CM6

January 29, 2021

1 Finding best k for datasets

1.1 Required Libraries

[1]:

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
```

1.2 KNN on Heart Diseases Dataset

[2]:

```
df_heart= pd.read_csv("heart_disease_missing.csv")
df_heart=df_heart.interpolate(method ='linear', limit_direction ='forward')
```

Splitting the data in train validation and test sets

[3]:

```
X = df_heart.iloc[:, 0:13].values
y = df_heart.iloc[:,13].values
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.
 ↪4,random_state=275)
X_vali, X_test, y_vali, y_test =train_test_split(X_test, y_test, test_size=0.
 ↪5,random_state=275)
fe_sc=StandardScaler()
X_train=fe_sc.fit_transform(X_train)
X_vali=fe_sc.fit_transform(X_vali)
X_test=fe_sc.fit_transform(X_test)
```

Model is trained with the default parameters Accuracy obtained for default values of classifier is 88.09.

[4]:

```
classifier = KNeighborsClassifier()
classifier.fit(X_train,y_train)
y_output= classifier.predict(X_vali)
a=accuracy_score(y_vali,y_output)
a*100
```

```
[4]: 88.09523809523809
```

Finding k which gives highest accuracy Values 15,20,25 of k gives the highest accuracy.

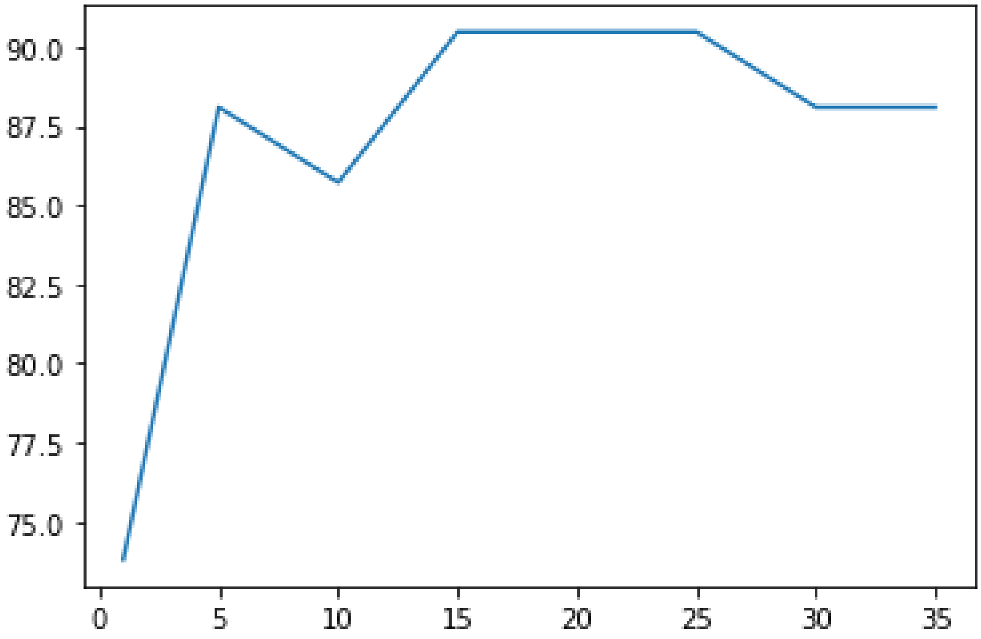
```
[5]: n_neighbors= [1, 5, 10, 15, 20, 25, 30, 35]
arr=[]

for i in n_neighbors:
            classifier = KNeighborsClassifier(i)
            classifier.fit(X_train,y_train)
            y_output= classifier.predict(X_vali)
            a=accuracy_score(y_vali,y_output)
            arr.append(a*100)

print(arr)
plt.plot(n_neighbors,arr)
```

```
[73.80952380952381, 88.09523809523809, 85.71428571428571, 90.47619047619048,
90.47619047619048, 90.47619047619048, 88.09523809523809, 88.09523809523809]
```

```
[5]: [<matplotlib.lines.Line2D at 0x21e82866f40>]
```



2 KNN on Iris Dataset

[6]:

```
df_iris= pd.read_csv("iris_dataset_missing.csv")
df_iris=df_iris.interpolate(method ='linear', limit_direction ='forward')
```

Splitting the data in train validation and test sets

[7]:

```
X = df_iris.iloc[:, 0:4].values
y = df_iris.iloc[:,4].values
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.
 ↪4,random_state=275)
X_vali, X_test, y_vali, y_test =train_test_split(X_test, y_test, test_size=0.
 ↪5,random_state=275)
fe_sc=StandardScaler()
X_train=fe_sc.fit_transform(X_train)
X_vali=fe_sc.fit_transform(X_vali)
X_test=fe_sc.fit_transform(X_test)
```

Model is trained with the default parameters Accuracy obtained for default values of classifier is 95.23.

[8]:

```
classifier = KNeighborsClassifier()
classifier.fit(X_train,y_train)
y_output= classifier.predict(X_vali)
a=accuracy_score(y_vali,y_output)
a*100
```

[8]:

```
95.23809523809523
```

Finding k which gives highest accuracy Values 1 and 5 gives the highest accuracy.

[9]:

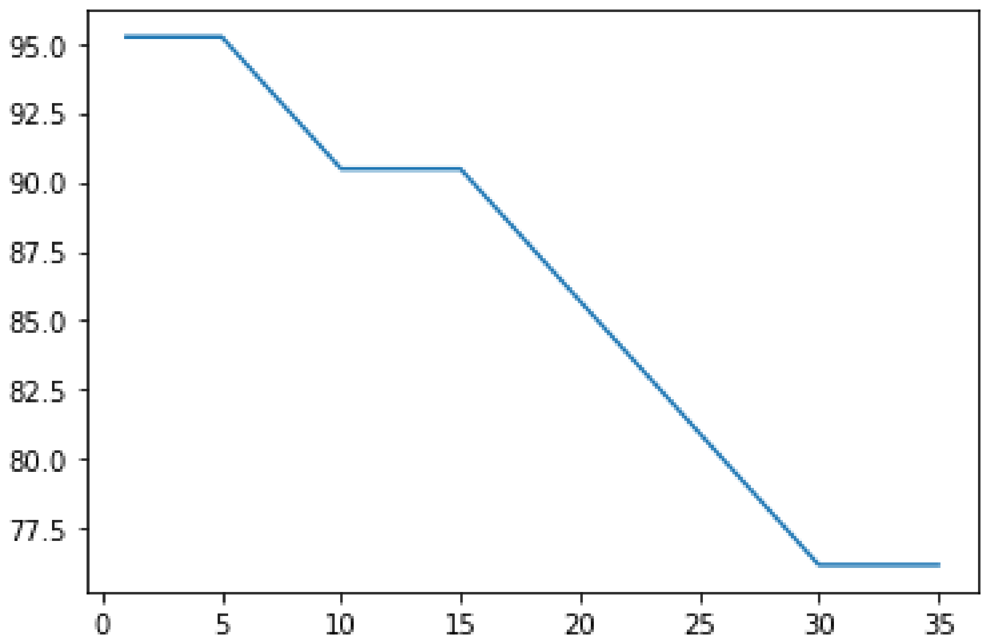
```
n_neighbors= [1, 5, 10, 15, 20, 25, 30, 35]
arr=[]
for i in n_neighbors:
            classifier = KNeighborsClassifier(i)
            classifier.fit(X_train,y_train)
            y_output= classifier.predict(X_vali)
            a=accuracy_score(y_vali,y_output)
            arr.append(a*100)

print(arr)
plt.plot(n_neighbors,arr)
```

```
[95.23809523809523, 95.23809523809523, 90.47619047619048, 90.47619047619048,
85.71428571428571, 80.95238095238095, 76.19047619047619, 76.19047619047619]
```

[9]:

```
[<matplotlib.lines.Line2D at 0x21e82ba2d00>]
```



2.1 References

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_csv.html
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.interpolate.html
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html