

Pooja_Exp6

1.Data Collection

2. Data Preparation

In [23]:

```
#a.Load the data
import pandas as pd
data=pd.read_csv(r"C:\Users\avcoe\OneDrive\Desktop\DSV36\Titanic-Dataset.csv")
data
```

Out[23]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

```
In [24]: #b. select the relevant features
data=data[['Survived', 'Pclass', 'Sex', 'Age']]
data
```

```
Out[24]:
```

	Survived	Pclass	Sex	Age
0	0	3	male	22.0
1	1	1	female	38.0
2	1	3	female	26.0
3	1	1	female	35.0
4	0	3	male	35.0
...
886	0	2	male	27.0
887	1	1	female	19.0
888	0	3	female	NaN
889	1	1	male	26.0
890	0	3	male	32.0

891 rows × 4 columns

```
In [37]: # c. Handle Missing Values
data['Age'].fillna(data['Age'].median(),inplace=True)
print(data)
```

	Survived	Pclass	Sex	Age
0	0	3	male	22.0
1	1	1	female	38.0
2	1	3	female	26.0
3	1	1	female	35.0
4	0	3	male	35.0
..
886	0	2	male	27.0
887	1	1	female	19.0
888	0	3	female	28.0
889	1	1	male	26.0
890	0	3	male	32.0

[891 rows x 4 columns]

C:\Users\avcoe\AppData\Local\Temp\ipykernel_4976\3734871363.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
data['Age'].fillna(data['Age'].median(),inplace=True)
```

3. Split the data

```
In [28]: from sklearn.model_selection import train_test_split
X = data[['Pclass', 'Sex', 'Age']]
y = data['Survived']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
data
```

```
Out[28]:
```

	Survived	Pclass	Sex	Age
0	0	3	male	22.0
1	1	1	female	38.0
2	1	3	female	26.0
3	1	1	female	35.0
4	0	3	male	35.0
...
886	0	2	male	27.0
887	1	1	female	19.0
888	0	3	female	28.0
889	1	1	male	26.0
890	0	3	male	32.0

891 rows × 4 columns

4. Build the Model

```
In [32]: import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.metrics import accuracy_score, confusion_matrix
numeric_cols = X.select_dtypes(include=['int64', 'float64']).columns.tolist()
categorical_cols = X.select_dtypes(include=['object', 'category']).columns.tolist()
preprocessor = ColumnTransformer(
    transformers=[
        ('num', StandardScaler(), numeric_cols),
        ('cat', OneHotEncoder(drop='first'), categorical_cols)
    ]
)
pipeline = Pipeline(steps=[
    ('preprocessor', preprocessor),
    ('classifier', LogisticRegression())
])
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
pipeline.fit(X_train, y_train)
y_pred = pipeline.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)

print(f'Accuracy: {accuracy:.4f}')
print(f'Confusion Matrix:\n{conf_matrix}')
```

Accuracy: 0.8101
 Confusion Matrix:
 [[91 14]
 [20 54]]

5. Predict probabilities

In [36]:

```

y_pred = pipeline.predict(X_test)

probabilities = pipeline.predict_proba(X_test)[: , 1] # Probabilities for the positive class

accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)

print(f'Accuracy: {accuracy:.4f}')
print(f'Confusion Matrix:\n{conf_matrix}')
print(f'Predicted Probabilities of Survival:\n{probabilities}')

```

Accuracy: 0.8101

Confusion Matrix:

[[91 14]

[20 54]]

Predicted Probabilities of Survival:

```

[0.10069314 0.22679726 0.12063761 0.86809553 0.65517008 0.91801007
 0.57110169 0.13183442 0.64360795 0.93043301 0.41590649 0.06940741
 0.57110169 0.09618669 0.20530729 0.93520502 0.38542993 0.57110169
 0.24510084 0.35582614 0.11027096 0.43452248 0.62594992 0.12063761
 0.10069314 0.15027061 0.39752739 0.22679726 0.17077782 0.55234986
 0.12335741 0.60794992 0.47226166 0.57110169 0.12612974 0.10301596
 0.47226166 0.57110169 0.9100333 0.10069314 0.27437074 0.09841694
 0.10069314 0.10069314 0.46373219 0.13183442 0.12335741 0.11278702
 0.11027096 0.2946627 0.70482409 0.8526332 0.07626172 0.57947205
 0.04192386 0.93205875 0.22237533 0.85890076 0.75911598 0.62594992
 0.1179697 0.84637862 0.79010975 0.47226166 0.10069314 0.64318207
 0.27437074 0.09841694 0.12895521 0.8526332 0.79840773 0.93205875
 0.48493401 0.88948666 0.11278702 0.08373203 0.55548702 0.92533761
 0.74491271 0.48906271 0.13183442 0.78586771 0.88696577 0.10069314
 0.30532663 0.23095113 0.90793258 0.9100333 0.40975139 0.10069314
 0.15354185 0.68326265 0.27908143 0.10069314 0.10069314 0.10069314
 0.35002734 0.06778513 0.75444222 0.10301596 0.26401568 0.08570088
 0.93043301 0.07991721 0.09508858 0.10780421 0.73514322 0.35582614
 0.09618669 0.39752739 0.91410554 0.11929721 0.91410554 0.42208794
 0.47226166 0.12063761 0.23095113 0.20530729 0.81426161 0.44484926
 0.26437652 0.90578789 0.81398072 0.38689317 0.10780421 0.42829389
 0.86195014 0.49762575 0.57110169 0.1179697 0.57110169 0.10069314
 0.24043301 0.70482409 0.34427255 0.64941095 0.91410554 0.09841694
 0.07276022 0.63187598 0.1179697 0.71490063 0.24043301 0.20119532
 0.50175725 0.76372864 0.23128098 0.1076258 0.91410554 0.07106556
 0.10069314 0.11027096 0.12612974 0.57110169 0.10069314 0.10069314
 0.10069314 0.57110169 0.75444222 0.54606329 0.11152279 0.47226166
 0.19315705 0.91410554 0.10069314 0.25911126 0.26437652 0.91410554
 0.11535305 0.08570088 0.5039738 0.82913552 0.42208794 0.68326265
 0.12895521 0.10069314 0.50810469 0.83270287 0.71007933]

```

In []: