```
# import the required modules

import pandas as pd, numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn import model_selection
from sklearn.model_selection import train_test_split
import warnings
warnings.filterwarnings('ignore')
import matplotlib.pyplot as plt

# download dataset from kaggle and load to pandas data frame

liver = pd.read_csv("indian_liver_patient.csv")

liver.head()
```

```
   Age  Gender  Total_Bilirubin  Direct_Bilirubin
Alkaline_Phosphotase  \
0   65  Female              0.7               0.1
187
1   62    Male             10.9               5.5
699
2   62    Male              7.3               4.1
490
3   58    Male              1.0               0.4
182
4   72    Male              3.9               2.0
195

   Alamine_Aminotransferase  Aspartate_Aminotransferase
Total_Protiens  \
0                        16                          18
6.8
1                        64                         100
7.5
2                        60                          68
7.0
3                        14                          20
6.8
4                        27                          59
7.3

   Albumin  Albumin_and_Globulin_Ratio  Dataset
0      3.3                        0.90        1
1      3.2                        0.74        1
2      3.3                        0.89        1
3      3.4                        1.00        1
4      2.4                        0.40        1
```

```
# data preparation and cleaning

# let check the dimensions of the dataframe

liver.shape
```

```
(583, 11)
```

```
# let check the missing  values in the dataframe

round(100*(liver.isnull().sum()/len(liver.index)),2)
```

```
Age                           0.00
Gender                        0.00
Total_Bilirubin               0.00
Direct_Bilirubin              0.00
Alkaline_Phosphotase          0.00
Alamine_Aminotransferase      0.00
Aspartate_Aminotransferase    0.00
Total_Protiens                0.00
Albumin                       0.00
Albumin_and_Globulin_Ratio    0.69
Dataset                       0.00
dtype: float64
```

```
# only Albumin_and_Globulin_Ratio has some missing values.
# let drop the NAN values

liver.dropna(inplace=True)

liver.shape
```

```
(579, 11)
```

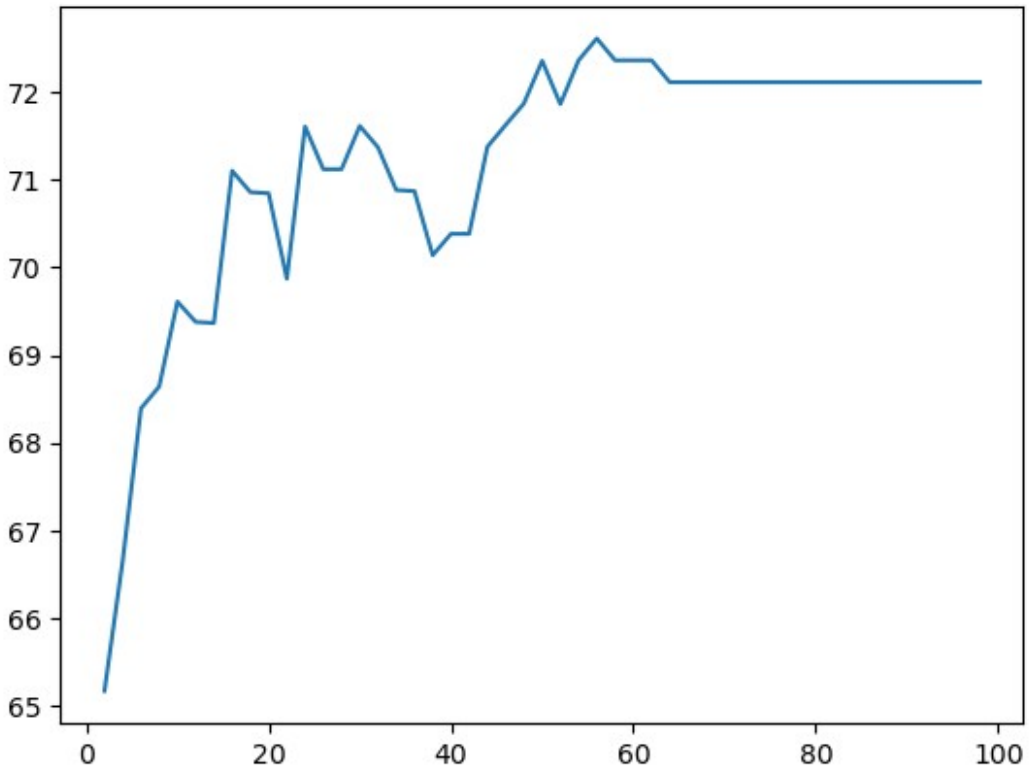
```
liver['Gender'] = liver.Gender.map({'Female':2,'Male':1})

# putting  feature variable to x

x = liver.drop('Dataset',axis=1)

# putting  response variable to y

y = liver['Dataset']

# rescaling and split the data set into test and train

x_std  = StandardScaler().fit_transform(x)

# Splitting the data into train and test

x_train, x_test, y_train, y_test =
train_test_split(x_std,y,test_size=0.30,random_state=100)

neighbors = [x for x in list(range(1,100)) if x % 2==0]
```

```
cv_scores = []

# perfrom 10-fold cross validation on training set for odd values of
k:

seed = 123

for k in neighbors:
    k_value = k+1
    knn = KNeighborsClassifier(n_neighbors =
k_value,weights='uniform',p=2,metric='euclidean')
    kfold = model_selection.KFold(n_splits=10)
    scores =
model_selection.cross_val_score(knn,x_train,y_train,cv=kfold,scoring='
accuracy')
    cv_scores.append(scores.mean()*100)

optimal_k = neighbors[cv_scores.index(max(cv_scores))]

print(optimal_k)
```

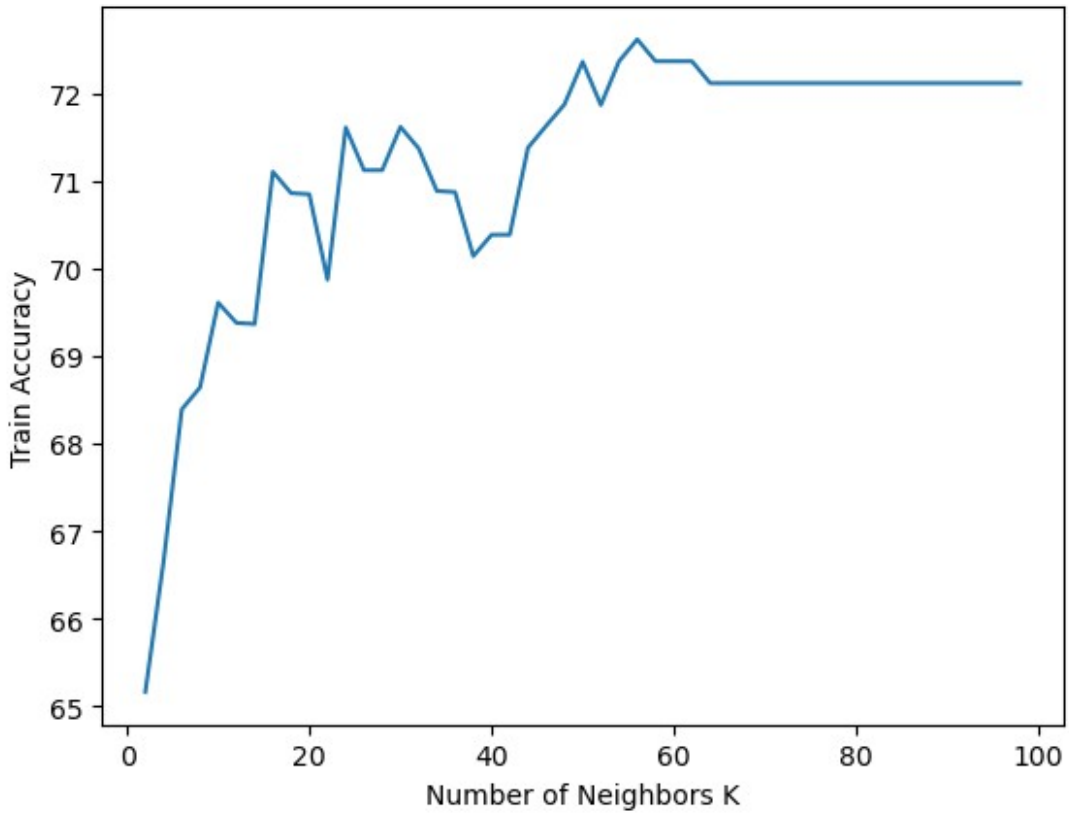
```
56
```

```
# print(("The optimum number of neighbors is %d with %0.1f%%"%
(optimal_k,cv_scores[optimal_k])))

plt.plot(neighbors,cv_scores)
```

```
[<matplotlib.lines.Line2D at 0x1829d671c50>]
```




```
plt.xlabel('Number of Neighbors K')
plt.ylabel('Train Accuracy')
plt.show(plt.plot(neighbors,cv_scores))
```




```
# model building

knn = KNeighborsClassifier(n_neighbors = 56)

knn.fit(x_train,y_train)

KNeighborsClassifier(n_neighbors=56)

y_pred = knn.predict(x_test)

acc_train = round(knn.score(x_train,y_train)*100,2)
acc_val = round(knn.score(x_test,y_test)*100,2)
print("Accuracy of training dataset:"+str(acc_train))
print("Accuracy of test dataset:"+str(acc_val))

Accuracy of training dataset:72.84
Accuracy of test dataset:70.69

# At neighbors 56, the accuracy of prediction of liver disease from
KNN is 70.69
```