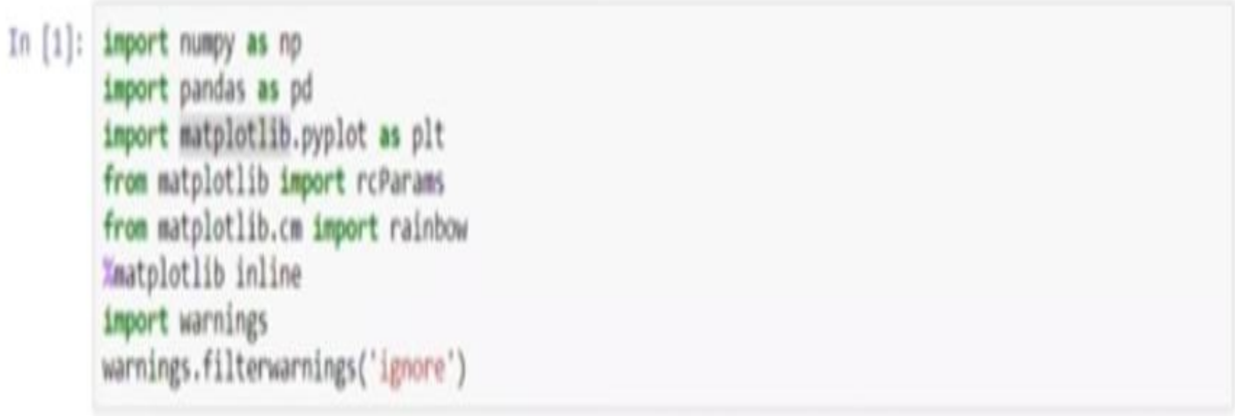
ASSIGNMENT- 2

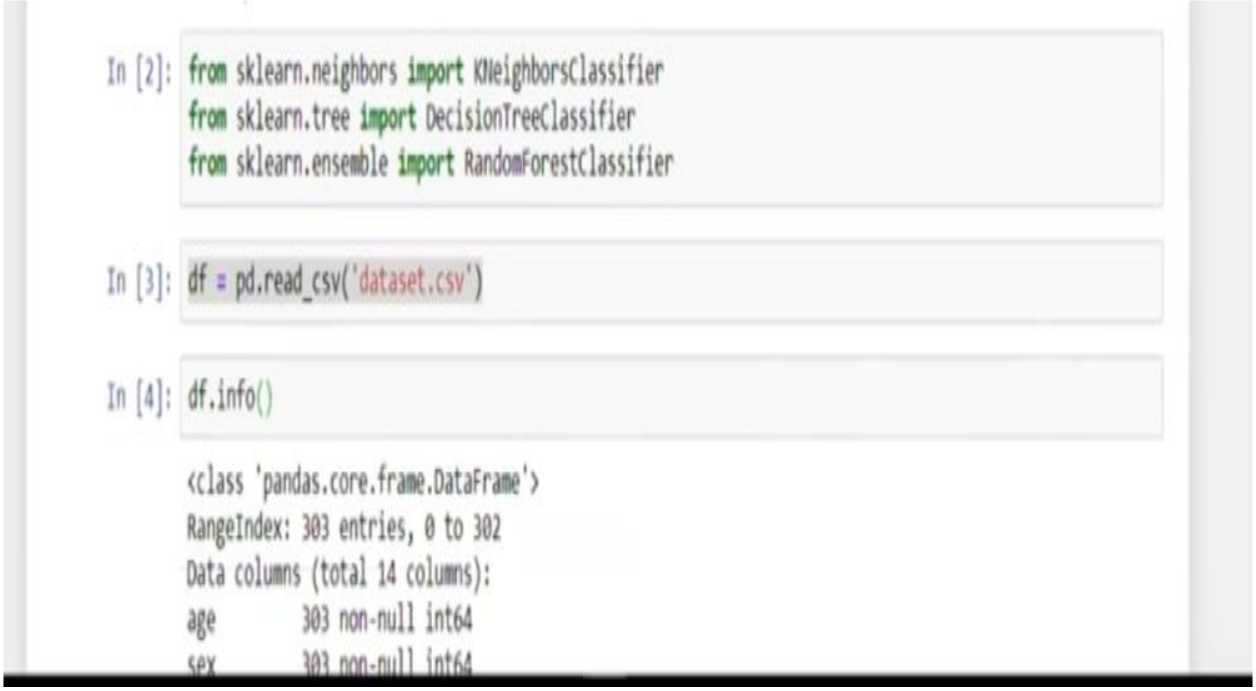
TECHNOLOGY:As machine learning and python

DESCRIPTION: The project is combinate of data analysis and machine learning In the project will focus on predicting heart disease using neural network

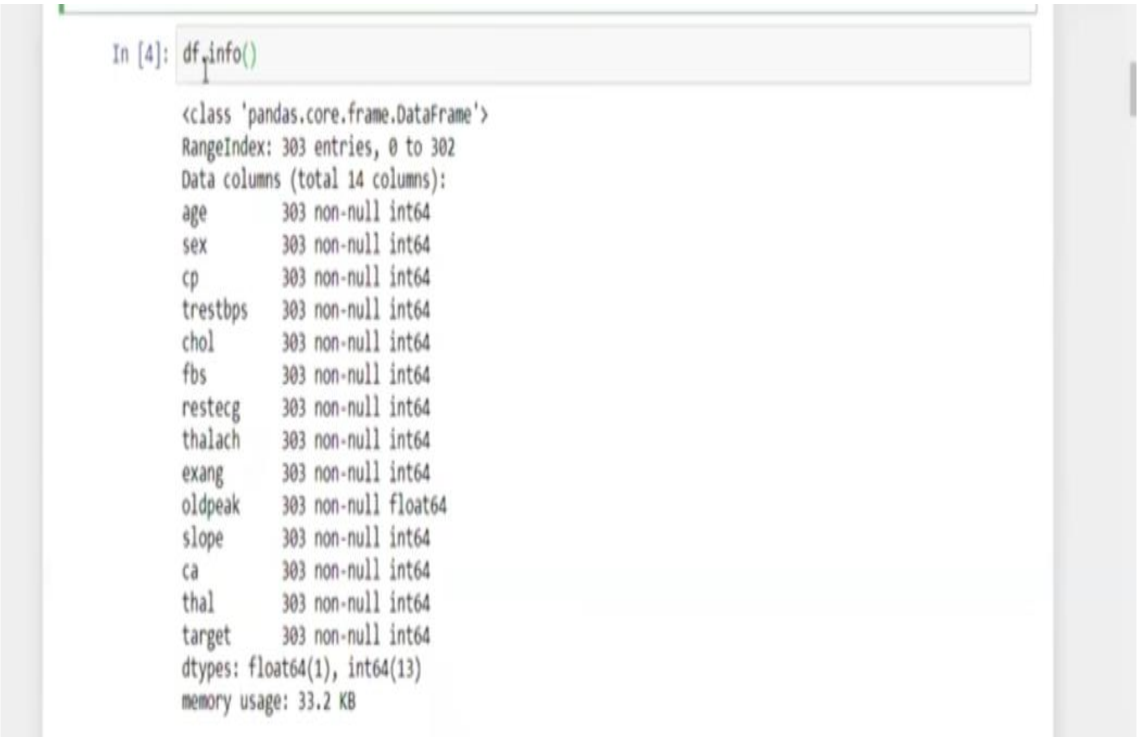
PROGRAM:



```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        from matplotlib import rcParams
        from matplotlib.cm import rainbow
        %matplotlib inline
        import warnings
        warnings.filterwarnings('ignore')
```

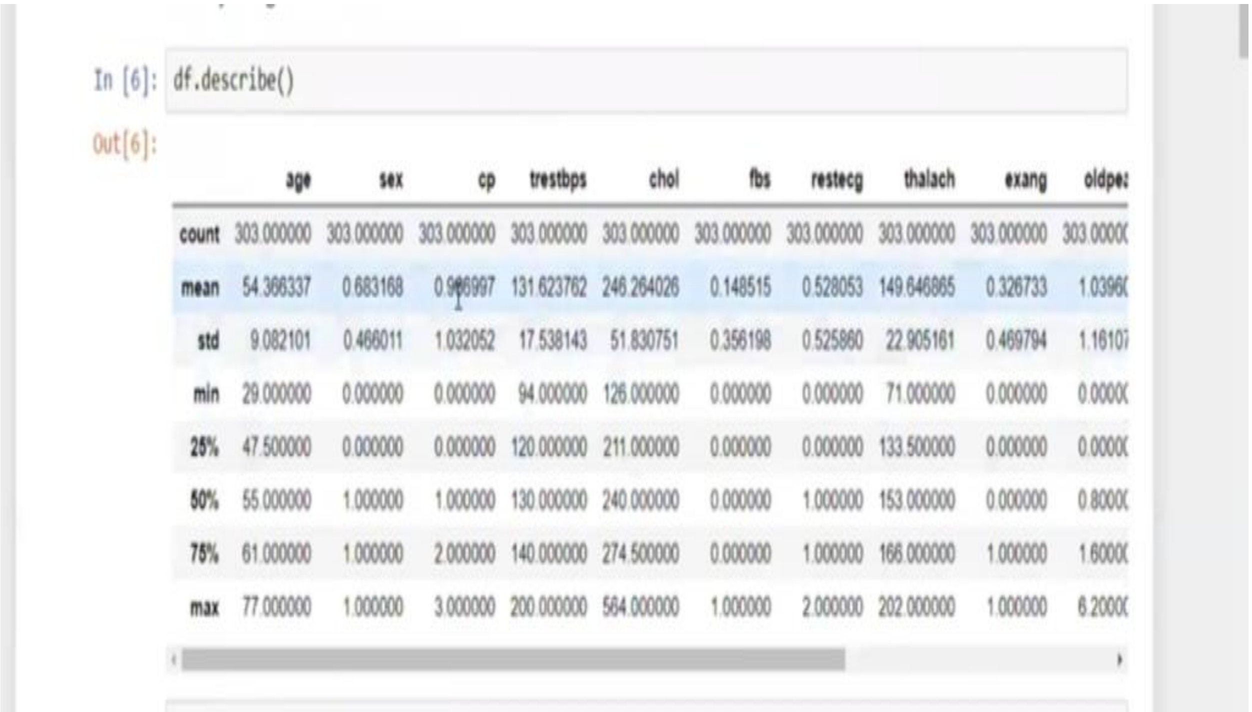


```
In [2]: from sklearn.neighbors import KNeighborsClassifier
        from sklearn.tree import DecisionTreeClassifier
        from sklearn.ensemble import RandomForestClassifier

In [3]: df = pd.read_csv('dataset.csv')

In [4]: df.info()

        <class 'pandas.core.frame.DataFrame'>
        RangeIndex: 303 entries, 0 to 302
        Data columns (total 14 columns):
        age        303 non-null int64
        sex        303 non-null int64
```



In [4]:
```
df.info()
```

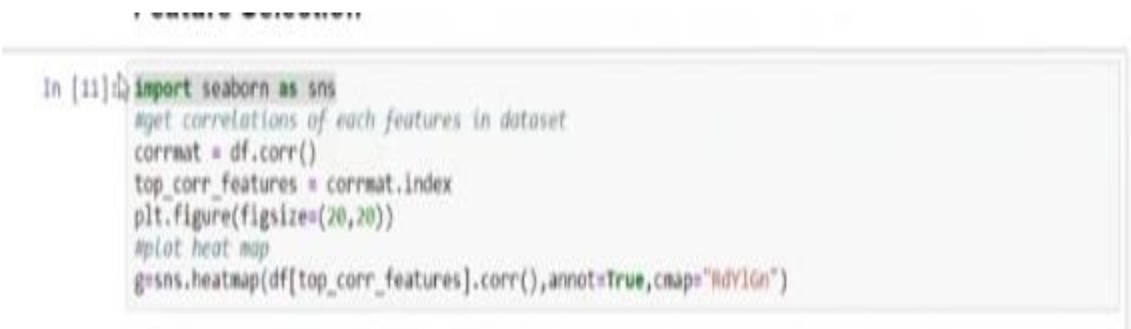
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
age         303 non-null int64
sex         303 non-null int64
cp          303 non-null int64
trestbps    303 non-null int64
chol        303 non-null int64
fbs         303 non-null int64
restecg     303 non-null int64
thalach     303 non-null int64
exang       303 non-null int64
oldpeak     303 non-null float64
slope       303 non-null int64
ca          303 non-null int64
thal        303 non-null int64
target      303 non-null int64
dtypes: float64(1), int64(13)
memory usage: 33.2 KB
```



In [6]:
```
df.describe()
```

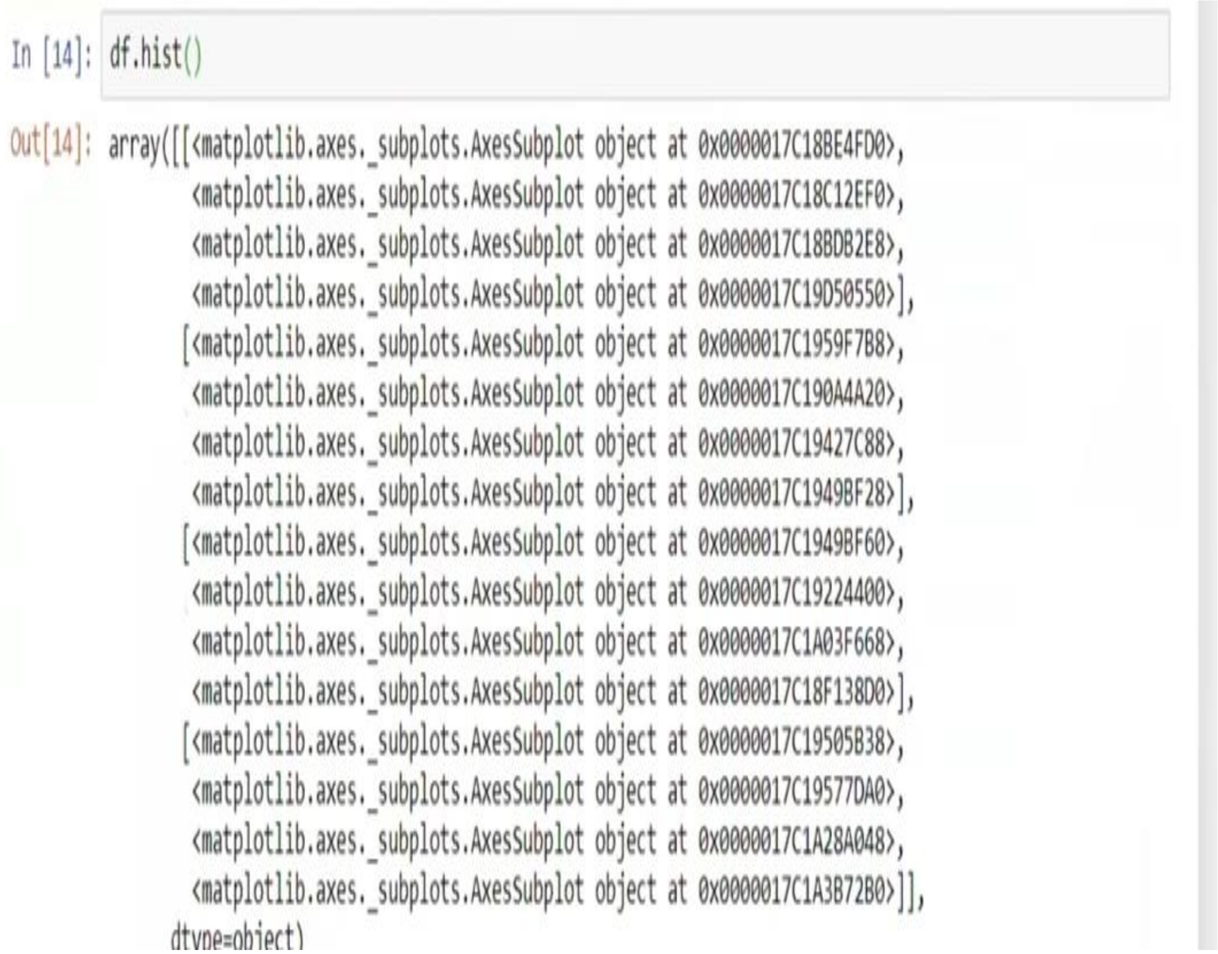
Out[6]:

| | age | sex | cp | trestbps | chol | fbs | restecg | thalach | exang | oldpe |
|---|---|---|---|---|---|---|---|---|---|---|
| **count** | 303.000000 | 303.000000 | 303.000000 | 303.000000 | 303.000000 | 303.000000 | 303.000000 | 303.000000 | 303.000000 | 303.0000 |
| **mean** | 54.366337 | 0.683168 | 0.966997 | 131.623762 | 246.264026 | 0.148515 | 0.528053 | 149.646865 | 0.326733 | 1.0396 |
| **std** | 9.082101 | 0.466011 | 1.032052 | 17.538143 | 51.830751 | 0.356198 | 0.525860 | 22.905161 | 0.469794 | 1.1610 |
| **min** | 29.000000 | 0.000000 | 0.000000 | 94.000000 | 126.000000 | 0.000000 | 0.000000 | 71.000000 | 0.000000 | 0.0000 |
| **25%** | 47.500000 | 0.000000 | 0.000000 | 120.000000 | 211.000000 | 0.000000 | 0.000000 | 133.500000 | 0.000000 | 0.0000 |
| **50%** | 55.000000 | 1.000000 | 1.000000 | 130.000000 | 240.000000 | 0.000000 | 1.000000 | 153.000000 | 0.000000 | 0.8000 |
| **75%** | 61.000000 | 1.000000 | 2.000000 | 140.000000 | 274.500000 | 0.000000 | 1.000000 | 166.000000 | 1.000000 | 1.6000 |
| **max** | 77.000000 | 1.000000 | 3.000000 | 200.000000 | 564.000000 | 1.000000 | 2.000000 | 202.000000 | 1.000000 | 6.2000 |

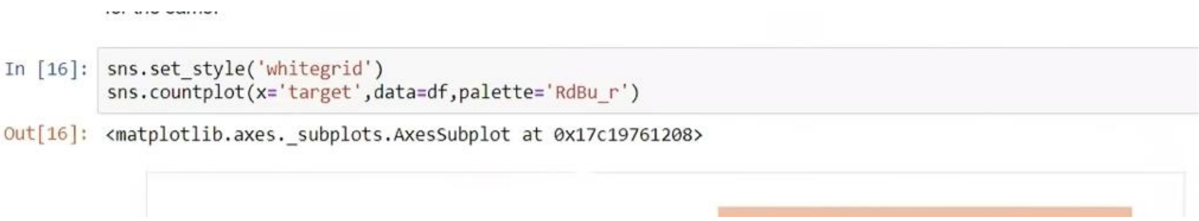


Feature Selection

```
In [11]: import seaborn as sns
         #get correlations of each features in dataset
         corrmat = df.corr()
         top_corr_features = corrmat.index
         plt.figure(figsize=(20,20))
         #plot heat map
         g=sns.heatmap(df[top_corr_features].corr(),annot=True,cmap="RdYlGn")
```

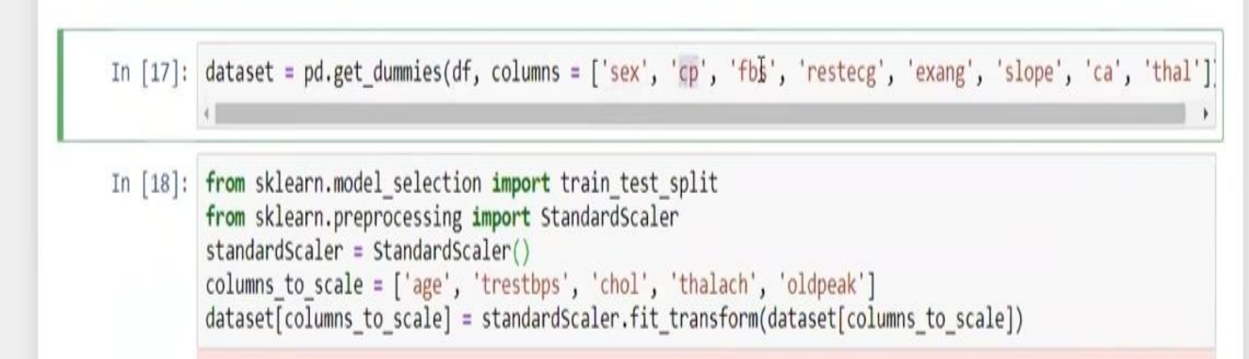


```
In [14]: df.hist()

Out[14]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000017C18BE4FD0>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000017C18C12EF0>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000017C18BDB2E8>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000017C19D50550>],
                [<matplotlib.axes._subplots.AxesSubplot object at 0x0000017C1959F7B8>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000017C190A4A20>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000017C19427C88>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000017C1949BF28>],
                [<matplotlib.axes._subplots.AxesSubplot object at 0x0000017C1949BF60>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000017C19224400>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000017C1A03F668>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000017C18F138D0>],
                [<matplotlib.axes._subplots.AxesSubplot object at 0x0000017C19505B38>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000017C19577DA0>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000017C1A28A048>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000017C1A3B72B0>]],
               dtype=object)
```

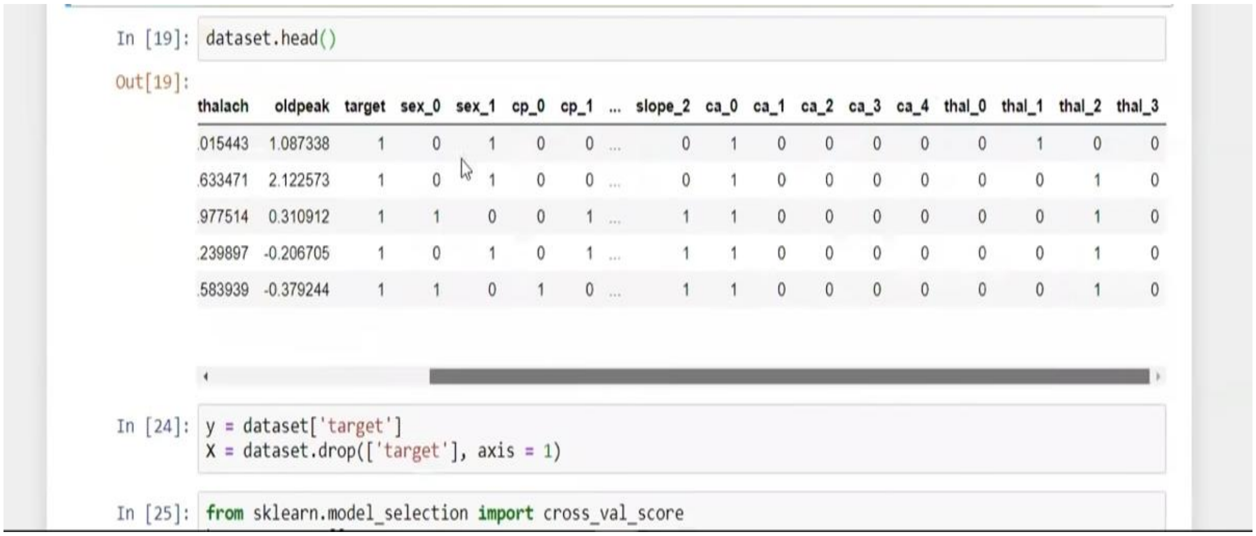


for the same.

```
In [16]: sns.set_style('whitegrid')
         sns.countplot(x='target',data=df,palette='RdBu_r')

Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x17c19761208>
```



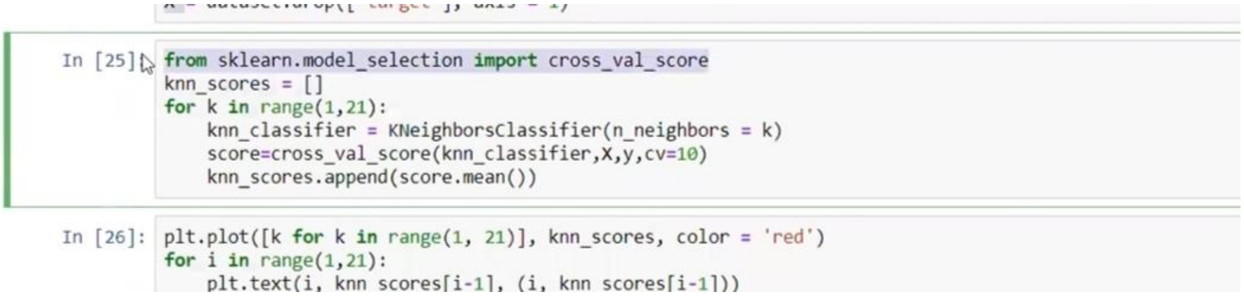
```
In [17]: dataset = pd.get_dummies(df, columns = ['sex', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'ca', 'thal']

In [18]: from sklearn.model_selection import train_test_split
         from sklearn.preprocessing import StandardScaler
         standardScaler = StandardScaler()
         columns_to_scale = ['age', 'trestbps', 'chol', 'thalach', 'oldpeak']
         dataset[columns_to_scale] = standardScaler.fit_transform(dataset[columns_to_scale])
```



```
In [19]: dataset.head()

Out[19]:
```

| thalach | oldpeak | target | sex_0 | sex_1 | cp_0 | cp_1 | ... | slope_2 | ca_0 | ca_1 | ca_2 | ca_3 | ca_4 | thal_0 | thal_1 | thal_2 | thal_3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| .015443 | 1.087338 | 1 | 0 | 1 | 0 | 0 | ... | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| .633471 | 2.122573 | 1 | 0 | 1 | 0 | 0 | ... | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| .977514 | 0.310912 | 1 | 1 | 0 | 0 | 1 | ... | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| .239897 | -0.206705 | 1 | 0 | 1 | 0 | 1 | ... | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| .583939 | -0.379244 | 1 | 1 | 0 | 1 | 0 | ... | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

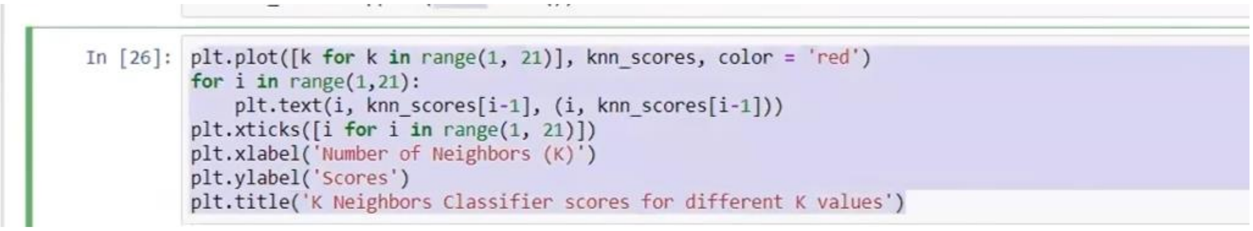
```
In [24]: y = dataset['target']
         X = dataset.drop(['target'], axis = 1)

In [25]: from sklearn.model_selection import cross_val_score
```



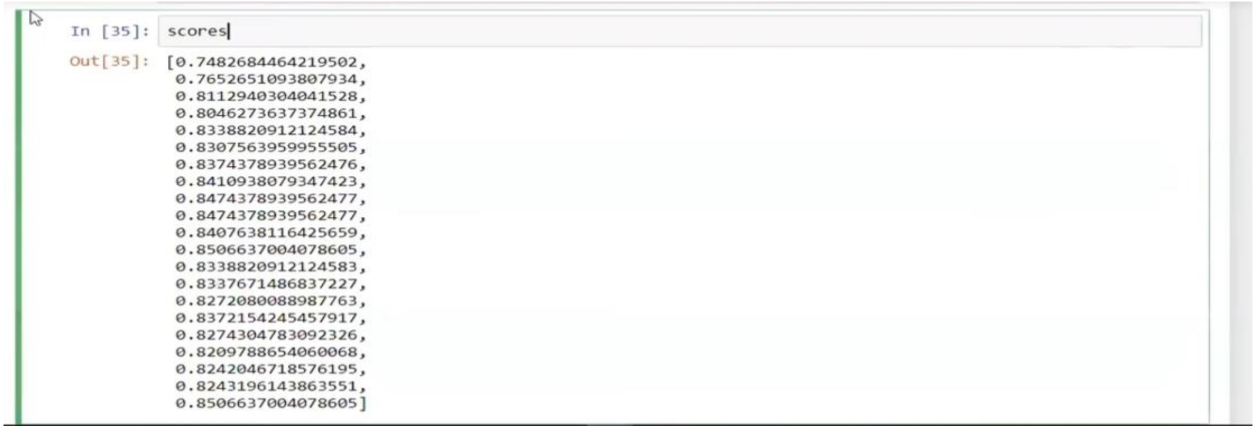
```
X = dataset.drop(['target'], axis = 1)
```

```
In [25]: from sklearn.model_selection import cross_val_score
         knn_scores = []
         for k in range(1,21):
             knn_classifier = KNeighborsClassifier(n_neighbors = k)
             score=cross_val_score(knn_classifier,X,y,cv=10)
             knn_scores.append(score.mean())
```

```
In [26]: plt.plot([k for k in range(1, 21)], knn_scores, color = 'red')
         for i in range(1,21):
             plt.text(i, knn_scores[i-1], (i, knn_scores[i-1]))
```

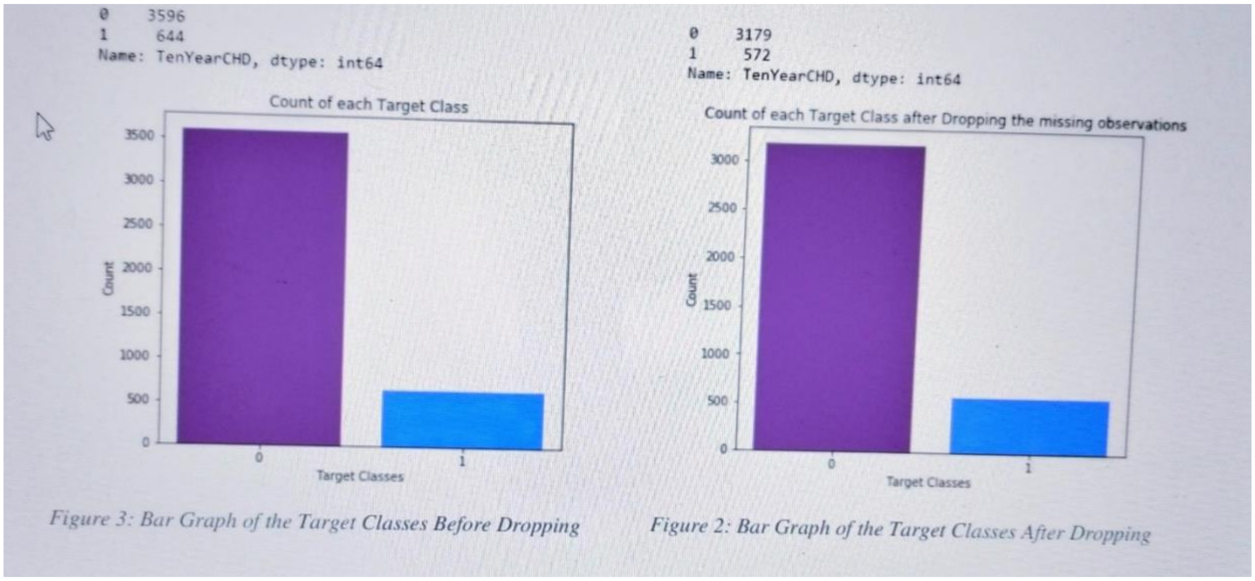


```
In [26]: plt.plot([k for k in range(1, 21)], knn_scores, color = 'red')
         for i in range(1,21):
             plt.text(i, knn_scores[i-1], (i, knn_scores[i-1]))
         plt.xticks([i for i in range(1, 21)])
         plt.xlabel('Number of Neighbors (K)')
         plt.ylabel('Scores')
         plt.title('K Neighbors Classifier scores for different K values')
```



```
In [35]: scores
Out[35]: [0.7482684464219502,
          0.7652651093807934,
          0.8112940304041528,
          0.8046273637374861,
          0.8338820912124584,
          0.8307563959955505,
          0.8374378939562476,
          0.8410938079347423,
          0.8474378939562477,
          0.8474378939562477,
          0.8407638116425659,
          0.8506637004078605,
          0.8338820912124583,
          0.8337671486837227,
          0.8272080088987763,
          0.8372154245457917,
          0.8274304783092326,
          0.8209788654060068,
          0.8242046718576195,
          0.8243196143863551,
          0.8506637004078605]
```

DIAGRAM:

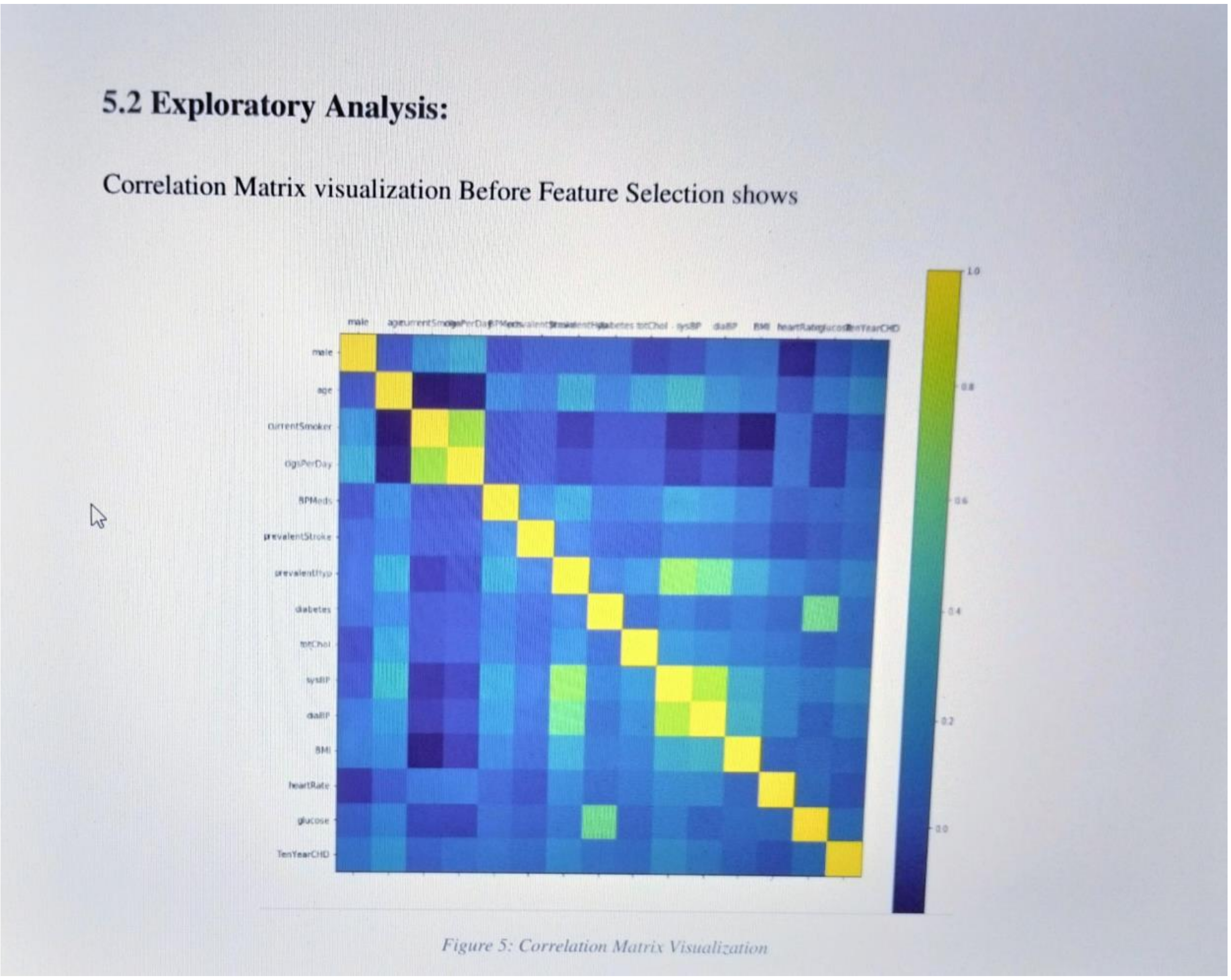


```
0    3596
1     644
Name: TenYearCHD, dtype: int64
```

*Figure 3: Bar Graph of the Target Classes Before Dropping*

```
0    3179
1     572
Name: TenYearCHD, dtype: int64
```

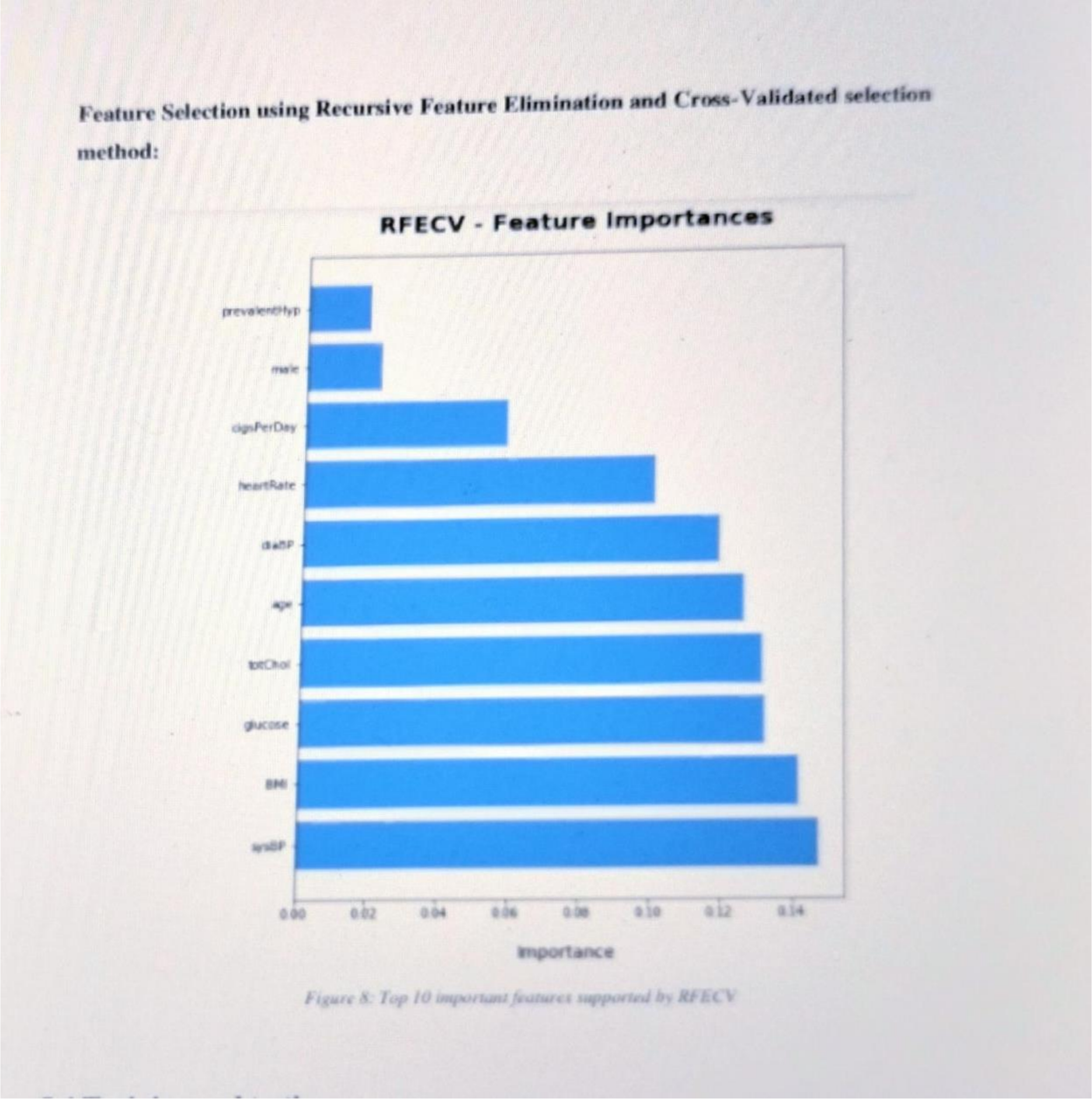
*Figure 2: Bar Graph of the Target Classes After Dropping*



## 5.2 Exploratory Analysis:

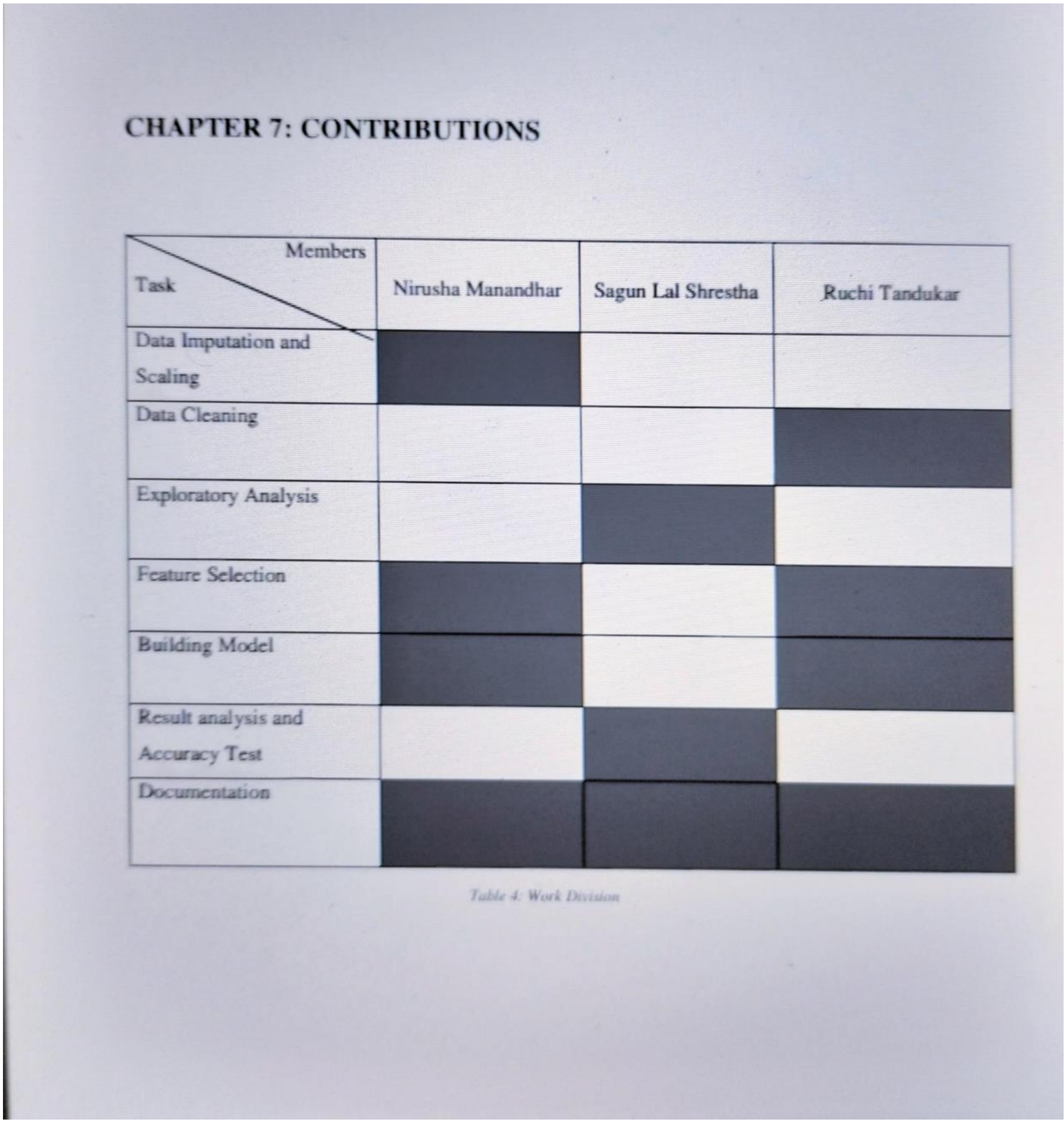
Correlation Matrix visualization Before Feature Selection shows

*Figure 5: Correlation Matrix Visualization*



**Feature Selection using Recursive Feature Elimination and Cross-Validated selection method:**

*Figure 8: Top 10 important features supported by RFECV*



# CHAPTER 7: CONTRIBUTIONS

| Members<br>Task | Nirusha Manandhar | Sagun Lal Shrestha | Ruchi Tandukar |
|---|---|---|---|
| Data Imputation and Scaling | ■ |  |  |
| Data Cleaning |  |  | ■ |
| Exploratory Analysis |  | ■ |  |
| Feature Selection | ■ |  | ■ |
| Building Model | ■ |  | ■ |
| Result analysis and Accuracy Test |  | ■ |  |
| Documentation | ■ | ■ | ■ |

*Table 4: Work Division*

TEAM LEADER: S.AFREEN BANU

TEAM Members

- B.SANGEETHA
- K.KAVIYA
- R.GAYATHRI

TEAM ID: PNT2022TMID37943

TEAM SIZE:4

TEAM MENTORS: VIJAYAKUMARI