Titanic Dataset

Importing Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
from sklearn.preprocessing import LabelEncoder
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.metrics import classification_report
import warnings
warnings.filterwarnings('ignore')
```

Load dataset

```
titanic = pd.read_csv('tested.csv')
```

Understanding of data

```
titanic.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S

```
titanic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Survived     418 non-null    int64  
 2   Pclass       418 non-null    int64  
 3   Name         418 non-null    object 
 4   Sex          418 non-null    object 
 5   Age          332 non-null    float64
 6   SibSp        418 non-null    int64  
 7   Parch        418 non-null    int64  
 8   Ticket       418 non-null    object 
 9   Fare         417 non-null    float64
 10  Cabin        91 non-null     object 
 11  Embarked     418 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 39.3+ KB
```

Data Cleaning

```
# Checking null values
titanic.isna().sum()
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

```
# Handling the null values
columns = ['Age', 'Fare']
for col in columns:
    titanic[col].fillna(titanic[col].median(), inplace = True)
titanic['Cabin'].fillna('Unknown', inplace=True)
```

```
#checking duplicate values
dup = titanic.duplicated().sum()
print("The number of duplicated values in the dataset are: ", dup)
```

```
The number of duplicated values in the dataset are:  0
```

```
#Checking if there are any typos
for col in titanic.select_dtypes(include = "object"):
    print(f"Name of Column: {col}")
    print(titanic[col].unique())
    print('\n', '-'*60, '\n')
```

```
 'C54' 'C97' 'D22' 'B10' 'F4' 'E45' 'E52' 'D30' 'B58 B60' 'E34' 'C62 C64'
 'A11' 'B11' 'C80' 'F33' 'C85' 'D37' 'C86' 'D21' 'C89' 'F E46' 'A34' 'D'
 'B26' 'C22 C26' 'B69' 'C32' 'B78' 'F E57' 'F2' 'A18' 'C106' 'B51 B53 B55'
 'D10 D12' 'E60' 'E50' 'E39 E41' 'B52 B54 B56' 'C39' 'B24' 'D28' 'B41'
 'C7' 'D40' 'D38' 'C105']

 ------------------------------------------------------------

Name of Column: Embarked
['Q' 'S' 'C']

 ------------------------------------------------------------
```

Insights:

- The following columns: **Age, Fare, and Cabin had null values** in the dataset
- The null values in **Age and Fare** column were filled with **median** instead of mean due to the presence of outliers
- The null values in **Cabin** column were filled with **Unknown**
- Later, we checked the unique values inside **Categorical Columns** to see if there are any **typos or useful information**

Feature Engineering

```
titanic.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	Unknown	Q
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	Unknown	S
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	Unknown	Q
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	Unknown	S
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	Unknown	S

```
# Creating a new feature of title from name column based on the pattern found above
titanic['Title'] = titanic['Name'].str.extract(r',\s(.*?)\.')
titanic['Title'] = titanic['Title'].replace('Ms', 'Miss')
titanic['Title'] = titanic['Title'].replace('Dona', 'Mrs')
titanic['Title'] = titanic['Title'].replace(['Col', 'Rev', 'Dr'], 'Rare')
```

```
# Creating another feature of Age group by making bins
bins = [-np.inf, 17, 32, 45, 50, np.inf]
labels = ["Children", "Young", "Mid-Aged", "Senior-Adult", 'Elderly']
titanic['Age_Group'] = pd.cut(titanic['Age'], bins = bins, labels = labels)
```

```
# Generting another new feature of family size
titanic['Family'] = titanic['SibSp'] + titanic['Parch']
```

```
# Dropping non essential coclumns
titanic.drop(['PassengerId', 'Name', 'Ticket'], axis = 1, inplace = True)
```

```
titanic.head()
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked	Title	Age_Group	Family
0	0	3	male	34.5	0	0	7.8292	Unknown	Q	Mr	Mid-Aged	0
1	1	3	female	47.0	1	0	7.0000	Unknown	S	Mrs	Senior-Adult	1
2	0	2	male	62.0	0	0	9.6875	Unknown	Q	Mr	Elderly	0
3	0	3	male	27.0	0	0	8.6625	Unknown	S	Mr	Young	0
4	1	3	female	22.0	1	1	12.2875	Unknown	S	Mrs	Young	2

```
# Chaning the positon of columns to place them right after their parent column
col_to_move = titanic.pop('Age_Group')
titanic.insert(4, 'Age_Group', col_to_move)
```

```
col_to_move = titanic.pop('Family')
titanic.insert(7, 'Family', col_to_move)
titanic['Age_Group'] = titanic['Age_Group'].astype('object')
```

Insights:

- Following of the 3 new features were created: **Title, Age_Group, and Family**
- Next, **positions** of these new columns were changed and their **data type** as well

Exploratory Data Analysis

Descriptive Analysis

```
titanic.describe()
```

	Survived	Pclass	Age	SibSp	Parch	Family	Fare
count	418.000000	418.000000	418.000000	418.000000	418.000000	418.000000	418.000000
mean	0.363636	2.265550	29.599282	0.447368	0.392344	0.839713	35.576535
std	0.481622	0.841838	12.703770	0.896760	0.981429	1.519072	55.850103
min	0.000000	1.000000	0.170000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	1.000000	23.000000	0.000000	0.000000	0.000000	7.895800
50%	0.000000	3.000000	27.000000	0.000000	0.000000	0.000000	14.454200
75%	1.000000	3.000000	35.750000	1.000000	0.000000	1.000000	31.471875
max	1.000000	3.000000	76.000000	8.000000	9.000000	10.000000	512.329200

```
titanic.describe(include = 'O')
```

	Sex	Age_Group	Cabin	Embarked	Title
count	418	418	418	418	418
unique	2	5	77	3	5
top	male	Young	Unknown	S	Mr
freq	266	257	327	270	240

```
titanic.groupby('Sex')[['Survived', 'Pclass', 'Age', 'SibSp', 'Parch', 'Family', 'Fare']].mean()
```

	Survived	Pclass	Age	SibSp	Parch	Family	Fare
Sex							
female	1.0	2.144737	29.734145	0.565789	0.598684	1.164474	49.747699
male	0.0	2.334586	29.522218	0.379699	0.274436	0.654135	27.478728

```
titanic.groupby('Embarked')[['Survived', 'Pclass', 'Age', 'SibSp', 'Parch', 'Family', 'Fare']].mean()
```

	Survived	Pclass	Age	SibSp	Parch	Family	Fare
Embarked							
C	0.392157	1.794118	33.220588	0.421569	0.382353	0.803922	66.259765
Q	0.521739	2.869565	28.108696	0.195652	0.021739	0.217391	10.957700
S	0.325926	2.340741	28.485185	0.500000	0.459259	0.959259	28.179413

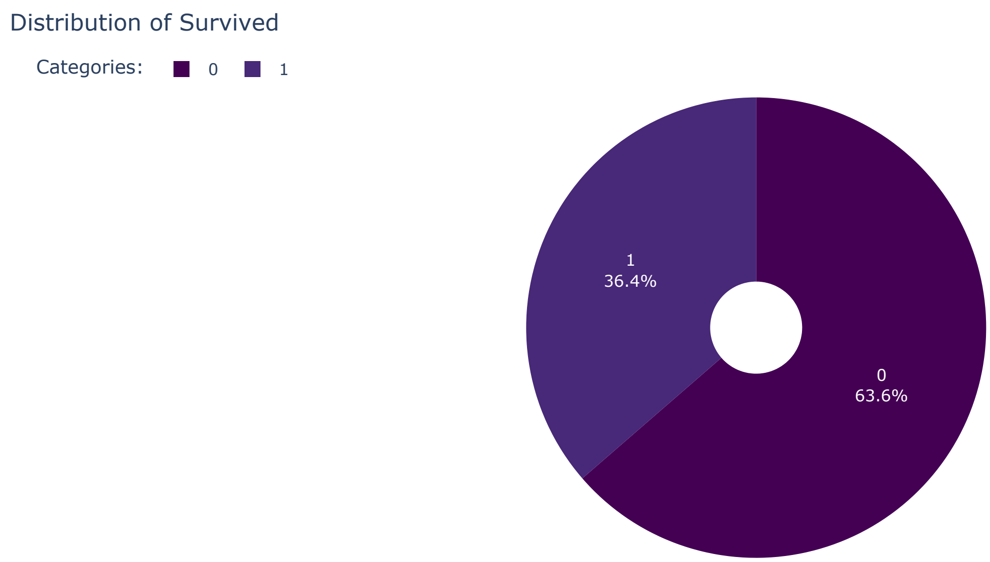
Insights:

- The analysis revealed that mostly people are: **Young male who have traveled more Southampton**

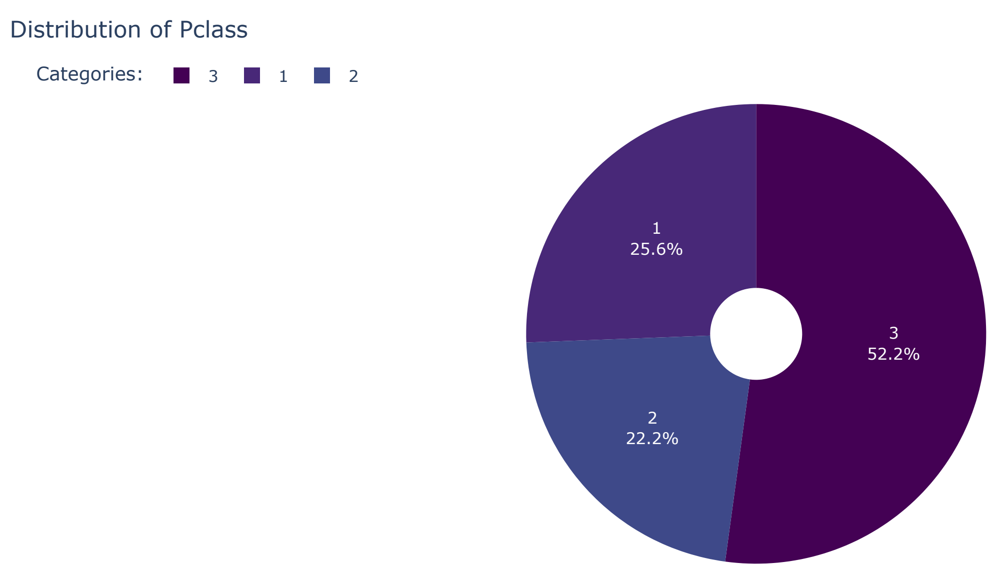
- **Females** are more likely to travel with someone and pay high fares
- Furthermore, people embarked from **Cherbourg** have an **average age of 33 and fares to pay around 66 pounds**

Univariate Analysis

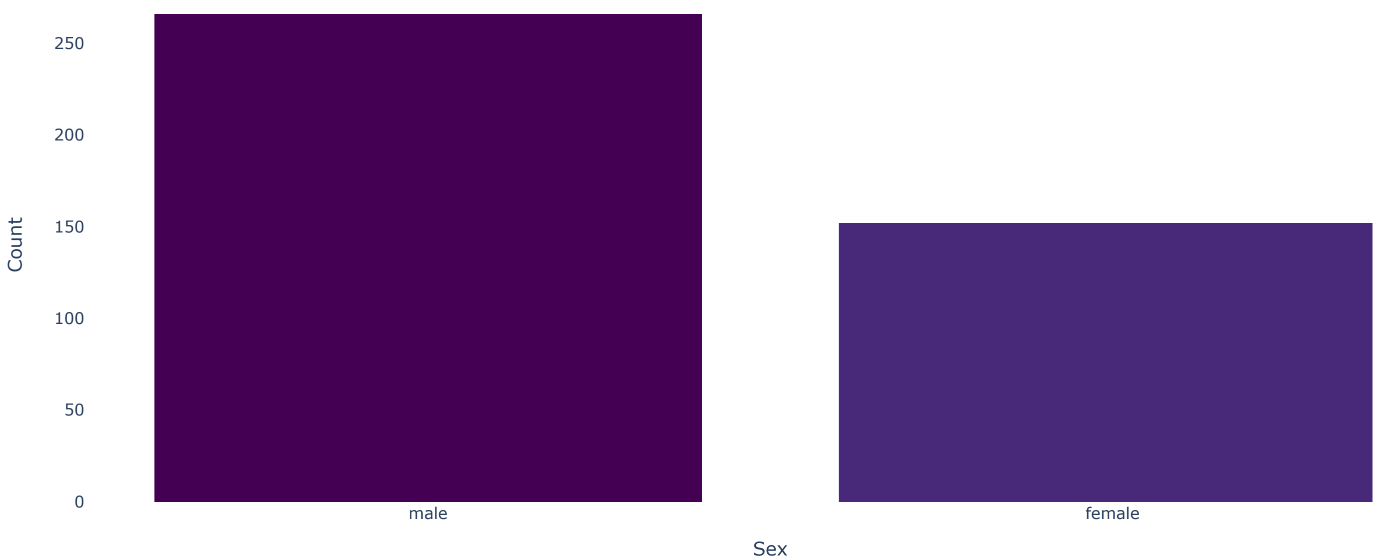
```
survived_counts = titanic['Survived'].value_counts()
fig_surv_perc = px.pie(titanic, names= survived_counts.index,  values = survived_counts.values, title=f'Distribution of Survived', hole=0.2,
fig_surv_perc.update_traces(textinfo='percent+label')
fig_surv_perc.update_layout(legend_title_text='Categories:', legend=dict(orientation="h", yanchor="bottom", y=1.02))
fig_surv_perc.show()
```

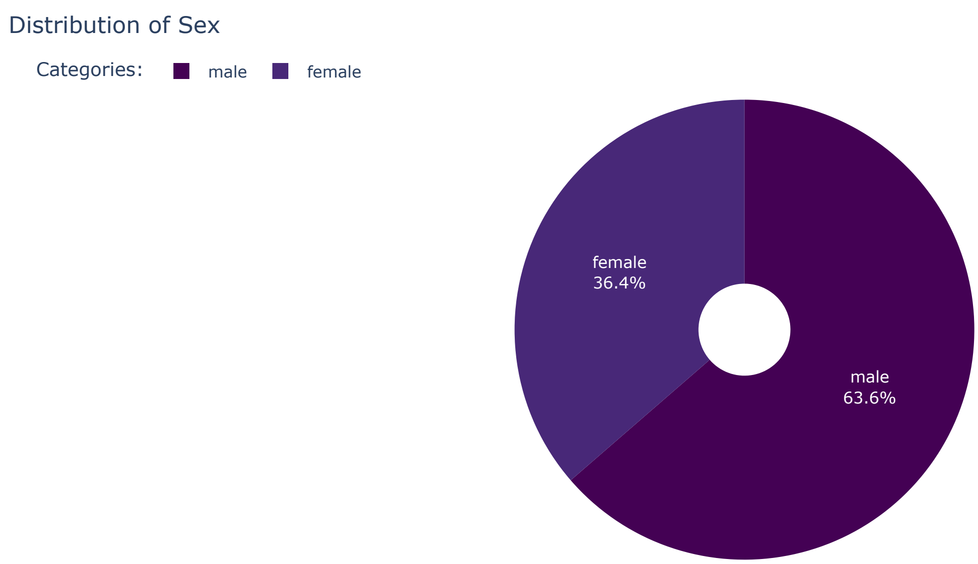



```
pclass_counts = titanic.Pclass.value_counts()
fig_pclass_perc = px.pie(titanic, names= pclass_counts.index, values = pclass_counts.values, title=f'Distribution of Pclass', hole=0.2, color
fig_pclass_perc.update_traces(textinfo='percent+label')
fig_pclass_perc.update_layout(legend_title_text='Categories:', legend=dict(orientation="h", yanchor="bottom", y=1.02))
fig_pclass_perc.show()
```

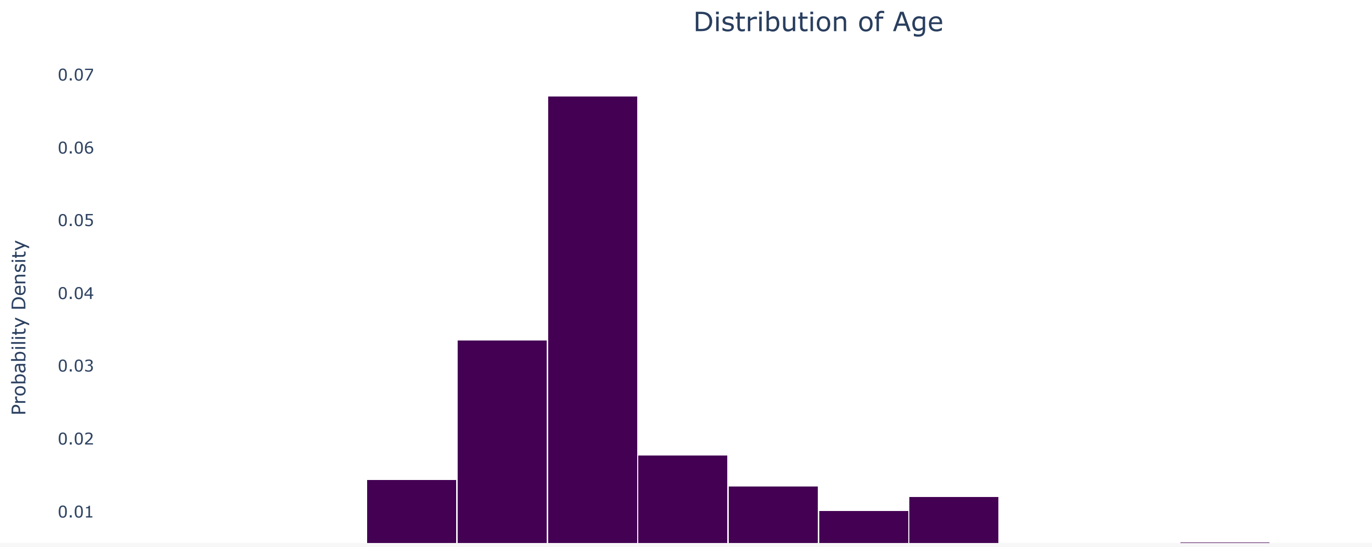


```
fig_sex_count = px.histogram(titanic, x = 'Sex', color = 'Sex', color_discrete_sequence=px.colors.sequential.Viridis)
fig_sex_count.update_layout(title_text='Count of different Sex', xaxis_title='Sex', yaxis_title='Count', plot_bgcolor = 'white')
fig_sex_count.show()
fig_sex_perc = px.pie(titanic, names= 'Sex', title=f'Distribution of Sex', hole=0.2, color_discrete_sequence=px.colors.sequential.Viridis)
fig_sex_perc.update_traces(textinfo='percent+label')
fig_sex_perc.update_layout(legend_title_text='Categories:', legend=dict(orientation="h", yanchor="bottom", y=1.02))
fig_sex_perc.show()
```

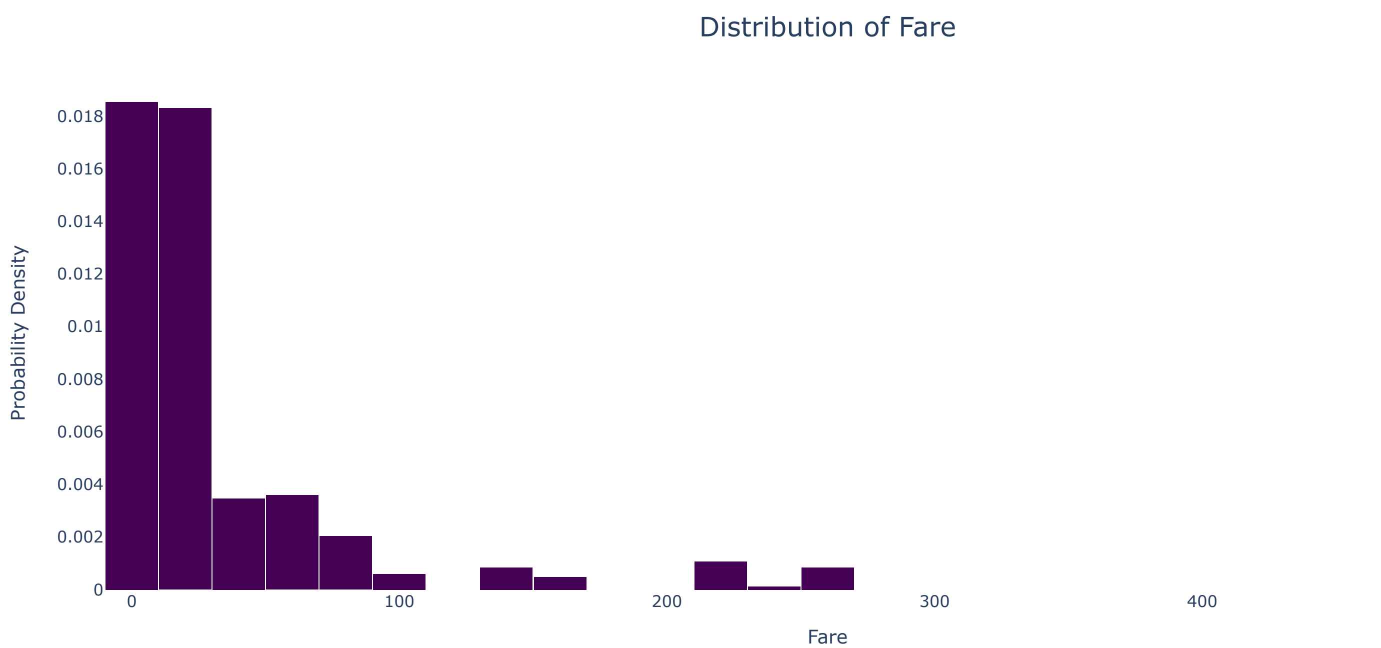





```
fig_age = px.histogram(titanic, x='Age', nbins=30, histnorm='probability density')
fig_age.update_traces(marker=dict(color='#440154'), selector=dict(type='histogram'))
fig_age.update_layout(title='Distribution of Age', title_x=0.5, title_pad=dict(t=20), title_font=dict(size=20), xaxis_title='Age', yaxis_titl
fig_age.show()
```

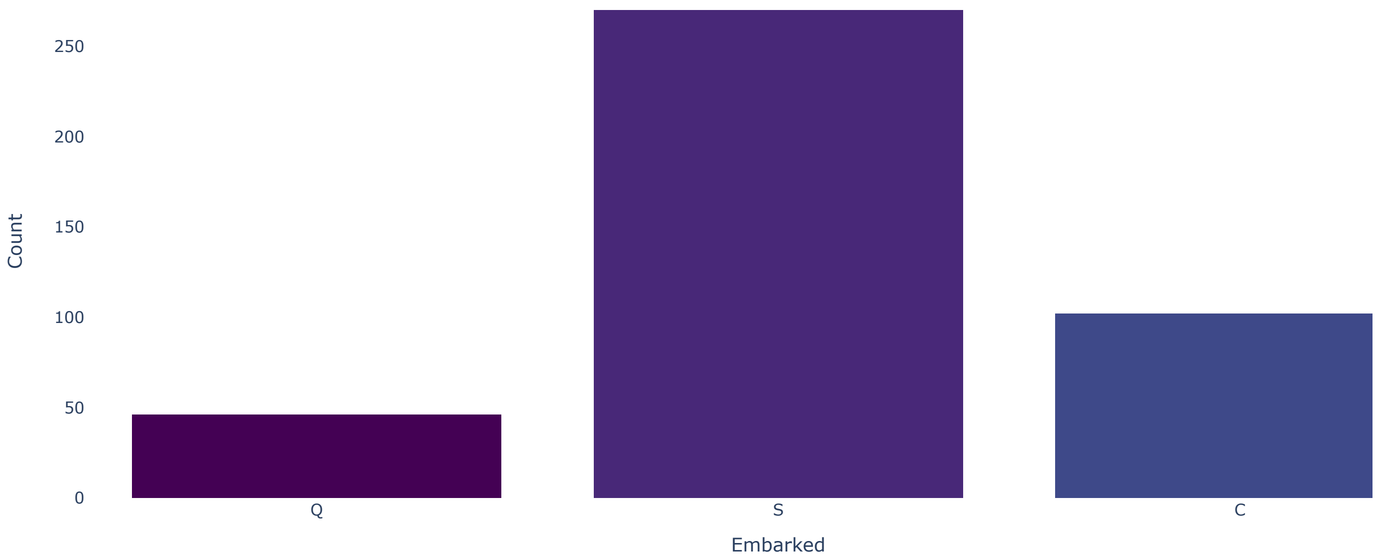


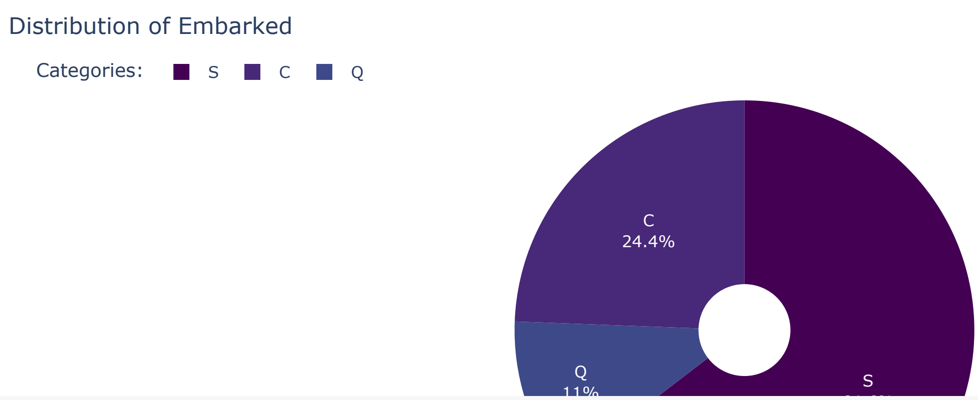

```
fig_fare = px.histogram(titanic, x='Fare', nbins=30, histnorm='probability density')
fig_fare.update_traces(marker=dict(color='#440154'), selector=dict(type='histogram'))
fig_fare.update_layout(title='Distribution of Fare', title_x=0.5, title_pad=dict(t=20), title_font=dict(size=20), xaxis_title='Fare', yaxis_t
fig_fare.show()
```



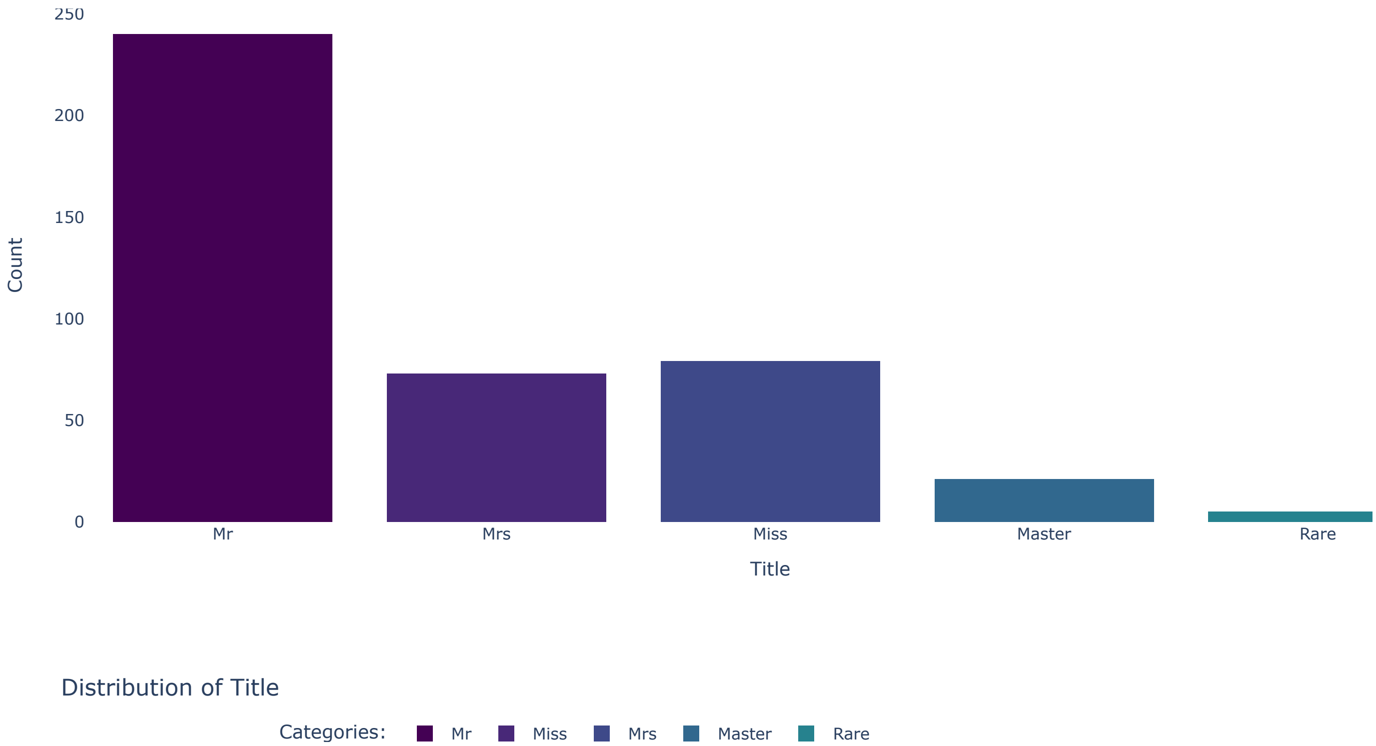

```
fig_embarked_count = px.histogram(titanic, x = 'Embarked', color = 'Embarked', color_discrete_sequence=px.colors.sequential.Viridis)
fig_embarked_count.update_layout(title_text='Count of different Embarked', xaxis_title='Embarked', yaxis_title='Count', plot_bgcolor = 'white
fig_embarked_count.show()

fig_embarked_perc = px.pie(titanic, names= 'Embarked', title=f'Distribution of Embarked', hole=0.2, color_discrete_sequence=px.colors.sequent
fig_embarked_perc.update_traces(textinfo='percent+label')
fig_embarked_perc.update_layout(legend_title_text='Categories:', legend=dict(orientation="h", yanchor="bottom", y=1.02))
fig_embarked_perc.show()
```






```
fig_title_count = px.histogram(titanic, x = 'Title', color = 'Title', color_discrete_sequence=px.colors.sequential.Viridis)
fig_title_count.update_layout(title_text='Count of different Title', xaxis_title='Title', yaxis_title='Count', plot_bgcolor = 'white')
fig_title_count.show()
fig_title_perc = px.pie(titanic, names= 'Title', title=f'Distribution of Title', hole=0.2, color_discrete_sequence=px.colors.sequential.Virid
fig_title_perc.update_traces(textinfo='percent+label')
fig_title_perc.update_layout(legend_title_text='Categories:', legend=dict(orientation="h", yanchor="bottom", y=1.02))
fig_title_perc.show()
```



Insights:

- Only **36.4% of the people** survived the crash
- The dataset also have a high distribution of poeple from **Pclass = 3**, and **high ratio of males**
- The distribution of age is centered around **25-29**, and fare is around **10-30**
- Most of the people are **embarked from Southampton**, and mostly the title holded by passengers are **Mr. = Single Male**

Bivariate Analysis

```
fig_pclass_surv = px.histogram(titanic, x = 'Pclass', barmode = 'group', color = 'Survived', color_discrete_sequence=px.colors.sequential.Vir
fig_pclass_surv.update_layout(title = 'Survival according to passenger classes', plot_bgcolor = 'white')
fig_pclass_surv.show()
```

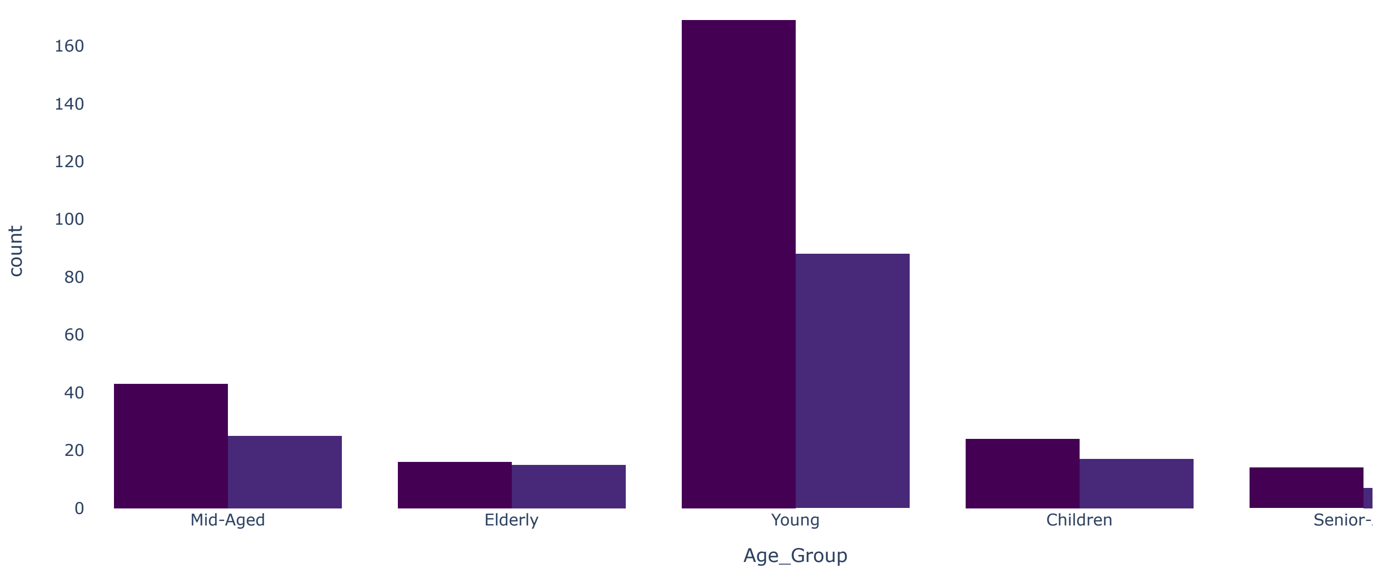
Survival according to passenger classes

```
fig_pclass_surv = px.histogram(titanic, x = 'Sex', barmode = 'group', color = 'Survived', color_discrete_sequence=px.colors.sequential.Viridi
fig_pclass_surv.update_layout(title = 'Survival according to gender', plot_bgcolor = 'white')
fig_pclass_surv.show()
```

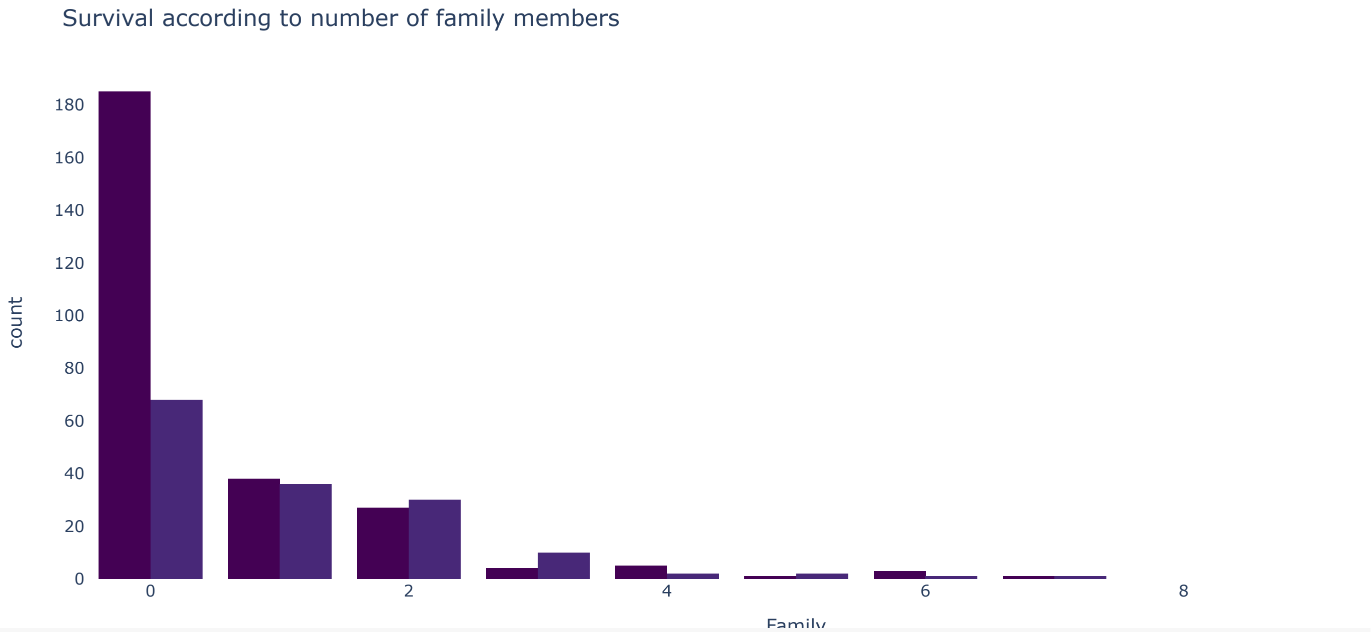
Survival according to gender


```
fig_embarked_surv = px.histogram(titanic, x = 'Age_Group', barmode = 'group', color = 'Survived', color_discrete_sequence=px.colors.sequentia
fig_embarked_surv.update_layout(title = 'Survival according to age groups', plot_bgcolor = 'white')
fig_embarked_surv.show()
```

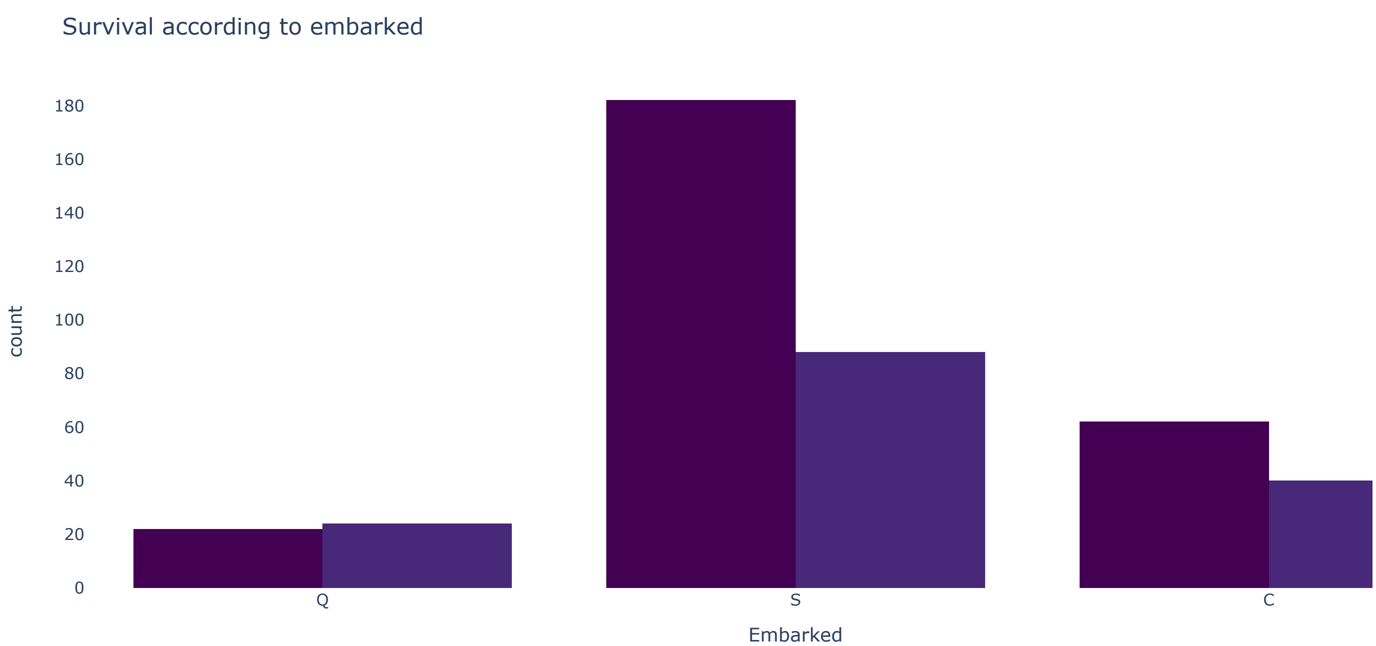
Survival according to age groups




```
fig_family_surv = px.histogram(titanic, x = 'Family', barmode = 'group', color = 'Survived', color_discrete_sequence=px.colors.sequential.Vir
fig_family_surv.update_layout(title = 'Survival according to number of family members', plot_bgcolor = 'white')
fig_family_surv.show()
```




```
fig_embarked_surv = px.histogram(titanic, x = 'Embarked', barmode = 'group', color = 'Survived', color_discrete_sequence=px.colors.sequential
fig_embarked_surv.update_layout(title = 'Survival according to embarked', plot_bgcolor = 'white')
fig_embarked_surv.show()
```

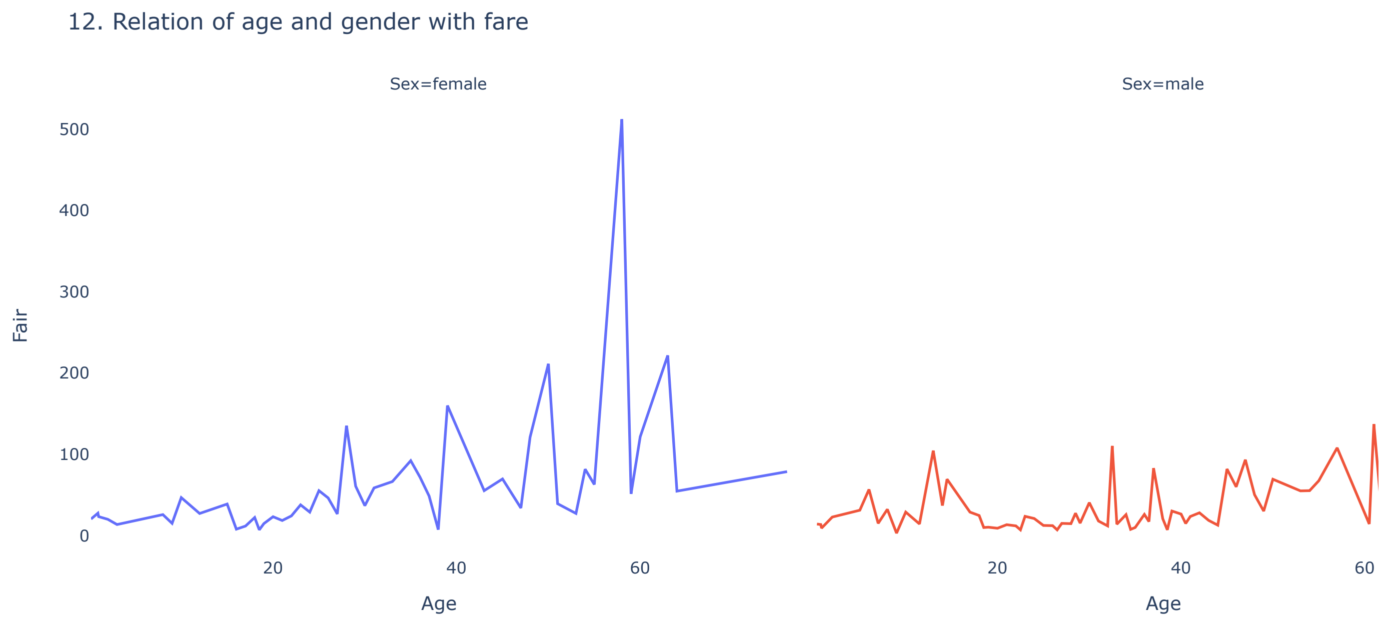


Insights:

- The least deaths are from **Pclass = 1** and the highest number of deaths are from **Pclass = 3**
- The dataset also have a high distribution of poeple from **Pclass = 3**, and **high ratio of males**
- **None of the male survived**, and all the **females survived**
- The highest death count is from **Young Age Group**, and **Elderly People** have a good survival count
- Poeple with **few family members** are more likely to survive according to analysis
- A high ratio of poeple who **embarked from Queenstown survived**, and **Southampton** has the highest death casualities

Multivariate Analysis

```
grouped_data = titanic.groupby(['Age', 'Sex', 'Survived']).agg({'Fare': 'mean'}).reset_index()
fig = px.line(grouped_data, x='Age', y='Fare', color='Survived', facet_col='Sex', facet_col_wrap=2, labels={'Fare': 'Fare', 'Survived': 'Surv
fig.update_layout(hovermode='x unified', plot_bgcolor = 'white')
fig.update_xaxes(title_text='Age')
fig.update_yaxes(title_text='Fair', row=1, col=1)
fig.show()
```



Insights:

- The analysis revealed that **Fare is a bit high for females compared to males**, and the **Fare is likely to increase according to Age for females**. Overall, **Age doesn't have a significant impact on survival**

Data Preprocessing

1. Label Encoding

```
# Labeling the ordinal variables
le = LabelEncoder()
cols = ['Sex', 'Age_Group', 'Cabin', 'Embarked', 'Title']
for col in cols:
    titanic[col] = le.fit_transform(titanic[col])
```

2. Class Imbalance

```
# Checking the class count for target variable
titanic.Survived.value_counts()
```

```
0    266
1    152
Name: Survived, dtype: int64
```

```
X = titanic.drop('Survived', axis = 1)
y = titanic['Survived']
```

```
# Using the SMOTE technique to handle class imbalance
smote = SMOTE(random_state = 42)
X_balanced, y_balanced = smote.fit_resample(X, y)
```

3. Splitting into training and testing

```
# Splitting the dataset into training and testing parts
X_train, X_test, y_train, y_test = train_test_split(X_balanced, y_balanced, test_size = 0.3, random_state = 42)
```

4. Feature Scaling

```
# Doing feature scaling by StandardScaler
sc = StandardScaler()
X_train_scaled = sc.fit_transform(X_train)
X_test_scaled = sc.transform(X_test)
```

Model Building

```
# Building the models
lr = LogisticRegression()
rf = RandomForestClassifier()
gbc = GradientBoostingClassifier()
lr.fit(X_train_scaled, y_train)
rf.fit(X_train_scaled, y_train)
gbc.fit(X_train_scaled, y_train)
lr_pred = lr.predict(X_test_scaled)
rf_pred = rf.predict(X_test_scaled)
gbc_pred = gbc.predict(X_test_scaled)
```

Model Evaluation

```
# Evaluating the models by generating classification report and cross validation scores

lr_report = classification_report(y_test, lr_pred)
lr_scores = cross_val_score(lr, X_train_scaled, y_train, cv=5, scoring='accuracy')

rf_report = classification_report(y_test, rf_pred)
rf_scores = cross_val_score(rf, X_train_scaled, y_train, cv=5, scoring='accuracy')

gbc_report = classification_report(y_test, gbc_pred)
gbc_scores = cross_val_score(gbc, X_train_scaled, y_train, cv=5, scoring='accuracy')


print('The classification report of Logistic Regression is below : ', '\n\n\n', lr_report)
print(f"Logistic Regression Mean Cross-Validation Score: {lr_scores}")

print('\n', '='*100, '\n')
print('The classification report of Random Forest is below : ', '\n\n\n', rf_report)
print(f"Random Forest Mean Cross-Validation Score: {rf_scores}")

print('\n', '='*100, '\n')
print('The classification report of Gradient Bossting Classifier is below : ', '\n\n\n', rf_report)
print(f"Gradient Boosting Classifier Mean Cross-Validation Score: {gbc_scores}")
```

```
The classification report of Logistic Regression is below :


              precision    recall  f1-score   support

           0       1.00      1.00      1.00        78
           1       1.00      1.00      1.00        82

    accuracy                           1.00       160
   macro avg       1.00      1.00      1.00       160
weighted avg       1.00      1.00      1.00       160

Logistic Regression Mean Cross-Validation Score: [1. 1. 1. 1. 1.]

 ====================================================================================================

The classification report of Random Forest is below :
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        78
           1       1.00      1.00      1.00        82

    accuracy                           1.00       160
   macro avg       1.00      1.00      1.00       160
weighted avg       1.00      1.00      1.00       160

Random Forest Mean Cross-Validation Score: [1. 1. 1. 1. 1.]

 ====================================================================================================

The classification report of Gradient Bossting Classifier is below :


              precision    recall  f1-score   support

           0       1.00      1.00      1.00        78
           1       1.00      1.00      1.00        82

    accuracy                           1.00       160
   macro avg       1.00      1.00      1.00       160
weighted avg       1.00      1.00      1.00       160

Gradient Boosting Classifier Mean Cross-Validation Score: [1. 1. 1. 1. 1.]
```

Conclusion:

In this Titanic Survival Prediction analysis, we have explored various aspects of the dataset to understand the factors influencing survival. We found that only 36.4% of the passengers survived the crash, with significant differences in survival rates among different passenger classes, genders, and age groups. The dataset also revealed that certain features, such as Fare and embarkation location, played a role in survival. We trained several classification models to predict survival, all of which performed well, likely due to the relatively small dataset size.

Insights:

Our analysis unveiled key insights into the Titanic dataset. We addressed missing values by filling null entries in the Age and Fare columns with medians due to the presence of outliers, while the Cabin column was filled with "Unknown." New features, including Title, Age_Group, and Family, were created to enhance our understanding of passenger demographics. We discovered that young males traveling from Southampton constituted the majority, and females were more likely to travel with others and pay higher fares. Notably, passengers from Cherbourg had an average age of 33 and paid around 66 pounds in fares. Furthermore, we observed that Pclass 3 had the highest number of deaths, with no surviving males and all females surviving. Family size appeared to influence survival, and passengers from Queenstown had a higher survival rate compared to those from Southampton.

What's next?

For future analysis, it would be beneficial to explore more advanced machine learning techniques and consider feature engineering to improve model performance further. Additionally, investigating the impact of other variables not included in this analysis, such as cabin location and passenger demographics beyond age, gender, and family size, could provide deeper insights. Further exploration of the dataset and refining models could enhance our ability to predict Titanic passenger survival more accurately.
