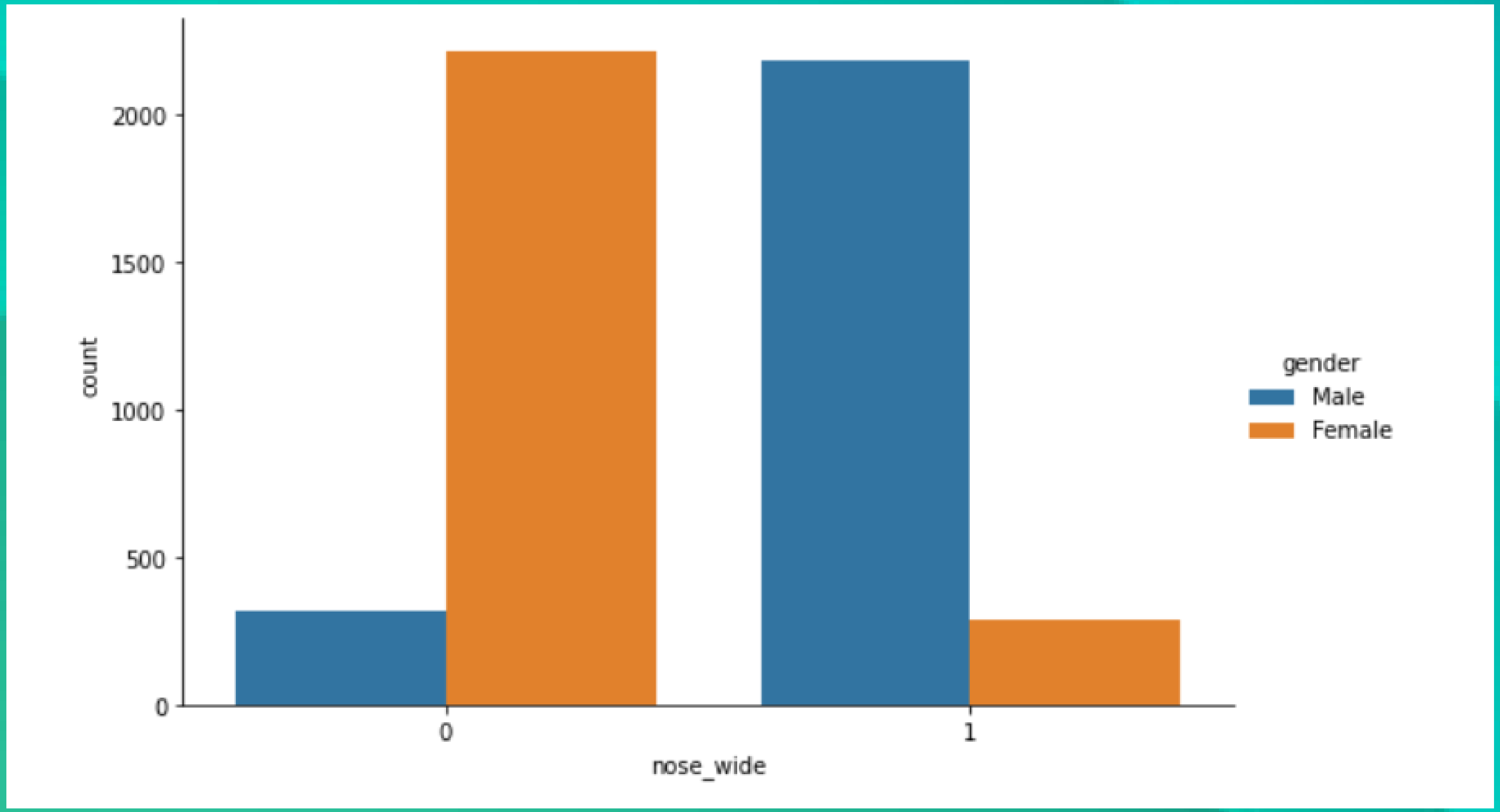


This bar plot depicts the number of male and female.

This graph describe the longer and wider the forehead, the more likely it is a man



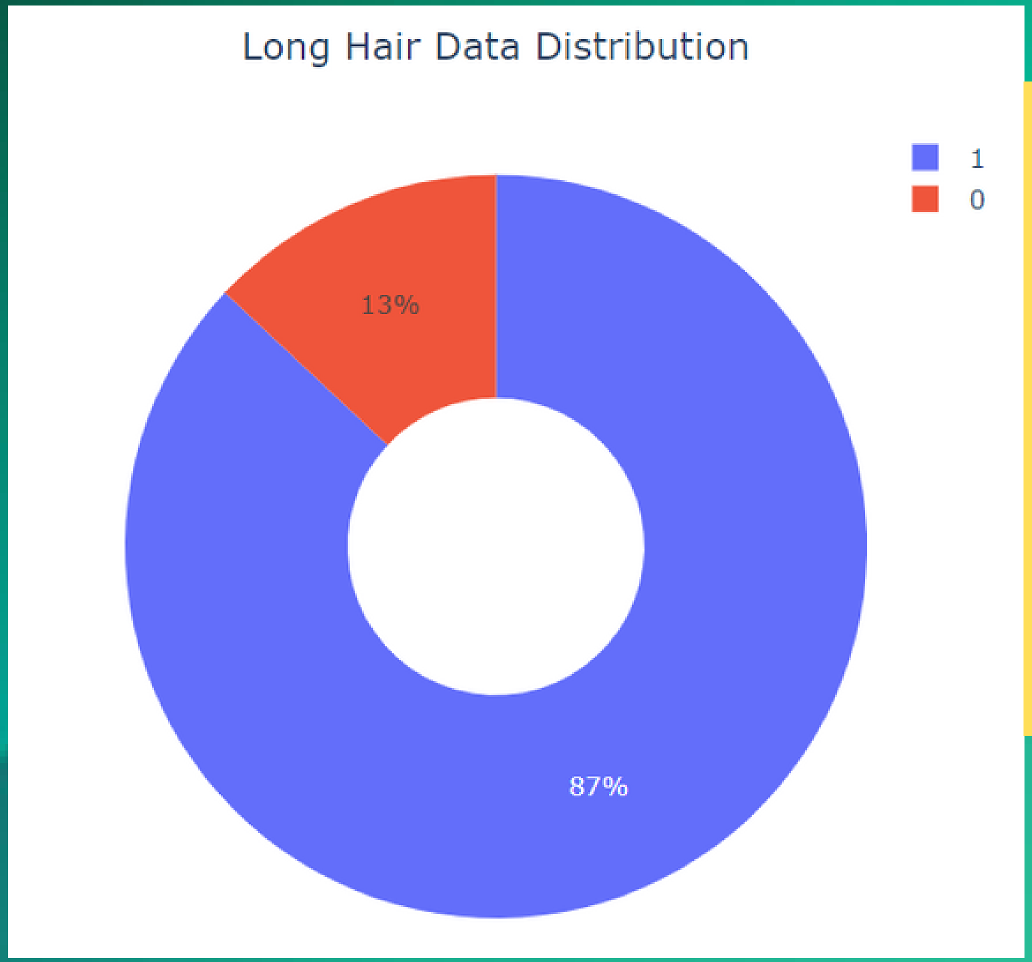




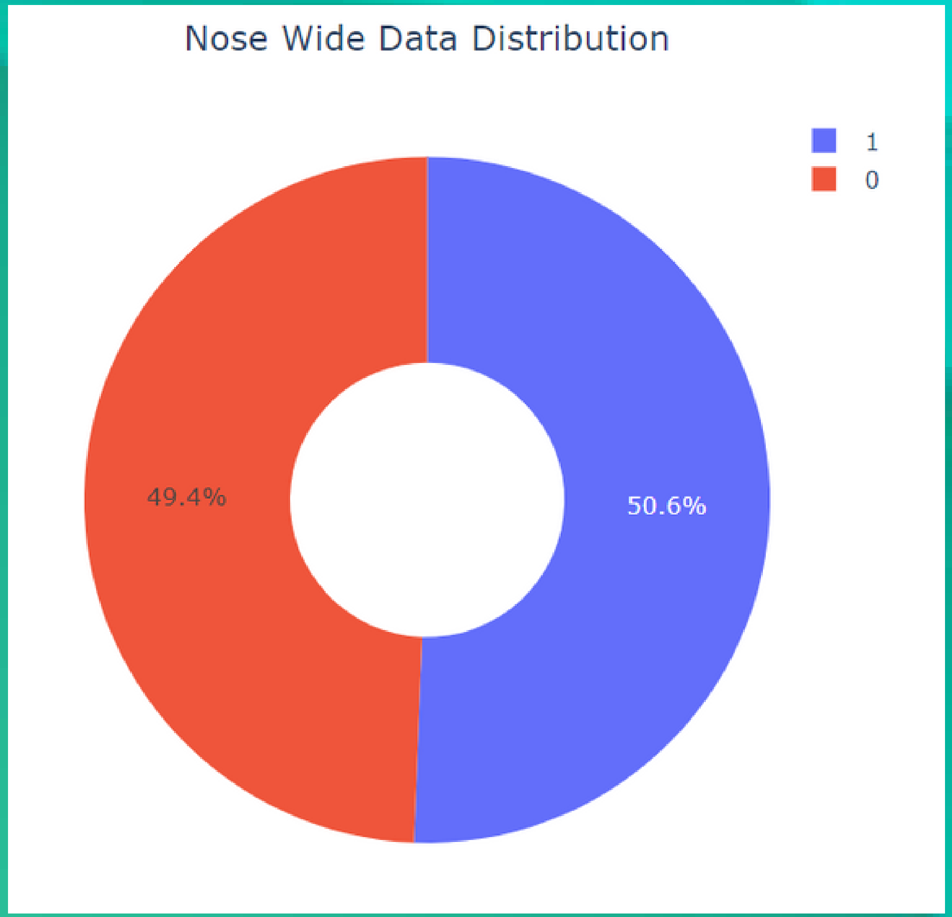
In this catplot with respect to nose wide we can observe more male are having nose wide when compared to female.

Here we can observe very less female are having nose long when compared to male.

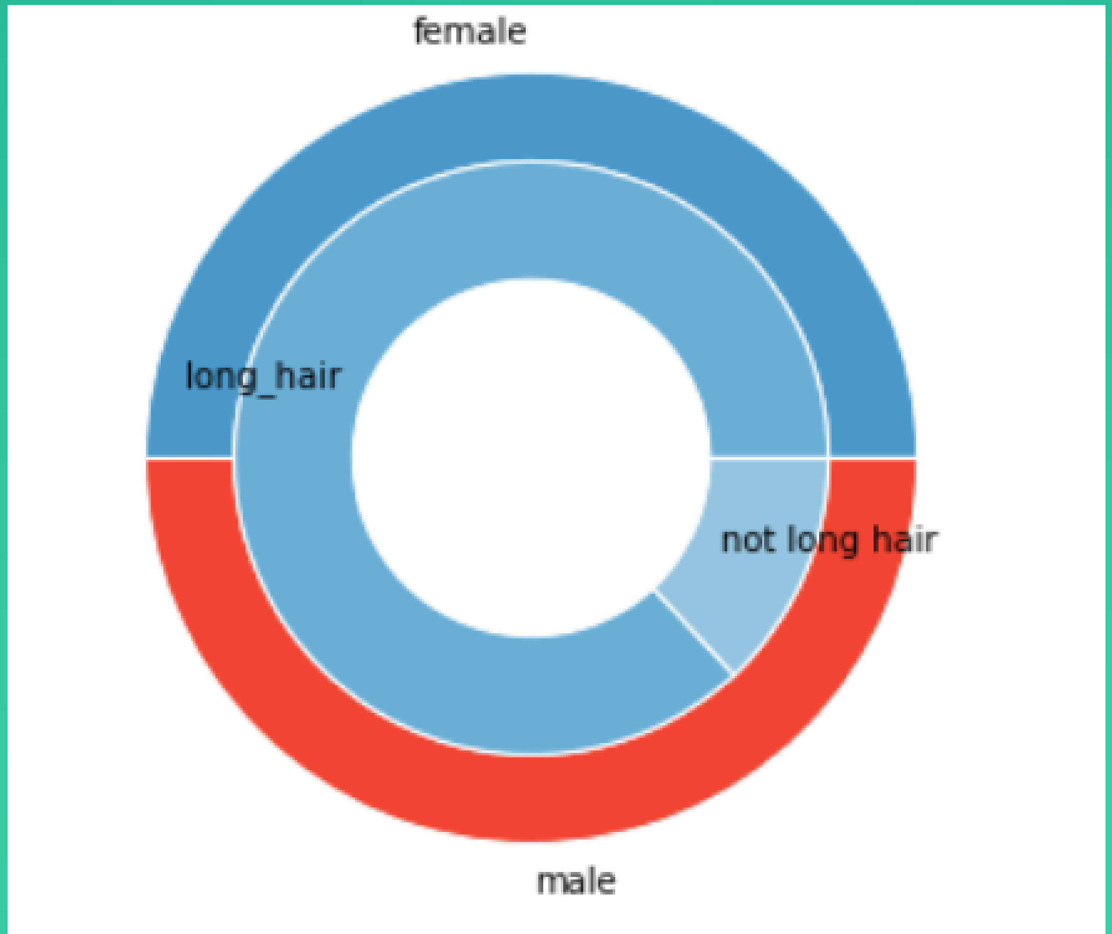
Exploratory Data Analysis



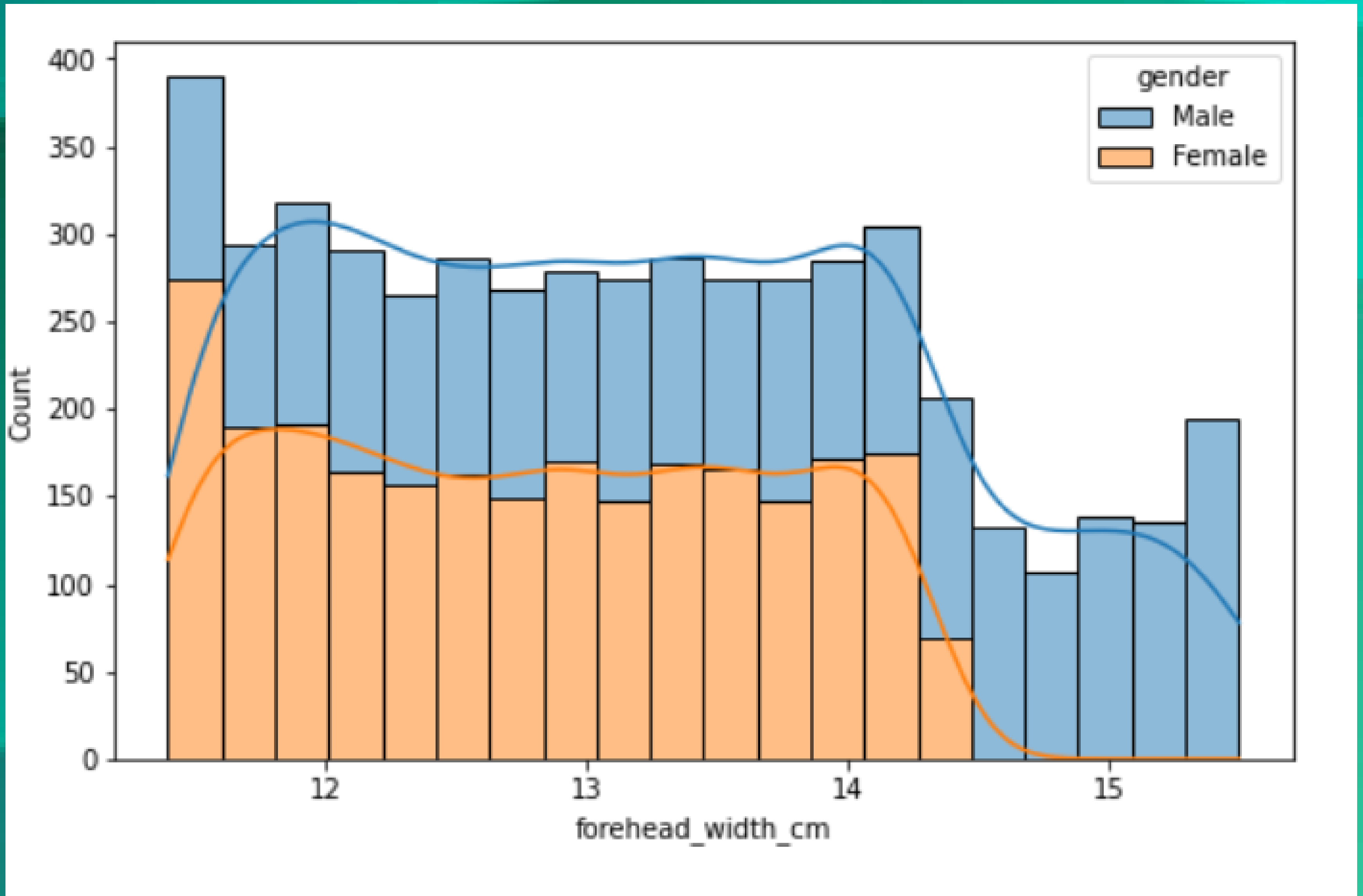
- Here are the donut charts depicting the percentage of people having long hair and nose wide.
- So from th long hair distribution we can say most people of both male and female are having long hair.
- In nose wide distribution chart the percentage of people having nose wide and no nose wide is same

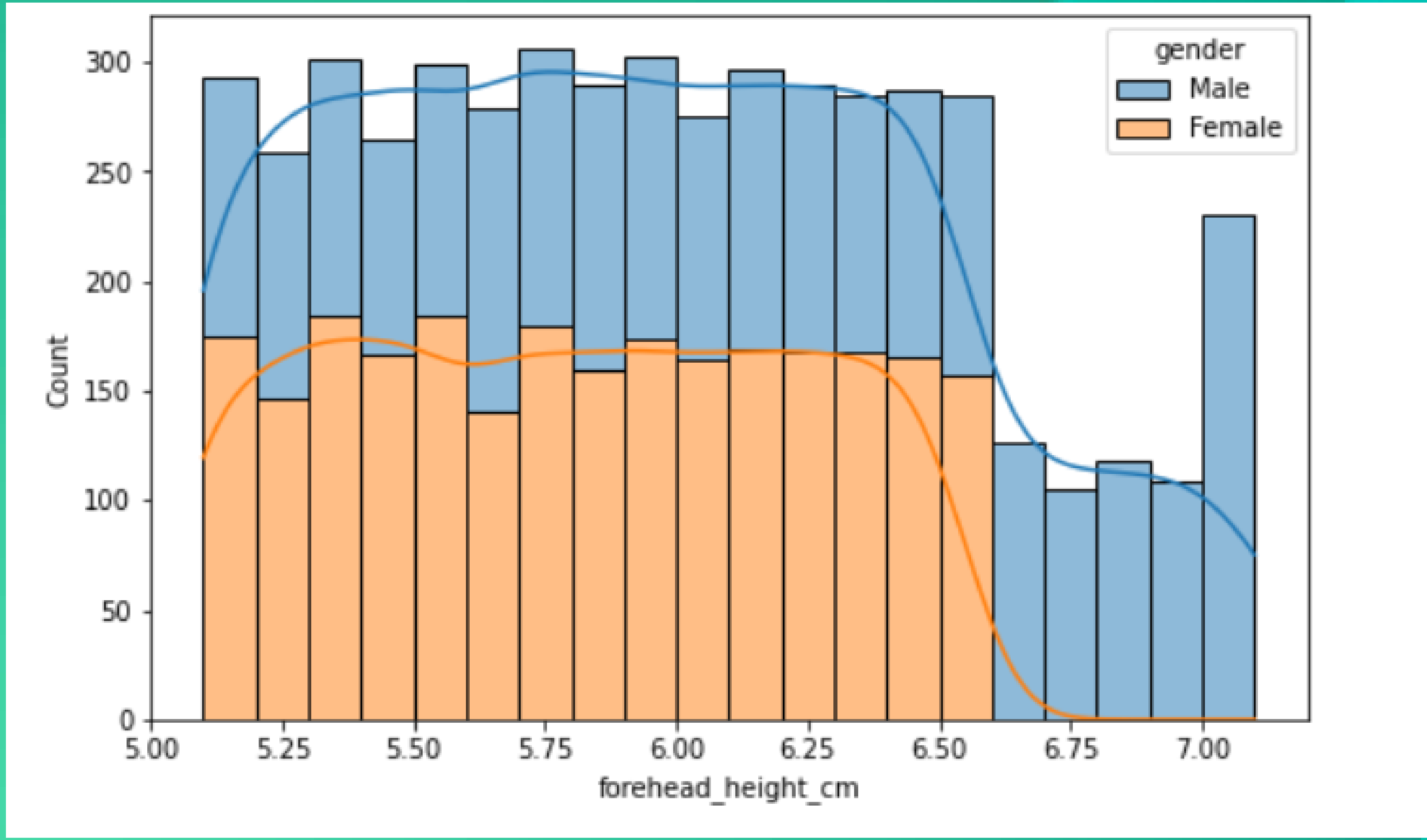


This is a stacked donut chart representing th long hair feature with respect to Gender



RELATION WITH REPECT TO GENDER





Data Transformation

	long_hair	forehead_width_cm	forehead_height_cm	nose_wide	nose_long	lips_thin	distance_nose_to_lip_long	gender
0	1	11.8	6.1	1	0	1	1	1
1	0	14.0	5.4	0	0	1	0	0
2	0	11.8	6.3	1	1	1	1	1
3	0	14.4	6.1	0	1	1	1	1

- We have our dependent column as categorical so we converted the female and male to binary values i.e 0's and 1's.

ModelFitting

Logistic Classification

- We have splitted the train and test ratios.
- We have trained the model.
- We tested the data and predicted the values .

Train-Test Proportion	Accuracy
80-20	97
70-30	96.85
60-40	96

Best ratio-80-20

Neural Networks

Sample of experiments

Train-Test Proportion	Architecture	Optimizer	epochs	Accuracy
70-30	5--2--1	Adam	500	0.9647
70-30	5--3--1	Adam	800	0.9574
70-30	5--3--1	SGD	800	0.9587
70-30	5--4--2--1	Adam	400	0.9607
70-30	5--4--2--1	SGD	400	0.9534
70-30	7--5--3--1	Adam	200	0.9594
70-30	5--2--1	SGD	400	0.9587
70-30	7--5--3--1	SGD	200	0.952
80--20	5--3--1	Adam	50	0.4845
80--20	5--2--1	Adam	500	0.968
80--20	5--3--1	SGD	800	0.5466
80--20	5--2--1	SGD	500	0.966
80-20	7--6--4--2--1	Adam	500	0.4885
80-20	7--6--4--2--1	SGD	500	0.963

- In the neural networks we are working with respect to two optimisers : Adam ,SGD
- So we applied it with various experiments and architectures

Maximum Accuracy -0.9680

Architecture-5--2--1

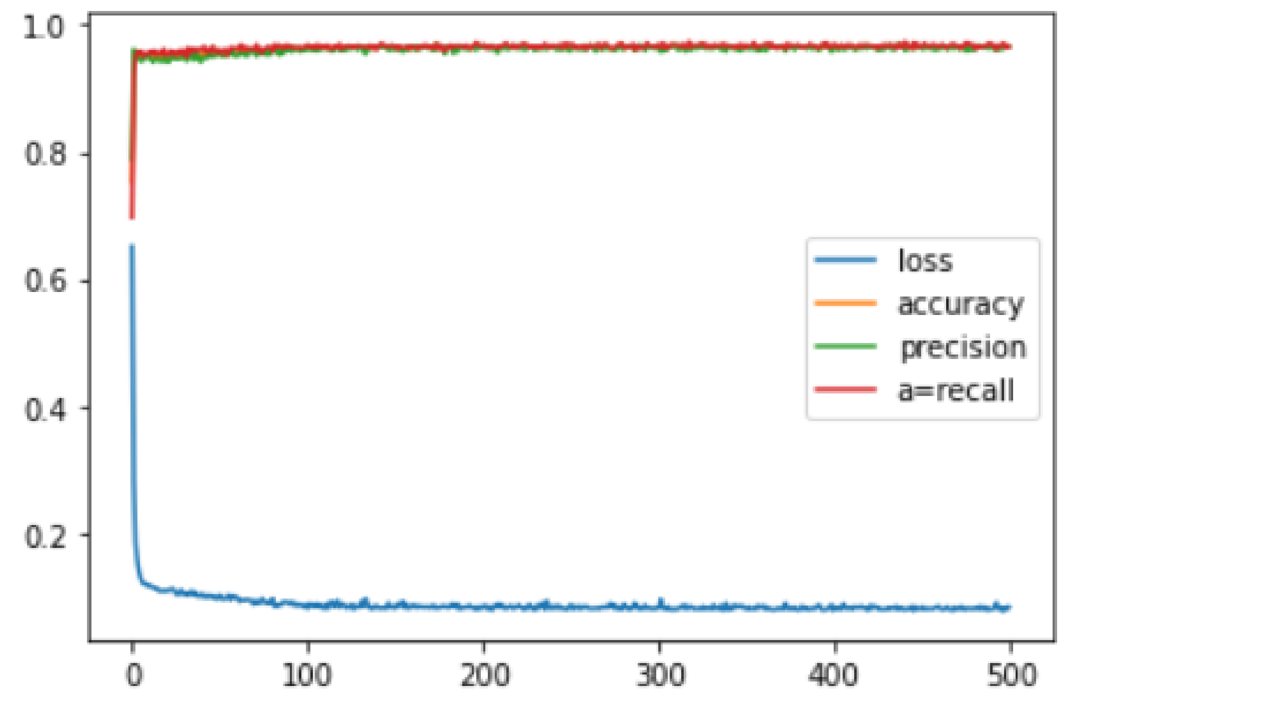
Optimiser-Adam

Epoch=500

Train test ratio-80-20

```
pd.DataFrame(history.history).plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2ada9f3850>
```



K-Nearest Neighbours

- The KNeighborsClassifier has some parameters to improve its performance.

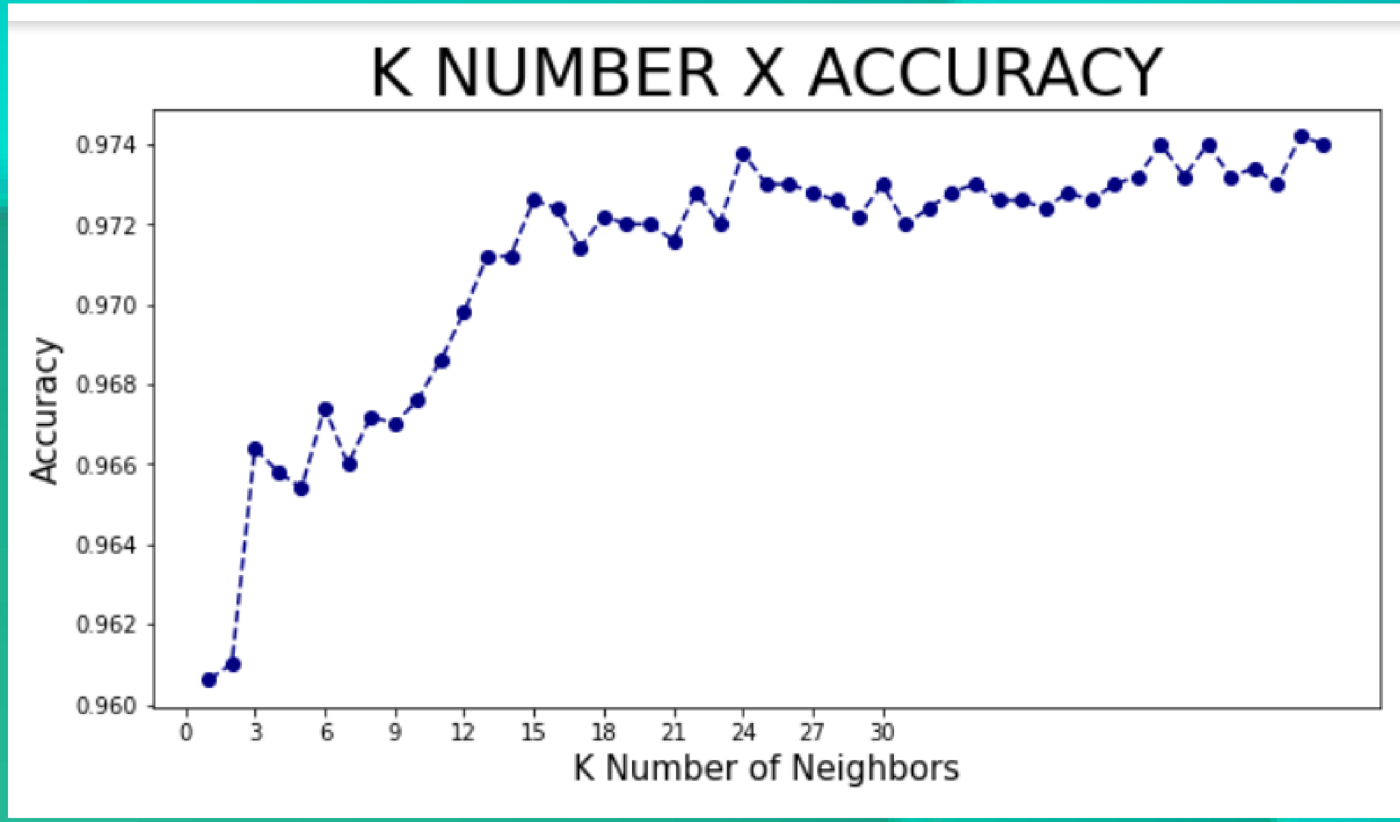
- At first only n_neighbors is going to be set, the others are to be as default. Later, an optimization analysis could be performed to adjust them.
  n_neighbors is set to be 3, what means it will take the gender classification to the average of three closest data.
- With the cross validation it is seen that this performance could vary from 95% to 100%.

- The performance of the classifier is considered as the average of the cross validation. In this case 96.6 (+/- 0.6).

Train-Test Proportion	Accuracy
80-20	0.96903
70-30	0.95869
60-40	0.96601

Grid seacrch

- This parameter is going to be optimized. Initially a list of possible k factors is created, from 1 to 50.
- we got 48 as the best neighbor with low standard deviation.It also has the lowest standard deviation among the ranked number 1.
- The rank depends only in the accuracy, as selected in the GridSearch.



- Here is the graph of K-numbers versus Accuracy:
- We observe as the values of the k-numbers increases the Accuracy is increasing.
- When the best value of K is 48 then the accuracy-0.974206

Support Vector Machine

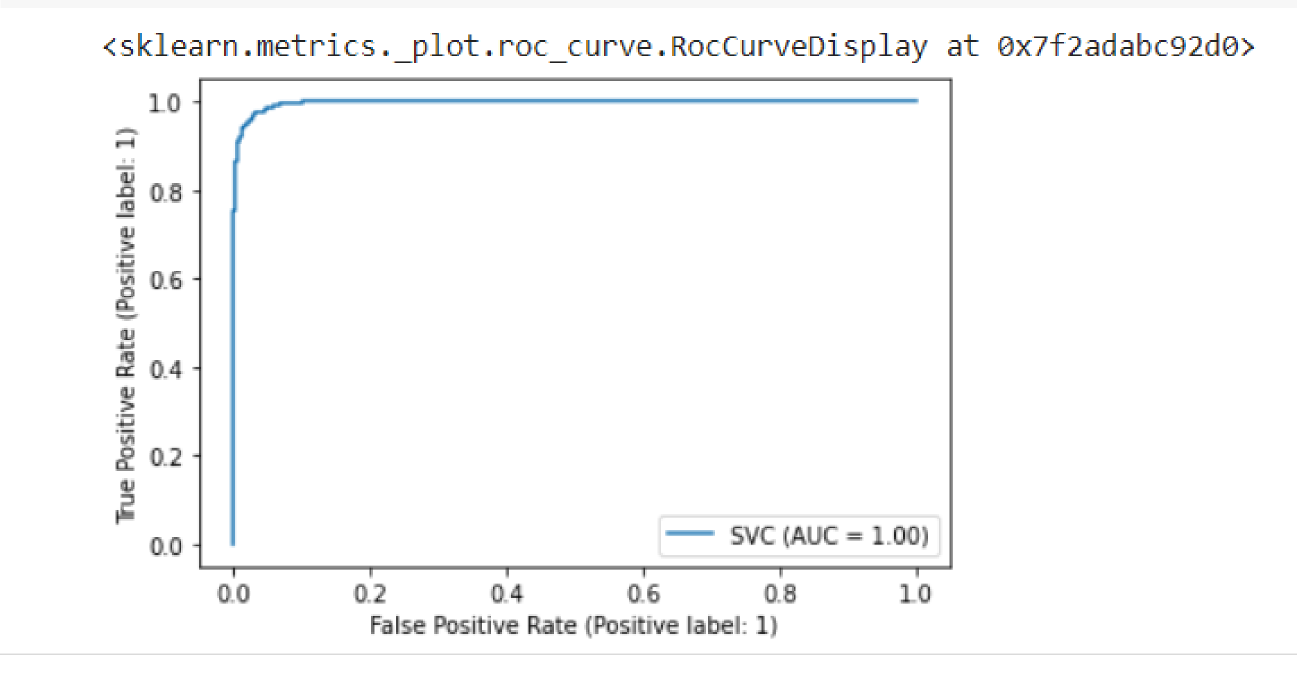
Kernel as Radial basis funtion

- SVM with respect to rbf as kernal.
- Trained the model
- Testesd the model
- Accuracy of SVC Classifier:: 0.97202

Train-Test Proportion	Accuracy with rbf	Accuracy with linear
80-20	0.97202	0.96803
70-30	0.97035	0.96688
60-40	0.97001	0.96701

Kernel as Linear function

- SVM with respect to Linear as kernel.
- Trained the model
- Testesd the model
- Confusion Matrix
- Accuracy = 0.96803



Bagging

Decision Tree

- Imported the decision tree classifier.
- Predicted values are of the form of array([1, 0, 0, ..., 1, 1, 1])

Train-Test Proportion	Accuracy
80-20	0.96
70-30	0.97
60-40	0.96

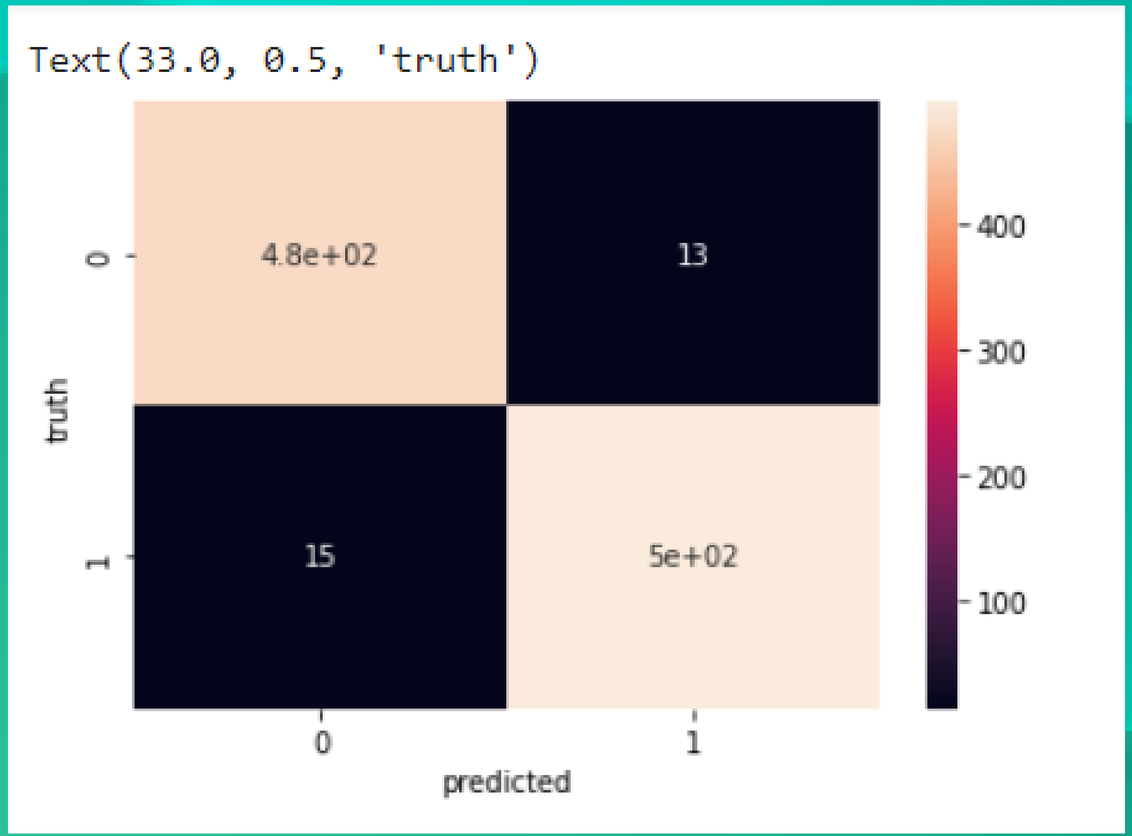
Accuracy 0.97
with 70-30

Decision treee with respect to Ginni Index

- Instantiate the DecisionTreeClassifier model with criterion gini index.
- fit the model with max depth 3 randomly
- Accuracy-0.9690

Random Forest

- Random forest method is an extension of bagging
- Imported the Random forest classifier.
- we gave N-estimators as 50 randomly
- Accuracy-0.97102



Train-Test Proportion	Accuracy
80-20	0.97102
70-30	0.96602
60-40	0.96851

In the confusion matrix:
476- 0's are predicted correctly and 13- 0's are predicted wrongly.
similarly,497- 1's are predicted correct and 15 -1's are predicted wrongly

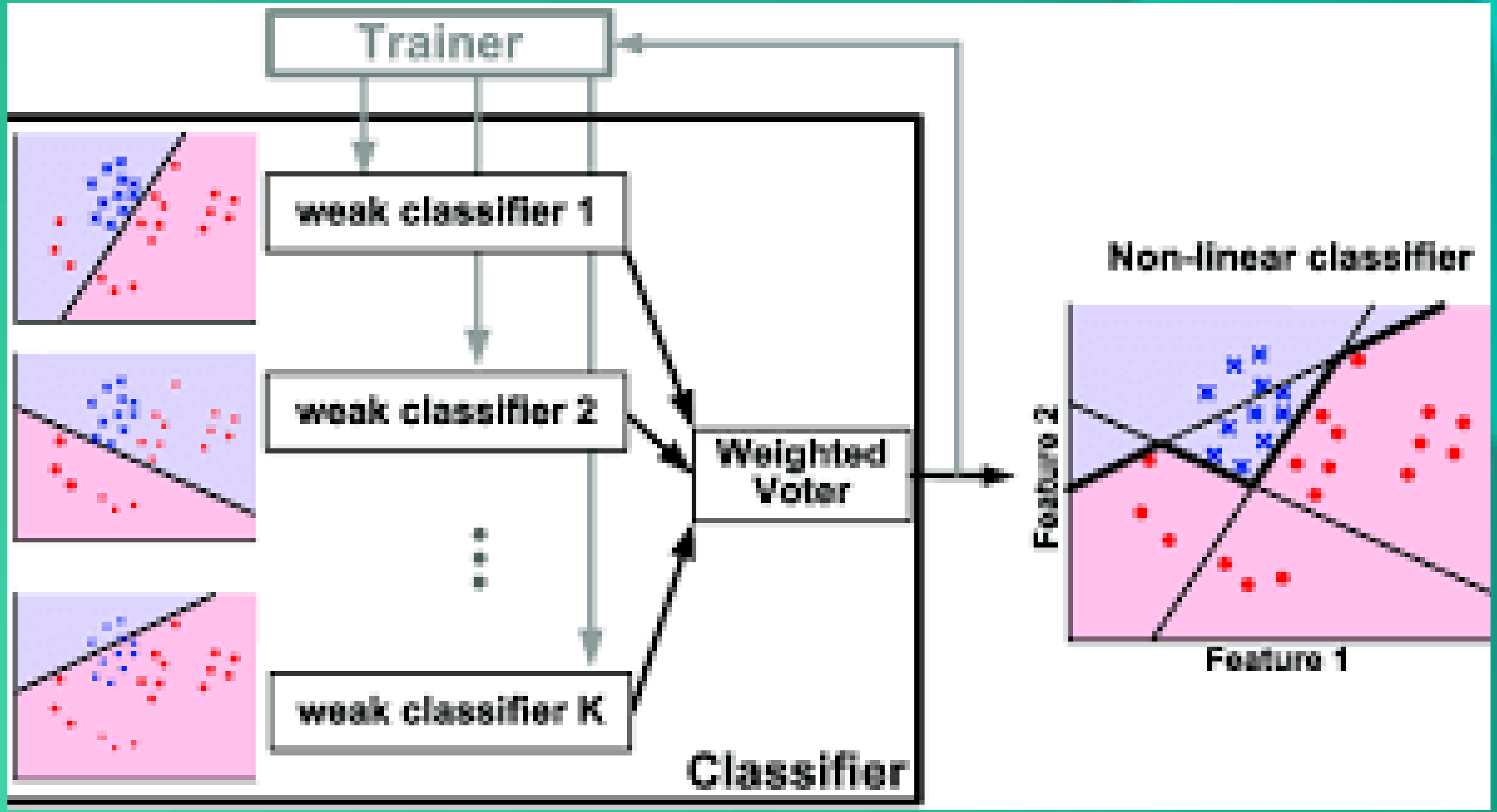
Boosting Algorithm

- Boosting is a method used in machine learning to reduce errors in predictive data analysis.

Adaboost with respect to Decision tree Estimator

- Imported the Adaboost classifier.
- Base estimator-Decision Tree
- fitted the model for training set

Train-Test Proportion	Accuracy with dt
80-20	0.96003
70-30	0.97734
60-40	0.97001



Adaboost with respect to SVM base estimator

- **Base estimator SVM with kernel as radial basis function**

Train-Test Proportion	Accuracy with SVM
80-20	0.95904
70-30	0.95602
60-40	0.62918

Adaboost with no estimators

- Adaboost without any base estimators and n-estimators as 50

Train-Test Proportion	Accuracy without base estimators
80-20	0.97502
70-30	0.97401
60-40	0.96801

Gradient Boosting

- We have applied Gradient boosting to the dataset .
- We have tested the data and predicted the values.

Train-Test Proportion	Accuracy
80-20	0.97602
70-30	0.98401
60-40	0.97701

- In the confusion matrix we observe that 485 0's are predicted correctly and 494 1's are predicted correctly.

Extreme Gradient Boosting

- Model fitting with respect to Extreme gradient boosting.

Train-Test Proportion	Accuracy
80-20	0.96603
70-30	0.98201
60-40	0.97601

Comparision of Algorithms

Models	Accuracy	Train Test-ratio
Gradient boosting	0.98401	70-30
XGB	0.98201	70-30
ADABOOST	0.97502	80-20
SVM with Rad	0.97202	80-20
Random Forest Classifier	0.97102	80-20
Adaboost Ensemble with dt	0.97001	70-30
Decision tree classifier	0.97000	70-30
Logistics Classification	0.97000	80-20
KNN	0.96903	80-20
SVM with Linear	0.96803	80-20
Neural Networks	0.96800	80-20
Adaboost with SVM	0.95602	70-30

Conclusion

- For the gender classification dataset we have applied various Machine learning algorithms.

- Among all the algorithms Gradient boosting and Extreme Gradient Boosting techniques gave us highest accuracy i.e 0.98402 and 0.98201 with 70 -30 ratio
- So we can conclude that boosting techniques fits good for the gender classification dataset and   can be used for further usage of the model.

Team members and their Roles

- N.Bhavana Reddy-Coding,PPT ,EDA
- UudhhayKiirran-Coding,PPT
- Sai Prasanna-Coding,EDA
- Jashwanth-PPT,EDA

Click on the image of colab and github for further more details of the project.

Google Colaboratory
google.com

GitHub


THANK YOU