

AFAME TECHNOLOGIES: TITANIC SURVIVAL PREDICTION

Importing Libraries


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

Dataset: TITANIC SURVIVAL PREDICTION

```
data=pd.read_csv("Titanic-Dataset.csv")
```

data.head(10)

	PassengerId int64 1 - 10	Survived int64 0 - 1	Pclass int64 1 - 3	Name object Braund, Mr. 10% Cumings, Mr... .. 10% 8 others 80%	Sex object male 50% female 50%	Age float64 2.0 - 54.0	SibSp int64 0 - 3	P
0	1	0	3	Braund, Mr. Owen...	male	22	1	0
1	2	1	1	Cumings, Mrs. Jo...	female	38	1	1
2	3	1	3	Heikkinen, Miss. ...	female	26	0	0
3	4	1	1	Futrelle, Mrs. Jac...	female	35	1	1
4	5	0	3	Allen, Mr. William ...	male	35	0	0
5	6	0	3	Moran, Mr. James	male	nan	0	0
6	7	0	1	McCarthy, Mr. Ti...	male	54	0	0
7	8	0	3	Palsson, Master. ...	male	2	3	3
8	9	1	3	Johnson, Mrs. Os...	female	27	0	0
9	10	1	2	Nasser, Mrs. Nich...	female	14	1	1

10 rows, 12 cols, showing 10 rows/page << < Page 1 of 1 > >> 

Categorical Columns - [Survived, PClass, Sex, SibSp, Parch, Embarked]

Numerical Columns - [Age, Fare, PassengerId]

Mixed Columns - [Name, Ticket, Cabin]

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
#   Column      Non-Null Count  Dtype
---  -
0   PassengerId  891 non-null    int64
1   Survived     891 non-null    int64
2   Pclass       891 non-null    int64
3   Name         891 non-null    object
4   Sex          891 non-null    object
5   Age          714 non-null    float64
6   SibSp        891 non-null    int64
7   Parch        891 non-null    int64
8   Ticket       891 non-null    object
9   Fare         891 non-null    float64
10  Cabin        204 non-null    object
11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

```
data.shape
```

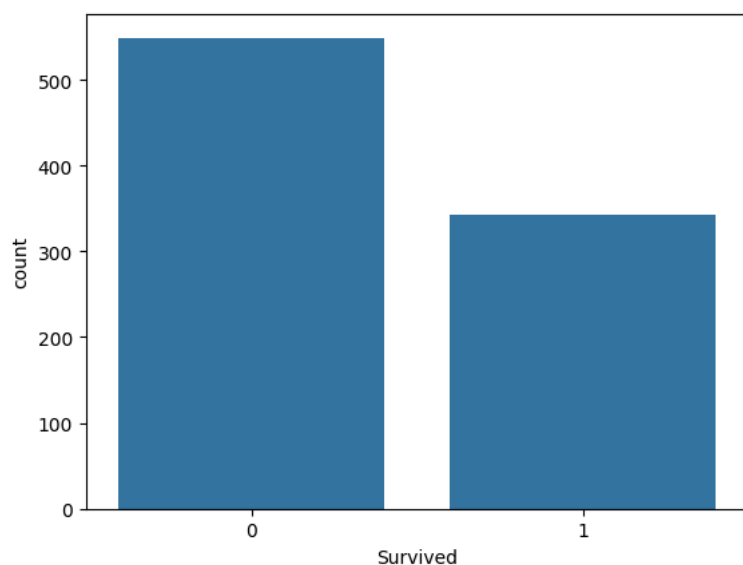
```
(891, 12)
```

```
data['Survived'].value_counts()
```

```
Survived
0      549
1      342
Name: count, dtype: int64
```

Data Visualization

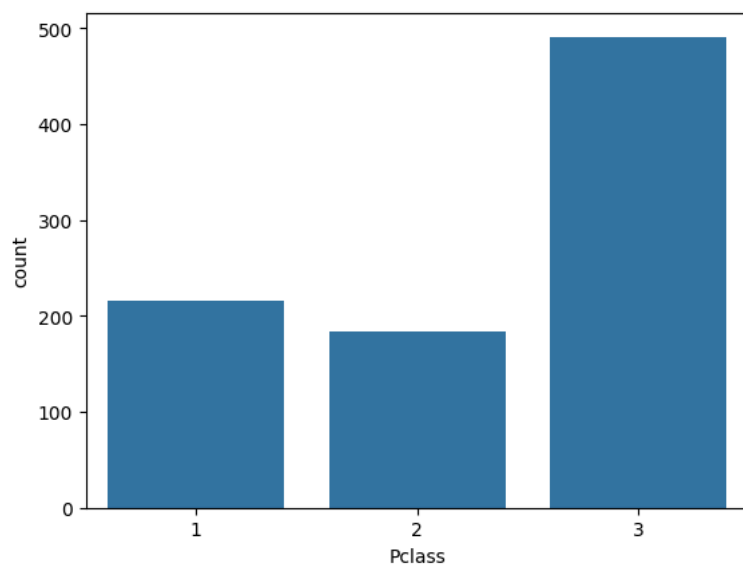
```
sns.countplot(data=data, x='Survived')
death_percent=round((data['Survived'].value_counts().values[0]/891)*100)
```



```
print((data['Pclass'].value_counts()/891)*100)
sns.countplot(data=data, x='Pclass')
```

```
Pclass
3    55.106622
1    24.242424
2    20.650954
Name: count, dtype: float64
```

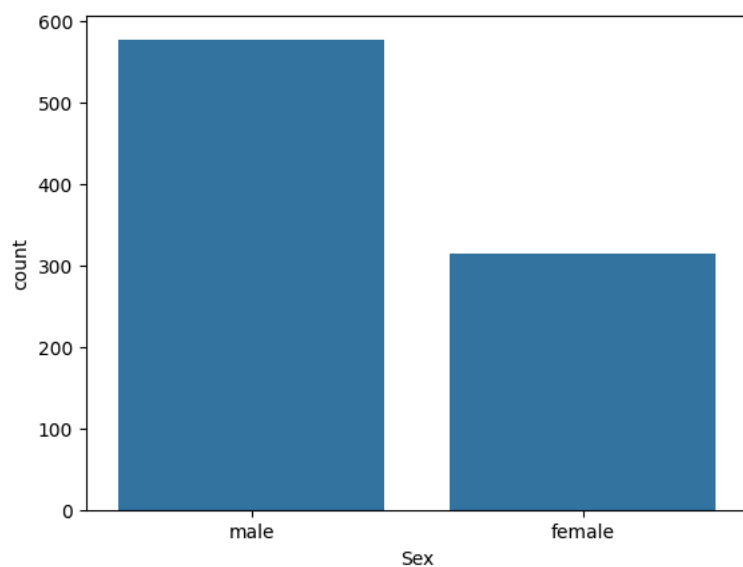
```
<AxesSubplot: xlabel='Pclass', ylabel='count'>
```



```
print((data['Sex'].value_counts()/891)*100)
sns.countplot(data=data, x='Sex')
```

```
Sex
male    64.758698
female  35.241302
Name: count, dtype: float64
```

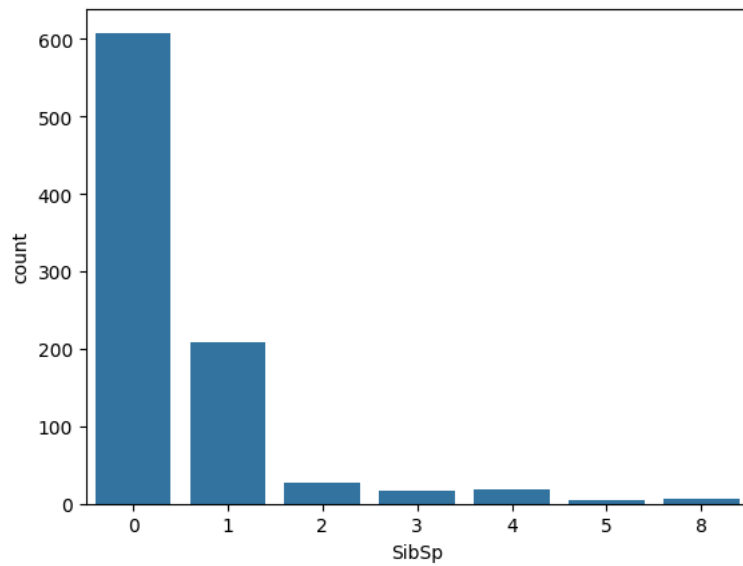
```
<AxesSubplot: xlabel='Sex', ylabel='count'>
```



```
print(data['SibSp'].value_counts())  
sns.countplot(data = data, x='SibSp')
```

```
SibSp  
0    608  
1    209  
2     28  
4     18  
3     16  
8      7  
5      5  
Name: count, dtype: int64
```

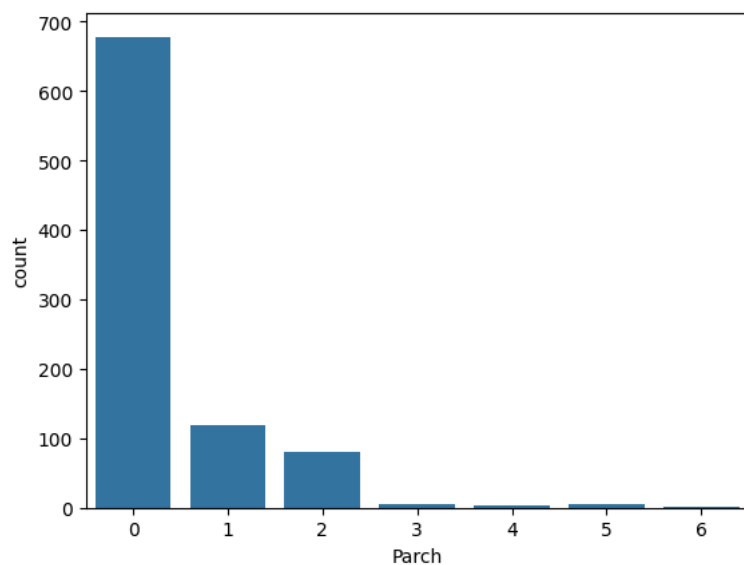
```
<AxesSubplot: xlabel='SibSp', ylabel='count'>
```



```
print((data['Parch'].value_counts()/891)*100)
sns.countplot(data=data, x='Parch')
```

```
Parch
0    76.094276
1    13.243547
2     8.978676
5     0.561167
3     0.561167
4     0.448934
6     0.112233
Name: count, dtype: float64
```

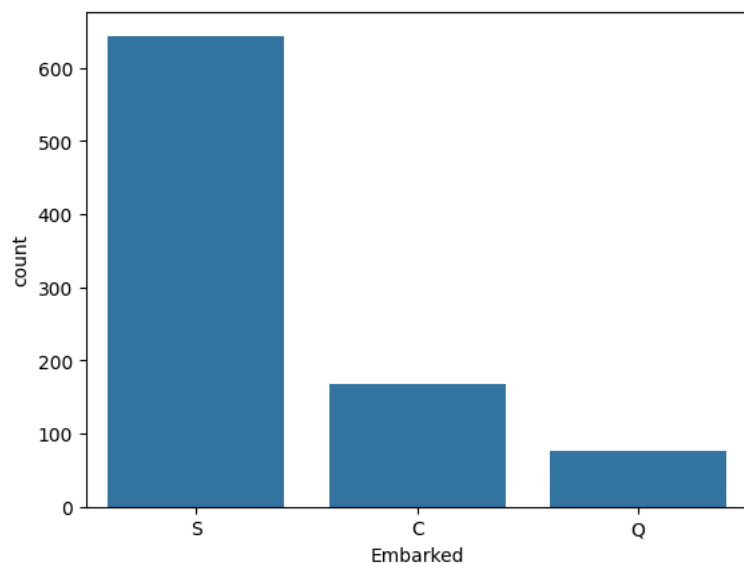
<AxesSubplot: xlabel='Parch', ylabel='count'>



```
print((data['Embarked'].value_counts()/891)*100)
sns.countplot(data=data, x='Embarked')
```

```
Embarked
S    72.278339
C    18.855219
Q     8.641975
Name: count, dtype: float64
```

<AxesSubplot: xlabel='Embarked', ylabel='count'>



```
sns.distplot(data['Age'])
print(data['Age'].skew())
print(data['Age'].kurt())
```

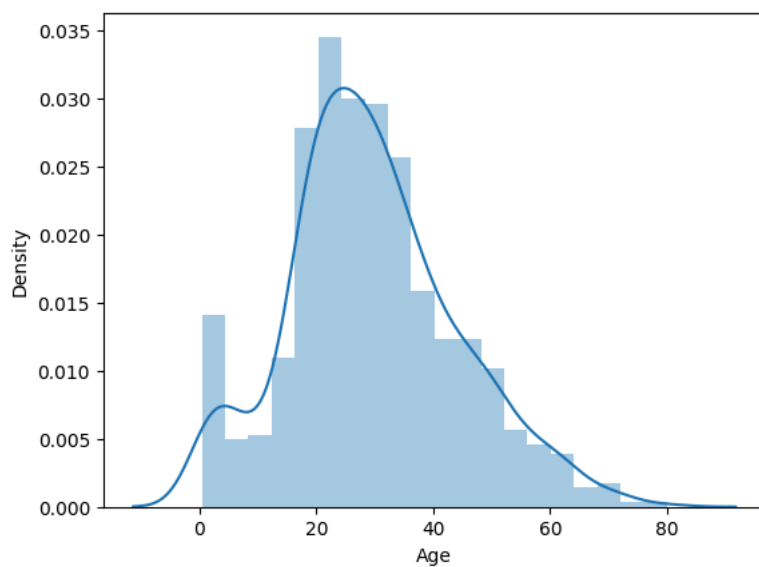
/tmp/ipykernel_259/1333658756.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

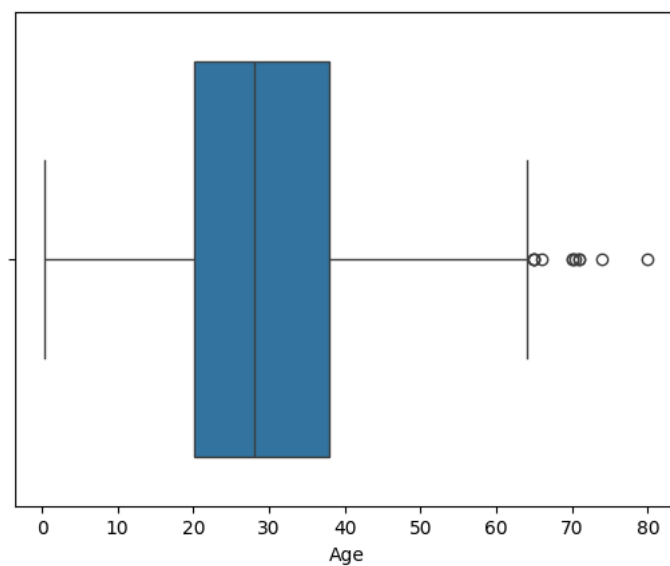
For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(data['Age'])
0.38910778230082704
0.17827415364210353
```



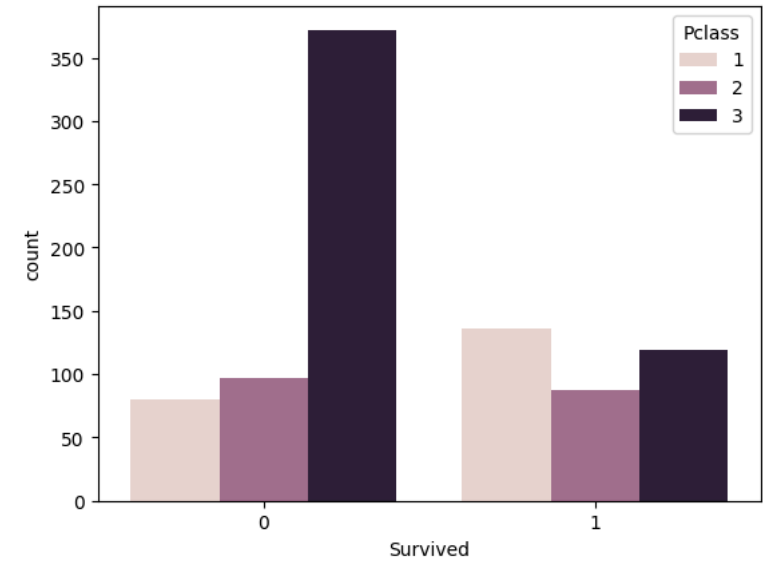
```
sns.boxplot(data, x = 'Age')
```

<AxesSubplot: xlabel='Age'>



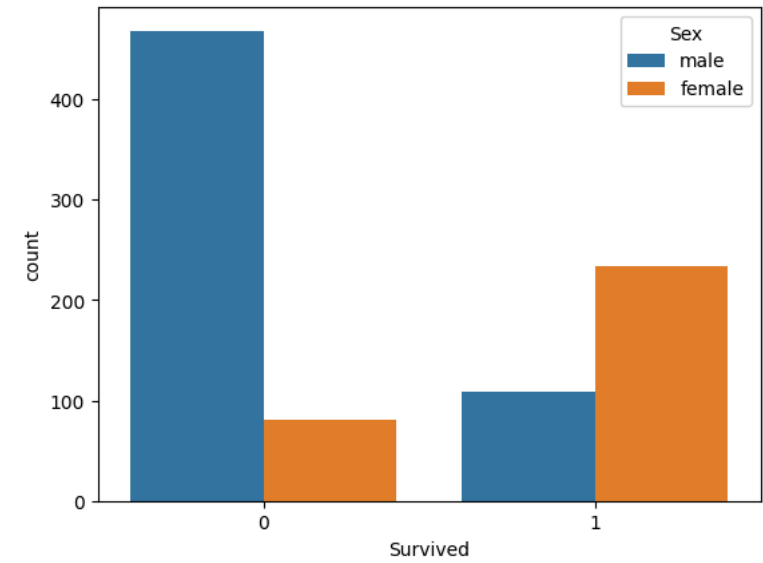
```
sns.countplot(data, x='Survived', hue='Pclass')
pd.crosstab(data['Pclass'], data['Survived']).apply(lambda r: round((r/r.sum())*100,1), axis=1)
```

	0 float64	1 float64	
1	37	63	
2	52.7	47.3	
3	75.8	24.2	



```
sns.countplot(data, x='Survived', hue='Sex')
pd.crosstab(data['Sex'], data['Survived']).apply(lambda r: round((r/r.sum())*100,1), axis=1)
```

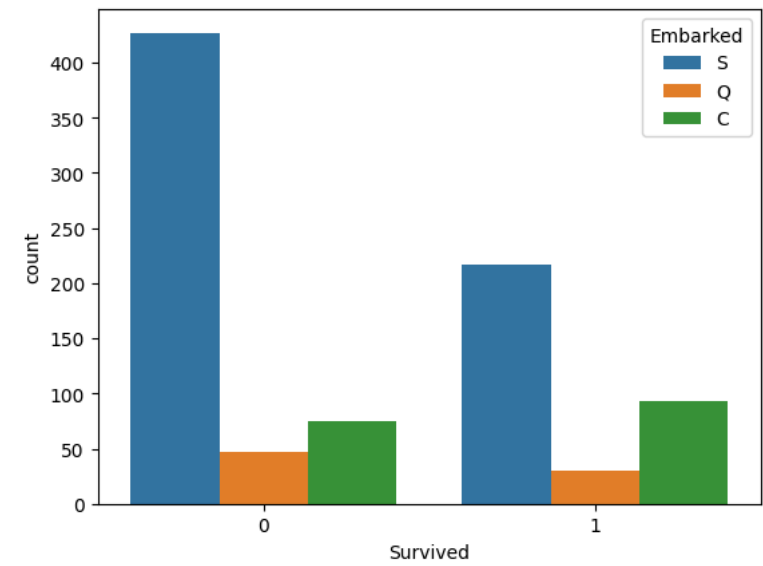
	0 float64	1 float64	
fe...	25.8	74.2	
male	81.1	18.9	



```
sns.countplot(data, x='Survived', hue='Embarked')

pd.crosstab(data['Embarked'], data['Survived']).apply(lambda r: round((r/r.sum())*100,1), axis=1)
```

	0 float64	1 float64	
C	44.6	55.4	
Q	61	39	
S	66.3	33.7	




```
plt.figure(figsize=(15,6))
sns.distplot(data[data['Survived']==0]['Age'])
sns.distplot(data[data['Survived']==1]['Age'])
```

/tmp/ipykernel_259/2170415421.py:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see

<https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(data[data['Survived']==0]['Age'])
```

/tmp/ipykernel_259/2170415421.py:3: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

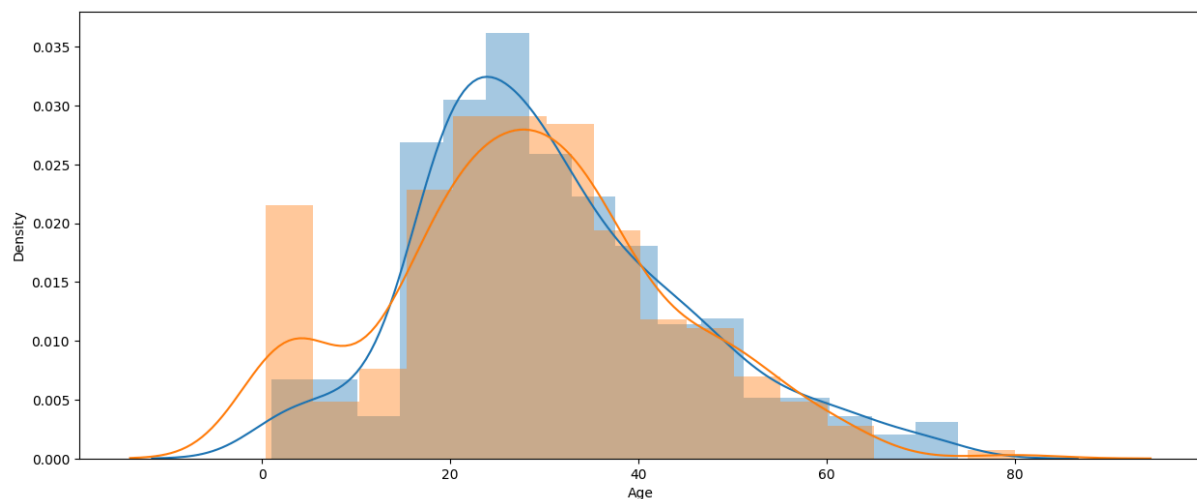
Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see

<https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(data[data['Survived']==1]['Age'])
```

<AxesSubplot: xlabel='Age', ylabel='Density'>



```
plt.figure(figsize=(15,6))
sns.distplot(data[data['Survived']==0]['Fare'])
sns.distplot(data[data['Survived']==1]['Fare'])
```

/tmp/ipykernel_259/1402514286.py:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(data[data['Survived']==0]['Fare'])
/tmp/ipykernel_259/1402514286.py:3: UserWarning:
```

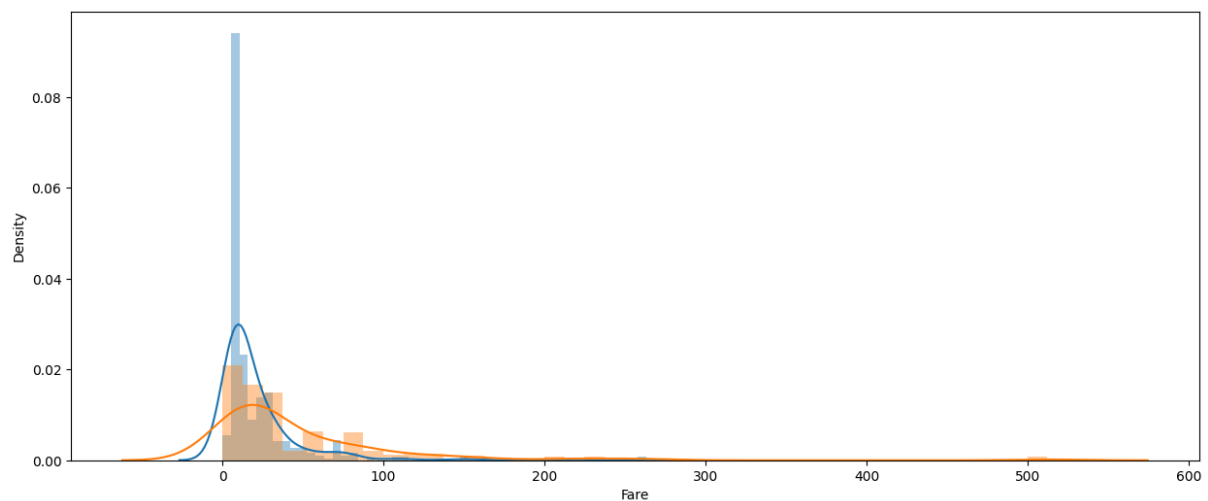
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(data[data['Survived']==1]['Fare'])
```

<AxesSubplot: xlabel='Fare', ylabel='Density'>



```
from sklearn import preprocessing as pp
```

```
data['Fare'].head()
```

```
0    7.2500
1   71.2833
2    7.9250
3   53.1000
4    8.0500
Name: Fare, dtype: float64
```

```
data_scaler=pp.MinMaxScaler(feature_range=(0,1))
fare_arr=data[['Fare']]
fare_arr
```

	Fare float64 0.0 - 512.3292	
0	7.25	
1	71.2833	
2	7.925	
3	53.1	
4	8.05	
5	8.4583	
6	51.8625	
7	21.075	
8	11.1333	
9	30.0708	

891 rows, 1 col, showing 10 rows/page << < Page 1 of 90 > >> [↓](#)

```
fare_scaled=data_scaler.fit_transform(fare_arr)
```

fare_scaled

```
array([[0.01415106],
       [0.13913574],
       [0.01546857],
       [0.1036443 ],
       [0.01571255],
       [0.0165095 ],
       [0.10122886],
       [0.04113566],
       [0.02173075],
       [0.05869429],
       [0.03259623],
       [0.05182215],
       [0.01571255],
       [0.06104473],
       [0.01533038],
       [0.03122992],
       [0.05684821],
       [0.02537431],
       [0.03513366],
       [0.01410226],
       [0.05074862],
       [0.02537431],
       [0.01567195],
       [0.06929139],
       [0.04113566],
       [0.06126432],
       [0.01410226],
       [0.51334181],
       [0.01537917],
       [0.01541158],
       [0.0541074 ],
       [0.28598956],
```

```
data['fare_scaled']=fare_scaled
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 13 columns):
#   Column      Non-Null Count  Dtype
---  -
0   PassengerId  891 non-null    int64
1   Survived     891 non-null    int64
2   Pclass       891 non-null    int64
3   Name         891 non-null    object
4   Sex          891 non-null    object
5   Age          714 non-null    float64
6   SibSp        891 non-null    int64
7   Parch        891 non-null    int64
8   Ticket       891 non-null    object
9   Fare         891 non-null    float64
10  Cabin        204 non-null    object
11  Embarked     889 non-null    object
12  fare_scaled  891 non-null    float64
dtypes: float64(3), int64(5), object(5)
memory usage: 90.6+ KB
```

```
col_to_drop=['Name', 'Cabin']
data=data.drop(col_to_drop,axis=1)
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 11 columns):
#   Column      Non-Null Count  Dtype
---  -
0   PassengerId  891 non-null    int64
1   Survived     891 non-null    int64
2   Pclass       891 non-null    int64
3   Sex          891 non-null    object
4   Age          714 non-null    float64
5   SibSp        891 non-null    int64
6   Parch        891 non-null    int64
7   Ticket       891 non-null    object
8   Fare         891 non-null    float64
9   Embarked     889 non-null    object
10  fare_scaled  891 non-null    float64
dtypes: float64(3), int64(5), object(3)
memory usage: 76.7+ KB
```

```
data=data.dropna()
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 712 entries, 0 to 890
Data columns (total 11 columns):
#   Column      Non-Null Count  Dtype
---  -
0   PassengerId  712 non-null    int64
1   Survived     712 non-null    int64
2   Pclass       712 non-null    int64
3   Sex          712 non-null    object
4   Age          712 non-null    float64
5   SibSp        712 non-null    int64
6   Parch        712 non-null    int64
7   Ticket       712 non-null    object
8   Fare         712 non-null    float64
9   Embarked     712 non-null    object
10  fare_scaled  712 non-null    float64
dtypes: float64(3), int64(5), object(3)
memory usage: 66.8+ KB
```

```
pip install xgboost
```

Requirement already satisfied: xgboost in /usr/local/lib/python3.9/site-packages (2.0.3)
Requirement already satisfied: numpy in /shared-libs/python3.9/py/lib/python3.9/site-packages (from xgboost) (1.23.4)
Requirement already satisfied: scipy in /shared-libs/python3.9/py/lib/python3.9/site-packages (from xgboost) (1.9.3)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is rec

[notice] A new release of pip is available: 23.0.1 -> 24.0
[notice] To update, run: python -m pip install --upgrade pip
Note: you may need to restart the kernel to use updated packages.

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
from sklearn.preprocessing import LabelEncoder
import numpy as np
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from xgboost import XGBClassifier
from sklearn.model_selection import train_test_split
```

```
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
data['Sex'] = label_encoder.fit_transform(data['Sex'])
print(data['Sex'].unique())
```

```
[1 0]
```

```
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
data['Embarked'] = label_encoder.fit_transform(data['Embarked'])
print(data['Embarked'].unique())
```

```
[2 0 1]
```

```
data.head()
```

	PassengerId int64	Survived int64	Pclass int64	Sex int64	Age float64	SibSp int64	Parch int64	T
0	1	0	3	1	22	1	0	A
1	2	1	1	0	38	1	0	F
2	3	1	3	0	26	0	0	S
3	4	1	1	0	35	1	0	1
4	5	0	3	1	35	0	0	3

5 rows, 11 cols, showing 10 rows/page

<< < Page 1 of 1 > >>



```
x=data.drop(['Survived', 'Ticket', 'Fare', 'PassengerId'],axis=1)
y=data['Survived']
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
X_train.shape,X_test.shape
```

```
((569, 7), (143, 7))
```

x.head()

	Pclass int64	Sex int64	Age float64	SibSp int64	Parch int64	Embarked int64	fare_scaled float64	
0	3	1	22	1	0	2	0.01415105756	
1	1	0	38	1	0	0	0.1391357354	
2	3	0	26	0	0	2	0.01546856982	
3	1	0	35	1	0	2	0.1036442975	
4	3	1	35	0	0	2	0.01571255357	

5 rows, 7 cols, showing 10 rows/page

<< < Page 1 of 1 > >>

↓

ML Model Learning

```

models = {
    'KNN': KNeighborsClassifier(),
    'SVM': SVC(),
    'BN': GaussianNB(),
    'RBF': RandomForestClassifier(),
    'DT': DecisionTreeClassifier(),
    'XGBoost': XGBClassifier()
}

results = {}

for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    tn, fp, fn, tp = confusion_matrix(y_test, y_pred).ravel()
    sensitivity = tp / (tp + fn)
    specificity = tn / (tn + fp)
    f1 = (2*tp)/(2*tp + fp + fn)
    precision = tp/(tp+fp)

    results[name] = {
        'Accuracy': accuracy,
        'Sensitivity': sensitivity, #recall
        'Specificity': specificity,
        'F1 Score': f1,
        'Precision': precision,
    }

# Print the results
for name, result in results.items():
    print(f"Model: {name}")
    print(f"Accuracy: {result['Accuracy']}")
    print(f"Precision: {result['Precision']}")
    print(f"Sensitivity: {result['Sensitivity']}")
    print(f"Specificity: {result['Specificity']}")
    print(f"F1 Score: {result['F1 Score']}")
    print()

```

Model: KNN
 Accuracy: 0.7412587412587412
 Precision: 0.75
 Sensitivity: 0.5901639344262295
 Specificity: 0.8536585365853658
 F1 Score: 0.6605504587155964

Model: SVM
 Accuracy: 0.6083916083916084
 Precision: 0.7777777777777778
 Sensitivity: 0.11475409836065574
 Specificity: 0.975609756097561
 F1 Score: 0.2

Model: BN
 Accuracy: 0.7482517482517482
 Precision: 0.7450980392156863
 Sensitivity: 0.6229508196721312
 Specificity: 0.8414634146341463
 F1 Score: 0.6785714285714286

Model: RBF
 Accuracy: 0.7832167832167832
 Precision: 0.7419354838709677
 Sensitivity: 0.7540983606557377
 Specificity: 0.8048780487804879
 F1 Score: 0.7479674796747967

Model: DT
 Accuracy: 0.7762237762237763

```
!pip install lightgbm==4.3.0
```

Requirement already satisfied: lightgbm==4.3.0 in /root/venv/lib/python3.9/site-packages (4.3.0)

Requirement already satisfied: numpy in /shared-lib/python3.9/py/lib/python3.9/site-packages (from lightgbm==4.3.0) (1.23.4)

Requirement already satisfied: scipy in /shared-lib/python3.9/py/lib/python3.9/site-packages (from lightgbm==4.3.0) (1.9.3)

[notice] A new release of pip is available: 23.0.1 -> 24.0

[notice] To update, run: `pip install --upgrade pip`

```
!pip install catboost==1.2.5
```

Requirement already satisfied: catboost==1.2.5 in /root/venv/lib/python3.9/site-packages (1.2.5)

Requirement already satisfied: matplotlib in /shared-lib/python3.9/py/lib/python3.9/site-packages (from catboost==1.2.5) (3.6.0)

Requirement already satisfied: pandas>=0.24 in /shared-lib/python3.9/py/lib/python3.9/site-packages (from catboost==1.2.5) (2.1.4)

Requirement already satisfied: six in /shared-lib/python3.9/py-core/lib/python3.9/site-packages (from catboost==1.2.5) (1.16.0)

Requirement already satisfied: plotly in /shared-lib/python3.9/py/lib/python3.9/site-packages (from catboost==1.2.5) (5.10.0)

Requirement already satisfied: numpy>=1.16.0 in /shared-lib/python3.9/py/lib/python3.9/site-packages (from catboost==1.2.5) (1.23.4)

Requirement already satisfied: scipy in /shared-lib/python3.9/py/lib/python3.9/site-packages (from catboost==1.2.5) (1.9.3)

Requirement already satisfied: graphviz in /root/venv/lib/python3.9/site-packages (from catboost==1.2.5) (0.20.3)

Requirement already satisfied: tzdata>=2022.1 in /shared-lib/python3.9/py/lib/python3.9/site-packages (from pandas>=0.24->catboost==1.2.5) (2023.3)

Requirement already satisfied: pytz>=2020.1 in /shared-lib/python3.9/py/lib/python3.9/site-packages (from pandas>=0.24->catboost==1.2.5) (2023.3)

Requirement already satisfied: python-dateutil>=2.8.2 in /shared-lib/python3.9/py-core/lib/python3.9/site-packages (from pandas>=0.24->catboost==1.2.5) (2.8.2)

Requirement already satisfied: pillow>=6.2.0 in /shared-lib/python3.9/py/lib/python3.9/site-packages (from matplotlib->catboost==1.2.5) (9.5.0)

Requirement already satisfied: pyparsing>=2.2.1 in /shared-lib/python3.9/py-core/lib/python3.9/site-packages (from matplotlib->catboost==1.2.5) (3.1.0)

Requirement already satisfied: contourpy>=1.0.1 in /shared-lib/python3.9/py/lib/python3.9/site-packages (from matplotlib->catboost==1.2.5) (1.0.7)

Requirement already satisfied: kiwisolver>=1.0.1 in /shared-lib/python3.9/py/lib/python3.9/site-packages (from matplotlib->catboost==1.2.5) (1.4.5)

Requirement already satisfied: packaging>=20.0 in /shared-lib/python3.9/py-core/lib/python3.9/site-packages (from matplotlib->catboost==1.2.5) (23.1)

Requirement already satisfied: cycler>=0.10 in /shared-lib/python3.9/py/lib/python3.9/site-packages (from matplotlib->catboost==1.2.5) (0.12.1)

Requirement already satisfied: fonttools>=4.22.0 in /shared-lib/python3.9/py/lib/python3.9/site-packages (from matplotlib->catboost==1.2.5) (4.42.0)

Requirement already satisfied: tenacity>=6.2.0 in /shared-lib/python3.9/py/lib/python3.9/site-packages (from plotly->catboost==1.2.5) (8.1.0)

[notice] A new release of pip is available: 23.0.1 -> 24.0

[notice] To update, run: `pip install --upgrade pip`


```

from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier, ExtraTreesClassifier
from sklearn.tree import DecisionTreeClassifier
from lightgbm import LGBMClassifier
from sklearn.linear_model import RidgeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import GradientBoostingClassifier
from catboost import CatBoostClassifier

models = {
    'ET': ExtraTreesClassifier(),
    'LIGHTGBM': LGBMClassifier(),
    'RC': RidgeClassifier(),
    'LR': LogisticRegression(),
    'gb': GradientBoostingClassifier(),
    'catboost': CatBoostClassifier()
}

results = {}

for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    tn, fp, fn, tp = confusion_matrix(y_test, y_pred).ravel()
    sensitivity = tp / (tp + fn)
    specificity = tn / (tn + fp)
    f1 = (2*tp)/(2*tp + fp + fn)
    precision = tp/(tp+fp)

    results[name] = {
        'Accuracy': accuracy,
        'Sensitivity': sensitivity,
        'Specificity': specificity,
        'F1 Score': f1,
        'Precision': precision,
    }

for name, result in results.items():
    print(f"Model: {name}")
    print(f"Accuracy: {result['Accuracy']}")
    print(f"Precision: {result['Precision']}")
    print(f"Sensitivity: {result['Sensitivity']}")
    print(f"Specificity: {result['Specificity']}")
    print(f"F1 Score: {result['F1 Score']}")

print()

```

```

43:  learn: 0.5328831      total: 63.9ms  remaining: 1.39s
44:  learn: 0.5309930      total: 64.2ms  remaining: 1.36s
45:  learn: 0.5284940      total: 64.6ms  remaining: 1.34s
46:  learn: 0.5266754      total: 64.9ms  remaining: 1.32s
47:  learn: 0.5250348      total: 65.2ms  remaining: 1.29s
48:  learn: 0.5229713      total: 65.6ms  remaining: 1.27s
49:  learn: 0.5210659      total: 66.1ms  remaining: 1.25s
50:  learn: 0.5185612      total: 66.5ms  remaining: 1.24s
51:  learn: 0.5160697      total: 66.9ms  remaining: 1.22s
52:  learn: 0.5137796      total: 67.4ms  remaining: 1.2s
53:  learn: 0.5118060      total: 67.8ms  remaining: 1.19s
54:  learn: 0.5096194      total: 68.3ms  remaining: 1.17s
55:  learn: 0.5079247      total: 68.5ms  remaining: 1.16s
56:  learn: 0.5061396      total: 68.9ms  remaining: 1.14s
57:  learn: 0.5040817      total: 69.3ms  remaining: 1.13s
58:  learn: 0.5018588      total: 69.8ms  remaining: 1.11s
59:  learn: 0.5002951      total: 70.2ms  remaining: 1.1s
60:  learn: 0.4989168      total: 70.7ms  remaining: 1.09s
61:  learn: 0.4973072      total: 71.1ms  remaining: 1.07s
62:  learn: 0.4965350      total: 71.3ms  remaining: 1.06s
63:  learn: 0.4943331      total: 71.7ms  remaining: 1.05s
64:  learn: 0.4929567      total: 72.1ms  remaining: 1.04s
65:  learn: 0.4917233      total: 72.5ms  remaining: 1.02s
66:  learn: 0.4910079      total: 72.7ms  remaining: 1.01s
67:  learn: 0.4892250      total: 73.2ms  remaining: 1s
68:  learn: 0.4873252      total: 73.6ms  remaining: 993ms
69:  learn: 0.4861628      total: 73.9ms  remaining: 981ms
70:  learn: 0.4847838      total: 74.3ms  remaining: 972ms
71:  learn: 0.4831270      total: 74.7ms  remaining: 963ms
72:  learn: 0.4817435      total: 75.1ms  remaining: 954ms

```

Among different ML algorithms, "GradientBoostingClassifier" outperforms with the highest accuracy.

```
new_passenger = [[0,0,43.0,1,0,1,0.103644]]
model=KNeighborsClassifier()
model.fit(X_train,y_train)
model.predict(X_test)
prediction = model.predict(new_passenger)
print("Survived Prediction:", prediction[0])
```

Survived Prediction: 1

/shared-ls/python3.9/py/lib/python3.9/site-packages/sklearn/base.py:450: UserWarning: X does not have valid feature names, but KNeighborsCl
warnings.warn(