

```

1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 from sklearn.metrics import confusion_matrix
5 from sklearn.neighbors import KNeighborsClassifier
6 from sklearn.model_selection import train_test_split
7 from sklearn import preprocessing

```

```

1 data=pd.read_csv('/content/Train Knn.csv')
2 data.dropna(inplace=True)
3 data.drop(columns=['loan_id'],inplace=True)

```

```
1 data.shape
```

```
(6755, 9)
```

```
1 data.columns
```

```

Index(['age', 'education', 'proof_submitted', 'loan_amount', 'asset_cost',
      'no_of_loans', 'no_of_curr_loans', 'last_delinq_none', 'loan_default'],
      dtype='object')

```

```
1 data.head(10)
```

	age	education	proof_submitted	loan_amount	asset_cost	no_of_loans	no_of_curr
0	27	1.0	Aadhar	504264	820920	2	
1	48	1.0	Aadhar	728556	831444	6	
2	30	2.0	VoterID	642936	826092	0	
3	28	1.0	Aadhar	746556	930924	0	
4	29	1.0	Aadhar	1139880	1902000	0	
5	34	2.0	Aadhar	779784	902040	0	
6	27	2.0	Aadhar	449268	847896	0	
7	27	2.0	Aadhar	582036	905604	0	
8	30	1.0	Aadhar	712956	866292	0	
9	46	2.0	Aadhar	554988	761724	3	

```
1 #look_up_fruit=dict(zip(data.fruit_label.unique(), data.fruit_name.unique()))
```

```
1 #look_up_fruit
```

```

1 label_encoder = preprocessing.LabelEncoder()
2 data['proof_submitted']= label_encoder.fit_transform(data['proof_submitted'])
3 data.head()

```

	age	education	proof_submitted	loan_amount	asset_cost	no_of_loans	no_of_curr
0	27	1.0	0	504264	820920	2	
1	48	1.0	0	728556	831444	6	
2	30	2.0	4	642936	826092	0	
3	28	1.0	0	746556	930924	0	
4	29	1.0	0	1139880	1902000	0	

```
1 X=data[['age','education','proof_submitted','loan_amount','asset_cost','no_of_loans','no_of_curr']
2 y=data['loan_default']
```

Double-click (or enter) to edit

```
1 X_train, X_test, y_train, y_test=train_test_split(X, y, random_state=0)
```

```
1 knn=KNeighborsClassifier(n_neighbors=5)
```

```
1 """if(np.any(np.isnan(X)) or np.all(np.isfinite(X)) or np.any(np.isnan(y)) or np.all(np.isfinite(y))):
2     print()
3 else:
4     knn.fit(X_train, y_train)"""
5 knn.fit(X_train, y_train)
```

KNeighborsClassifier()

```
1 tdata=pd.read_csv('/content/Test Knn.csv')
2 tdata.dropna(inplace=True)
3 tdata.drop(columns=['loan_id'],inplace=True)
4 tdata['proof_submitted']= label_encoder.fit_transform(tdata['proof_submitted'])
5 Xt=tdata[['age','education','proof_submitted','loan_amount','asset_cost','no_of_loans','no_of_curr']]
6 knn.fit(X_train,X_test)
7 y_predict=knn.predict(Xt)
8 #y_correct=np.array(y_test)
9 #print(np.concatenate((y_predict.reshape(len(y_predict), 1), y_correct.reshape(len(y_correct), 1)), axis=1))
```

```
1 knn.score(X_train, y_train)
```

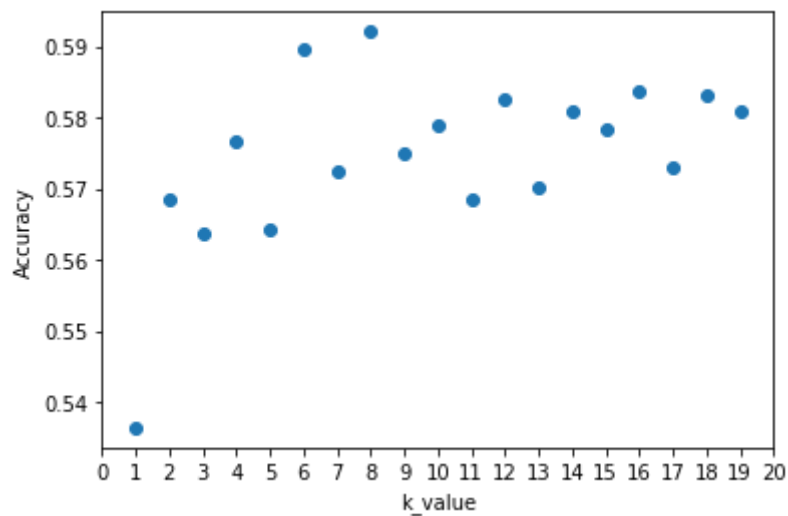
0.7090406632451638

```
1 k_range = range(1,20)
2 scores = []
3 for k in k_range:
4     knn = KNeighborsClassifier(n_neighbors = k)
5     knn.fit(X_train, y_train)
6     scores.append(knn.score(X_test, y_test))
7 plt.figure()
8 plt.xlabel('k_value')
```

```

9 plt.ylabel('Accuracy')
10 plt.scatter(k_range, scores)
11 plt.xticks(range(0,21));
12

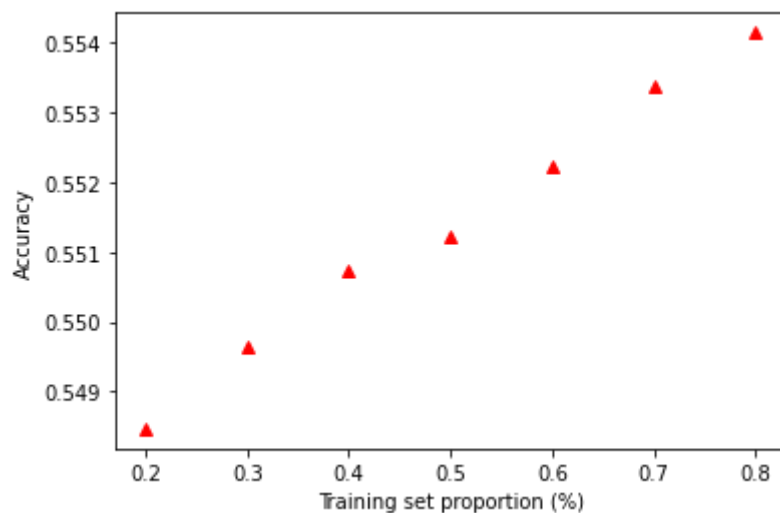
```



```

1 t = [0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8]
2 knn = KNeighborsClassifier(n_neighbors = 3)
3 plt.figure()
4 for split in t:
5     scores = []
6     for i in range(1,1000):
7         X_train, X_test, y_train, y_test = train_test_split(X, y, train_size = split)
8         knn.fit(X_train, y_train)
9         scores.append(knn.score(X_test, y_test))
10    plt.plot(split, np.mean(scores), 'r^')
11 plt.xlabel('Training set proportion (%)')
12 plt.ylabel('Accuracy');

```



Colab paid products - Cancel contracts here