Predicting User's Decision to purchase from ads based on age and salary

Overview of the POC project

Description of the data:

- This data consists of 5 rows and 500, four features are, Gender, Age, estimated salary
- The target variable is the 5th column which is a binary stating whether the customer purchased from the add or not

Data Cleaning:

- Data was quite clean and minimum data cleaning was required

Exploratory data analysis

- Feature selection was done by making a correlation matrix and checking it's correlation with the target variable, that is estimated salary
- It was found that the feature user ID is not realted to the target variable and it was dropped

Data Preprocessing and implementation of model

- Units of the features are in different range, normalization was performed using standard scaler
- data is now ready to be trained
- K-Nearest Neighbour was used with the distance metric as minkowski and a standard the n-neighbour to be5
- n-neighbours is always to be chosen an odd number as it avoids the draw-conflict when making a prediction
- Accuracy of 82% was obtained

Optimising the model for increasing accuracy

- The model was iterated over different values of n, best value of n was observed to be n = 7.
- The model was iterated over different distance metrics, all gave nearly the same result.

In [74]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Accessing the directory

In [75]:

```
pwd
```

Out[75]:

```
'C:\\Users\\user\\Desktop\\Refactored_Py_DS_ML_Bootcamp-master\\20-Natural
-Language-Processing'
```

In [76]:

```
df = pd.read_csv("original.csv")
df
```

Out[76]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0
...	...	...	...	...	...
395	15691863	Female	46	41000	1
396	15706071	Male	51	23000	1
397	15654296	Female	50	20000	1
398	15755018	Male	36	33000	0
399	15594041	Female	49	36000	1

400 rows × 5 columns

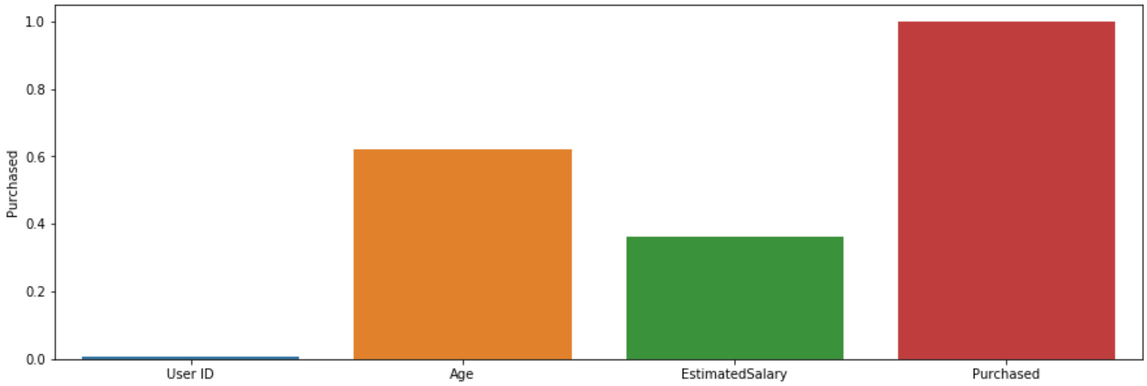
Correlation between different features

In [77]:

```
plt.figure(figsize = (15,5))
hm = df.corr()["Purchased"]
hm
sns.barplot(hm.index, hm)
```

Out[77]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2020fa2dd88>
```



In [ ]:

```

```

We can safely select the features Age, Estimated Salary, and drop user id

In [78]:

```
X = df.iloc[:,[2,3]]
```

In [79]:

```
X
```

Out[79]:

	Age	EstimatedSalary
0	19	19000
1	35	20000
2	26	43000
3	27	57000
4	19	76000
...	...	...
395	46	41000
396	51	23000
397	50	20000
398	36	33000
399	49	36000

400 rows × 2 columns

In [80]:

```
y = df.iloc[:,4]
```

In [81]:

```
y
```

Out[81]:

```
0      0
1      0
2      0
3      0
4      0
      ..
395    1
396    1
397    1
398    0
399    1
Name: Purchased, Length: 400, dtype: int64
```

In [82]:

```
from sklearn.model_selection import train_test_split
```

In [83]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=
42)
```

In [84]:

```
print(X_train.shape, y_train.shape , X_test.shape, y_test.shape)
```

```
(300, 2) (300,) (100, 2) (100,)
```

Z score: Normalization for compensating unit mismatching

In [85]:

```
from sklearn.preprocessing import StandardScaler
```

In [86]:

```
sc = StandardScaler()
```

In [87]:

```
X_train_std = sc.fit_transform(X_train)
X_train_std
```

Out[87]:

```
array([[ 1.8925893 ,  1.52189404],
       [ 0.1250379 ,  0.03213212],
       [ 0.9106163 , -1.31157471],
       [-1.34792161, -1.48684082],
       [-0.169554  , -0.58129926],
       [-0.56234321,  2.33980255],
       [ 1.0088136 , -1.19473064],
       [-0.75873781,  1.08372877],
       [ 2.1871812 , -1.04867555],
       [ 0.0268406 , -0.25997806],
       [-0.46414591, -1.1363086 ],
       [ 0.1250379 ,  0.03213212],
       [ 1.6961947 , -0.90262046],
       [ 1.1070109 , -0.90262046],
       [ 0.5178271 ,  1.22978386],
       [-1.05332971, -1.4576298 ],
       [-1.15152701, -1.54526286],
       [-0.0713567 ,  0.67477452],
       [ 0.4196298 , -0.46445519],
       [-0.2677513 , -0.25997806],
       [-0.85693511,  0.14897619],
       [ 0.0268406 ,  0.29503128],
       [ 0.7142217 , -1.28236369],
       [ 1.5979974 ,  1.11293979],
       [ 0.812419  , -1.36999675],
       [-1.44611891, -1.22394166],
       [-0.0713567 ,  0.14897619],
       [ 0.4196298 , -0.14313399],
       [-0.2677513 ,  0.03213212],
       [ 1.3034055 ,  2.22295848],
       [ 0.1250379 ,  0.76240757],
       [-1.34792161,  0.55793045],
       [ 1.9907866 ,  0.73319655],
       [-1.24972431, -1.39920777],
       [ 0.3214325 , -0.3184001 ],
       [-0.95513241,  0.55793045],
       [ 0.4196298 ,  0.29503128],
       [ 0.4196298 ,  1.11293979],
       [ 0.812419  ,  0.76240757],
       [ 0.9106163 ,  1.25899488],
       [-0.46414591, -1.22394166],
       [-1.83890811, -1.31157471],
       [ 1.1070109 ,  0.55793045],
       [-0.66054051, -1.60368489],
       [-0.75873781,  0.26582026],
       [ 1.0088136 ,  2.07690339],
       [-0.56234321,  1.37583895],
       [-0.0713567 ,  0.03213212],
       [-1.93710541,  0.47029739],
       [ 0.4196298 ,  0.26582026],
       [-1.05332971,  0.41187535],
       [ 0.2232352 , -0.14313399],
       [ 1.8925893 ,  0.11976517],
       [-1.15152701, -1.60368489],
       [-1.15152701,  0.29503128],
       [-0.85693511, -0.78577639],
       [-0.46414591,  2.31059153],
       [ 0.1250379 , -0.8149874 ],
       [ 1.5979974 ,  0.99609572],
       [-0.169554  , -1.07788657],
       [ 0.812419  , -1.10709759],
       [ 0.2232352 ,  2.1061144 ],
       [-0.0713567 , -0.23076704],
       [-0.85693511,  2.28138051],
       [-0.0713567 , -0.37682213],
       [-0.2677513 , -0.58129926],
       [ 0.4196298 , -0.49366621],
       [-0.2677513 , -0.93183148],
       [ 0.3214325 , -1.16551962],
       [ 0.2232352 ,  0.06134314],
       [-1.15152701, -1.60368489],
       [-0.66054051, -0.05550093],
       [-0.2677513 , -0.49366621],
       [-0.2677513 , -1.31157471],
       [-0.75873781,  0.55793045],
       [ 0.3214325 ,  0.06134314],
       [-0.95513241,  1.55110506],
       [ 0.812419  ,  0.35345332],
       [-1.54431621, -0.20155602],
       [ 0.7142217 , -1.39920777],
       [-0.75873781, -0.61051028],
       [-0.36594861, -1.31157471],
       [ 0.2232352 ,  0.14897619],
       [-0.56234321,  1.37583895],
       [-1.44611891,  0.35345332],
       [-1.15152701,  0.29503128],
       [ 1.0088136 ,  1.7847932 ],
       [ 2.0889839 ,  2.13532542],
       [-0.2677513 , -0.43524417],
       [-0.36594861, -0.78577639],
       [ 0.1250379 , -0.25997806],
       [-1.05332971,  0.76240757],
       [ 2.1871812 ,  0.38266434],
       [-1.34792161, -0.43524417],
       [ 1.9907866 ,  2.16453644],
       [ 1.4998001 ,  0.99609572],
       [-0.2677513 ,  0.26582026],
       [-0.169554  ,  0.85004063],
       [ 1.8925893 , -0.28918908],
       [-0.0713567 ,  1.96005931],
       [-0.46414591, -0.78577639],
       [ 0.3214325 ,  0.03213212],
       [ 1.4016028 , -1.42841878],
       [ 1.4016028 ,  2.33980255],
       [-0.0713567 ,  0.0029211 ],
       [-1.15152701,  0.41187535],
       [-1.15152701,  0.06134314],
       [-1.15152701, -0.52287722],
       [ 0.3214325 , -0.28918908],
       [-0.66054051, -0.11392297],
       [-0.0713567 ,  2.16453644],
       [ 0.0268406 , -0.25997806],
       [-0.66054051, -1.04867555],
       [ 0.4196298 ,  0.14897619],
       [ 0.812419  ,  1.37583895],
       [-0.169554  , -0.52287722],
       [ 0.0268406 ,  0.03213212],
       [ 1.1070109 ,  0.52871943],
       [ 1.8925893 , -1.07788657],
       [ 1.0088136 ,  1.98927033],
       [ 0.9106163 , -0.58129926],
       [-0.46414591, -0.02628992],
       [-0.0713567 ,  2.22295848],
       [-1.74071081,  0.35345332],
       [ 0.2232352 , -0.66893231],
       [-0.2677513 , -1.39920777],
       [-1.74071081, -0.99025351],
       [ 0.7142217 , -0.72735435],
       [-1.15152701, -0.78577639],
       [ 1.9907866 ,  0.90846266],
       [ 0.2232352 , -0.37682213],
       [ 0.0268406 ,  1.22978386],
       [-0.0713567 ,  0.29503128],
       [ 1.1070109 , -1.22394166],
       [-0.169554  ,  0.14897619],
       [ 0.3214325 ,  0.06134314],
       [-0.2677513 , -0.28918908],
       [ 0.5178271 ,  1.72637117],
       [ 0.3214325 ,  0.49950841],
       [ 0.0268406 ,  1.25899488],
       [ 1.9907866 , -1.36999675],
       [-1.15152701, -1.10709759],
       [-0.56234321, -1.51605184],
       [ 0.3214325 , -0.52287722],
       [-0.66054051, -1.51605184],
       [-0.2677513 ,  0.52871943],
       [ 1.0088136 , -1.01946453],
       [-1.05332971,  0.55793045],
       [-0.2677513 ,  0.79161859],
       [ 0.4196298 ,  0.09055416],
       [-1.64251351, -0.05550093],
       [ 0.3214325 ,  0.06134314],
       [-0.46414591, -0.28918908],
       [ 0.1250379 ,  1.87242626],
       [-0.95513241,  0.41187535],
       [ 0.812419  ,  0.52871943],
       [-1.34792161, -0.34761112],
       [-0.66054051,  1.40504997],
       [ 1.2052082 ,  0.52871943],
       [-1.74071081,  0.35345332],
       [-0.56234321,  1.90163728],
       [-1.74071081,  0.47029739],
       [-0.2677513 , -1.25315268],
       [-0.85693511, -0.78577639],
       [-1.64251351,  0.52871943],
       [-0.2677513 , -0.3184001 ],
       [ 0.7142217 , -1.10709759],
       [ 2.1871812 , -0.8149874 ],
       [-0.2677513 ,  0.61635248],
       [-0.2677513 , -0.75656537],
       [-1.93710541, -0.75656537],
       [ 0.4196298 ,  2.31059153],
       [ 0.7142217 ,  0.26582026],
       [ 0.2232352 , -0.28918908],
       [-1.34792161, -1.36999675],
       [-0.46414591, -0.55208824],
       [ 0.4196298 ,  0.0029211 ],
       [ 1.6961947 ,  1.75558219],
       [ 1.2052082 , -0.75656537],
       [ 1.0088136 ,  1.43426099],
       [-1.83890811,  0.17818721],
       [ 0.3214325 ,  0.06134314],
       [ 0.2232352 ,  0.03213212],
       [-1.05332971,  0.52871943],
       [-0.169554  ,  1.63873811],
       [-0.2677513 ,  0.14897619],
       [-0.2677513 , -0.34761112],
       [-0.36594861,  1.31741692],
       [-0.2677513 ,  0.09055416],
       [ 2.1871812 ,  1.11293979],
       [-1.24972431,  0.49950841],
       [-1.05332971, -0.46445519],
       [-1.64251351, -1.57447387],
       [-0.0713567 ,  0.11976517],
       [-0.2677513 , -0.90262046],
       [ 1.5979974 ,  0.0029211 ],
       [ 0.9106163 , -1.16551962],
       [ 0.0268406 , -0.58129926],
       [-0.2677513 ,  2.25216949],
       [-0.2677513 ,  0.20739823],
       [ 0.3214325 ,  0.26582026],
       [-0.0713567 , -1.07788657],
       [ 2.1871812 , -0.69814333],
       [-0.95513241, -0.3184001 ],
       [-1.44611891, -0.11392297],
       [-1.44611891, -0.20155602],
       [-0.75873781, -1.54526286],
       [-1.24972431, -1.07788657],
       [ 2.0889839 ,  0.38266434],
       [ 1.9907866 , -0.93183148],
       [-1.93710541,  0.35345332],
       [ 0.812419  , -0.3184001 ],
       [-1.05332971, -1.54526286],
       [ 1.9907866 , -0.66893231],
       [ 1.0088136 , -1.16551962],
       [ 1.1070109 , -0.14313399],
       [ 1.2052082 , -0.99025351],
       [ 1.4998001 ,  0.06134314],
       [ 0.2232352 , -0.37682213],
       [ 1.4016028 ,  1.2882059 ],
       [ 2.0889839 , -0.8149874 ],
       [-0.169554  , -0.20155602],
       [ 0.4196298 ,  0.99609572],
       [-0.0713567 , -0.52287722],
       [ 1.0088136 , -1.07788657],
       [ 2.0889839 , -1.19473064],
       [-0.0713567 ,  0.26582026],
       [ 0.2232352 , -0.25997806],
       [ 1.1070109 ,  0.11976517],
       [-1.24972431,  0.58714146],
       [-0.75873781, -1.60368489],
       [ 0.2232352 ,  0.23660925],
       [-1.34792161,  0.41187535],
       [ 0.0268406 , -0.58129926],
       [ 0.7142217 ,  1.7847932 ],
       [-1.64251351,  0.06134314],
       [-0.2677513 , -1.36999675],
       [-0.2677513 , -1.4576298 ],
       [-0.66054051,  0.55793045],
       [-0.75873781,  0.29503128],
       [ 0.9106163 , -0.66893231],
       [-1.05332971,  0.58714146],
       [-0.2677513 ,  0.06134314],
       [-0.75873781,  1.34662793],
       [ 0.1250379 ,  1.52189404],
       [-0.85693511,  0.38266434],
       [ 0.3214325 , -0.20155602],
       [ 1.0088136 ,  0.58714146],
       [ 0.0268406 , -0.3184001 ],
       [-0.56234321, -1.51605184],
       [ 0.1250379 ,  0.14897619],
       [-1.15152701,  0.3242423 ],
       [ 0.1250379 ,  1.05451775],
       [-1.54431621, -0.43524417],
       [-0.169554  ,  1.40504997],
       [ 2.1871812 , -0.8149874 ],
       [-0.95513241, -0.43524417],
       [ 1.3034055 ,  1.87242626],
       [ 1.1070109 , -1.22394166],
       [-0.169554  , -0.28918908],
       [ 1.794392  ,  0.99609572],
       [-1.05332971, -0.34761112],
       [-1.34792161, -1.10709759],
       [-0.36594861,  0.06134314],
       [-0.95513241, -1.10709759],
       [ 1.2052082 , -1.4576298 ],
       [-0.46414591, -0.84419842],
       [-0.85693511, -0.66893231],
       [-1.54431621, -1.51605184],
       [-0.75873781,  1.90163728],
       [ 0.9106163 ,  1.02530673],
       [ 0.812419  ,  0.26582026],
       [ 1.0088136 ,  1.87242626],
       [ 0.9106163 , -0.61051028],
       [ 1.1070109 ,  2.07690339],
       [-0.56234321,  0.87925164],
       [ 0.0268406 ,  0.03213212],
       [-1.83890811, -1.28236369],
       [-0.0713567 ,  0.20739823],
       [ 0.9106163 , -0.55208824],
       [ 0.2232352 , -0.37682213],
       [-0.169554  ,  1.6095271 ],
       [-1.74071081,  0.11976517],
       [-0.66054051, -0.34761112],
       [ 0.3214325 , -0.72735435],
       [ 0.4196298 , -0.46445519],
       [-0.95513241, -0.96104249],
       [ 0.1250379 ,  0.09055416],
       [-0.95513241,  0.44108637],
       [ 0.0268406 , -0.55208824],
       [ 0.9106163 , -0.78577639],
       [-0.0713567 ,  0.06134314],
       [ 1.1070109 , -0.99025351],
       [ 0.7142217 , -1.39920777],
       [-0.2677513 ,  0.06134314],
       [-1.34792161, -1.25315268],
       [-1.15152701, -1.01946453],
       [ 0.5178271 ,  1.84321524],
       [ 0.1250379 ,  0.20739823],
       [-0.56234321,  0.47029739]])
```

In [88]:

```
X_test_std = sc.transform(X_test)
X_test_std
```

Out[88]:

```
array([[ 0.812419  , -1.39920777],
       [ 2.0889839 ,  0.52871943],
       [-0.95513241, -0.75656537],
       [ 1.0088136 ,  0.76240757],
       [-0.85693511, -1.22394166],
       [-0.75873781, -0.23076704],
       [ 0.9106163 ,  1.08372877],
       [-0.85693511,  0.38266434],
       [ 0.2232352 ,  0.14897619],
       [ 0.4196298 , -0.14313399],
       [-0.2677513 , -0.14313399],
       [ 1.4998001 , -1.04867555],
       [-1.44611891, -0.6397213 ],
       [-1.74071081, -1.36999675],
       [-0.75873781,  0.49950841],
       [-0.2677513 ,  1.11293979],
       [ 1.4016028 , -0.93183148],
       [ 0.812419  ,  0.11976517],
       [ 0.1250379 , -0.8149874 ],
       [ 1.794392  , -0.28918908],
       [-1.54431621, -1.25315268],
       [-0.85693511,  0.29503128],
       [ 0.9106163 , -1.36999675],
       [ 2.0889839 ,  0.17818721],
       [-1.83890811, -1.48684082],
       [ 1.3034055 , -1.36999675],
       [ 0.4196298 ,  0.29503128],
       [-0.0713567 , -0.49366621],
       [ 1.6961947 ,  1.6095271 ],
       [-1.83890811, -1.42841878],
       [ 0.812419  , -0.84419842],
       [-1.83890811,  0.0029211 ],
       [-0.169554  ,  2.16453644],
       [-0.95513241,  0.26582026],
       [ 0.2232352 ,  1.08372877],
       [-0.2677513 ,  0.14897619],
       [-0.0713567 , -0.43524417],
       [ 0.0268406 , -0.14313399],
       [-1.15152701, -1.16551962],
       [-1.93710541, -0.05550093],
       [ 1.0088136 , -1.07788657],
       [-1.34792161, -0.43524417],
       [-1.93710541, -0.52287722],
       [ 0.9106163 , -1.4576298 ],
       [-1.74071081, -0.61051028],
       [ 0.6160244 ,  2.01848135],
       [-0.85693511, -0.25997806],
       [-0.66054051,  0.03213212],
       [ 1.0088136 , -0.84419842],
       [-0.36594861, -0.78577639],
       [-1.24972431,  0.26582026],
       [ 1.4998001 ,  0.35345332],
       [ 0.0268406 , -0.43524417],
       [-1.24972431,  0.29503128],
       [-0.0713567 ,  0.29503128],
       [-1.05332971, -1.1363086 ],
       [ 2.1871812 ,  0.93767368],
       [-1.15152701,  1.40504997],
       [-0.66054051,  0.11976517],
       [-0.66054051,  0.17818721],
       [ 0.3214325 , -0.55208824],
       [-0.2677513 , -0.37682213],
       [ 1.4016028 ,  0.58714146],
       [-0.95513241,  0.49950841],
       [-0.95513241, -0.3184001 ],
       [-1.05332971,  1.96005931],
       [ 0.4196298 ,  0.58714146],
       [ 0.9106163 ,  2.16453644],
       [ 0.1250379 , -0.3184001 ],
       [-0.46414591,  1.25899488],
       [ 1.4016028 ,  1.98927033],
       [-1.83890811,  0.44108637],
       [-1.05332971, -0.34761112],
       [-1.44611891, -1.4576298 ],
       [ 0.9106163 , -1.04867555],
       [-0.2677513 , -0.58129926],
       [ 1.794392  ,  1.84321524],
       [ 1.5979974 , -1.28236369],
       [-0.2677513 , -0.66893231],
       [-0.0713567 ,  0.23660925],
       [-1.05332971, -0.37682213],
       [ 0.812419  , -1.22394166],
       [ 0.1250379 ,  1.87242626],
       [ 0.6160244 , -0.90262046],
       [ 1.8925893 , -1.28236369],
       [-0.56234321,  1.46347201],
       [ 0.3214325 , -0.52287722],
       [ 1.0088136 ,  0.11976517],
       [-1.15152701,  0.47029739],
       [-1.54431621,  0.3242423 ],
       [ 1.1070109 ,  0.47029739],
       [-0.169554  , -0.46445519],
       [ 0.2232352 , -0.3184001 ],
       [ 0.3214325 ,  0.29503128],
       [-1.15152701, -1.57447387],
       [ 0.1250379 ,  0.26582026],
       [ 2.0889839 ,  1.75558219],
       [ 0.4196298 , -0.17234501],
       [ 1.4998001 ,  2.13532542],
       [-0.36594861,  1.22978386]])
```

Prediction and Classification Report

In [89]:

```
from sklearn.neighbors import KNeighborsClassifier
```

In [114]:

```
knn = KNeighborsClassifier(n_neighbors=5, metric="minkowski")

knn.fit(X_train, y_train)

y_predicted = knn.predict(X_test)
```

In [115]:

```
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix
```

In [116]:

```
print(classification_report(y_test, y_predicted))
```

```
              precision    recall  f1-score   support

           0       0.83      0.90      0.86        63
           1       0.81      0.68      0.74        37

    accuracy                           0.82       100
   macro avg       0.82      0.79      0.80       100
weighted avg       0.82      0.82      0.82       100
```

In [108]:

```
confusion_matrix(y_test, y_predicted)
```

Out[108]:

```
array([[59,  4],
       [13, 24]], dtype=int64)
```

In [109]:

```
accuracy = []
x_label = []
for k in np.arange(1,20):
    if k%2:
        knn = KNeighborsClassifier(n_neighbors=k, metric = "minkowski")
        x_label.append(k)
        knn.fit(X_train, y_train)

        y_predicted = knn.predict(X_test)
        accuracy.append(accuracy_score(y_test, y_predicted))
```

Which value of n_neighbour to take ?

In [110]:

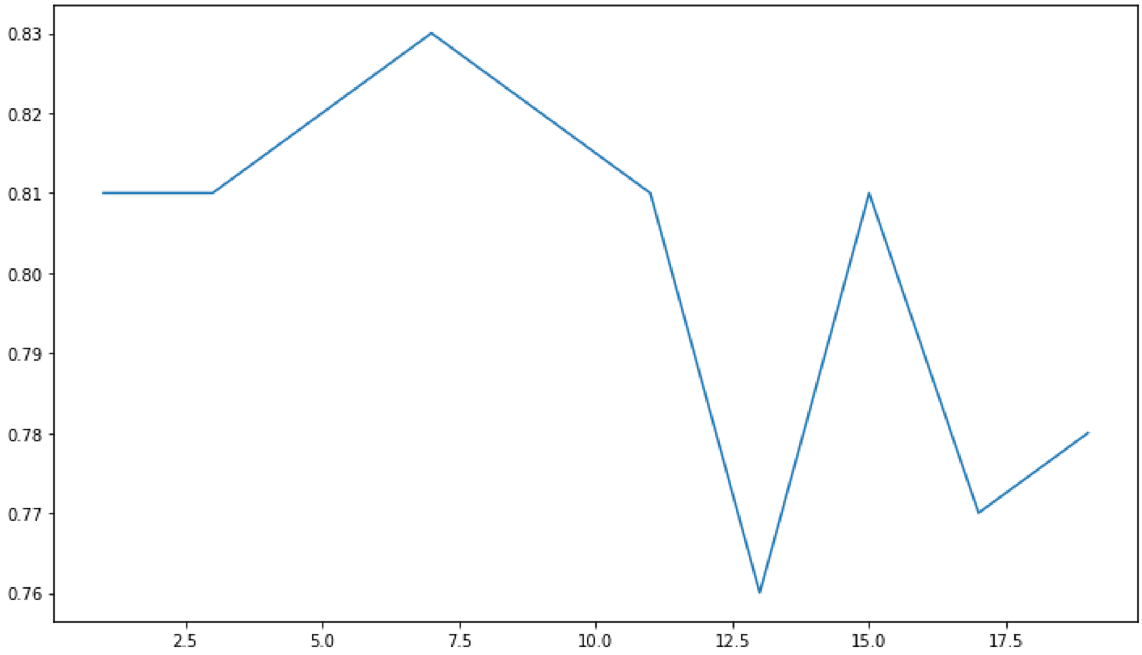
```
import matplotlib.pyplot as plt
```

In [111]:

```
plt.figure(figsize=(12,7))
plt.plot(x_label, accuracy)
```

Out[111]:

```
[<matplotlib.lines.Line2D at 0x20213888088>]
```



In [113]:

```
knn = KNeighborsClassifier(n_neighbors=7, metric="minkowski")

knn.fit(X_train, y_train)

y_predicted = knn.predict(X_test)
print(classification_report(y_test, y_predicted))
```

```
              precision    recall  f1-score   support

           0       0.82      0.94      0.87        63
           1       0.86      0.65      0.74        37

    accuracy                           0.83       100
   macro avg       0.84      0.79      0.81       100
weighted avg       0.83      0.83      0.82       100
```

The appropriate value for the n_neighbours will be 7

In [103]:

```
acc = []
metric = ["euclidean","manhattan", "minkowski" ]
for i in metric:
    knn = KNeighborsClassifier(n_neighbors=7, metric=i)

    knn.fit(X_train, y_train)

    y_predicted = knn.predict(X_test)
    acc.append(accuracy_score(y_test, y_predicted))
```

It is observed that all the metrics are giving nearly the same result

In [ ]:

In [ ]:

We use some other metric to evaluate the model based on the problem statement and data. Suppose, when one class is in majority, precission would not be a good measure to evaluate the model. We can use recall instead for more fraction of True values in our favour

Thank You

In [ ]: