Titanic Machine Learning from Disaster.

Introduction.

The Titanic disaster of 1912 claimed the lives of more than 1,500 passengers and crew, making it one of the deadliest maritime tragedies in history. Historical accounts suggest that survival chances were not random. Factors such as gender, age, and social class played a major role, with the “women and children first” policy influencing who made it into the lifeboats.

This project uses the Titanic dataset from Kaggle to explore these patterns and build a machine learning model that can predict whether a passenger survived based on their personal and travel details. By analyzing demographic information, ticket class, fares paid, and family relationships aboard the ship, the goal is to uncover which factors most strongly influenced survival and to create a model that can make accurate predictions. The results provide insight into the historical event and show how data science can be applied to understand real-world outcomes.

Problem Statement.

The objective of this project is to use the Titanic passenger dataset to develop a predictive model that can determine the likelihood of survival for each passenger based on available information. The aim is to combine historical understanding with data-driven analysis by exploring patterns in the data, identifying the most influential factors affecting survival, and applying machine learning techniques to make accurate predictions.

Through this process, the project seeks to demonstrate the complete data science workflow, including data exploration, cleaning, feature engineering, model training, and evaluation. The final goal is not only to create a reliable survival prediction model but also to gain meaningful insights into the factors that shaped the survival outcomes of Titanic passengers.

Scope of the Analysis.

This project focuses on analyzing the Titanic passenger dataset to build and evaluate a predictive model for survival. It aims to explore patterns in the data, clean and prepare the dataset, create new features, and apply machine learning models to identify the most important factors affecting survival. The scope is limited to the variables provided in the dataset and does not include any external data sources.

Dataset Description.

Overview of the Data.

The dataset used in this analysis is the Titanic passenger dataset from Kaggle. It contains historical records of passengers aboard the RMS Titanic, including demographic details, ticket and cabin information, travel class, and survival status. The training dataset has 891 passenger records with both feature values and survival labels, while the test dataset has 418 records with feature values only, intended for prediction.

Data Characteristics.

The dataset includes both numerical and categorical variables. Numerical variables include Age, Fare, SibSp (number of siblings/spouses aboard), and Parch (number of parents/children aboard). Categorical variables include Sex, Pclass (passenger class), and Embarked (port of embarkation). Some features contain missing values, particularly Age, Cabin, and Embarked, requiring data cleaning and imputation. Additional engineered features such as FamilySize, IsAlone, and Has Embarked were created to enhance model performance.

Data Preprocessing.

Before modeling, the dataset was prepared through several preprocessing steps to ensure data quality and consistency.

Structure of the Workflow:

I started my data analysis by loading my csv files into my model using the pandas library loading the gender_submisson.csv, test.csv and the tran.csv. This allowed me to have the training data, the test data for predictions, and a sample submission file for data analysis :

```
import pandas as pd

gender_submission_df = pd.read_csv("gender_submission.csv")
test_df = pd.read_csv("test.csv")
train_df = pd.read_csv("train.csv")
```

After loading the datasets, I performed an initial inspection using functions such as .head() to view the first few rows, .info() to check data types and non-null counts, and .describe() to see basic statistical summaries of the numerical features :

```
print(train_df.head())
```

```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                Name     Sex   Age  SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0      1
2                             Heikkinen, Miss. Laina  female  26.0      0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                           Allen, Mr. William Henry    male  35.0      0

   Parch            Ticket     Fare Cabin Embarked
0      0         A/5 21171   7.2500   NaN        S
1      0          PC 17599  71.2833   C85        C
2      0  STON/O2. 3101282   7.9250   NaN        S
3      0            113803  53.1000  C123        S
4      0            373450   8.0500   NaN        S
```

```
print(train_df.shape)
```

```
(891, 12)
```

```
print(train_df.columns)
```

```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

```
print(train_df.info)
```

```
<bound method DataFrame.info of      PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
2              3         1       3
3              4         1       1
4              5         0       3
..           ...       ...     ...
886          887         0       2
887          888         1       1
888          889         0       3
889          890         1       1
890          891         0       3
```

```
print(train_df.describe)
```

```
<bound method NDFrame.describe of      PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
2              3         1       3
3              4         1       1
4              5         0       3
..           ...       ...     ...
886          887         0       2
887          888         1       1
888          889         0       3
889          890         1       1
890          891         0       3
```

The next step was to explore the dataset for missing values using .isnull().sum(). This revealed that the Age, Cabin, and Embarked columns had missing entries. Based on this finding, I applied specific cleaning strategies: filling missing Embarked values with the most common category, imputing Age with the median based on passenger sex and class, filling missing Fare values in the test set with the median, and dropping the Cabin column due to excessive missing data :

```
print(train_df.isnull().sum())
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

```
print(train_df['Sex'].value_counts())
print(train_df['Pclass'].value_counts())
```

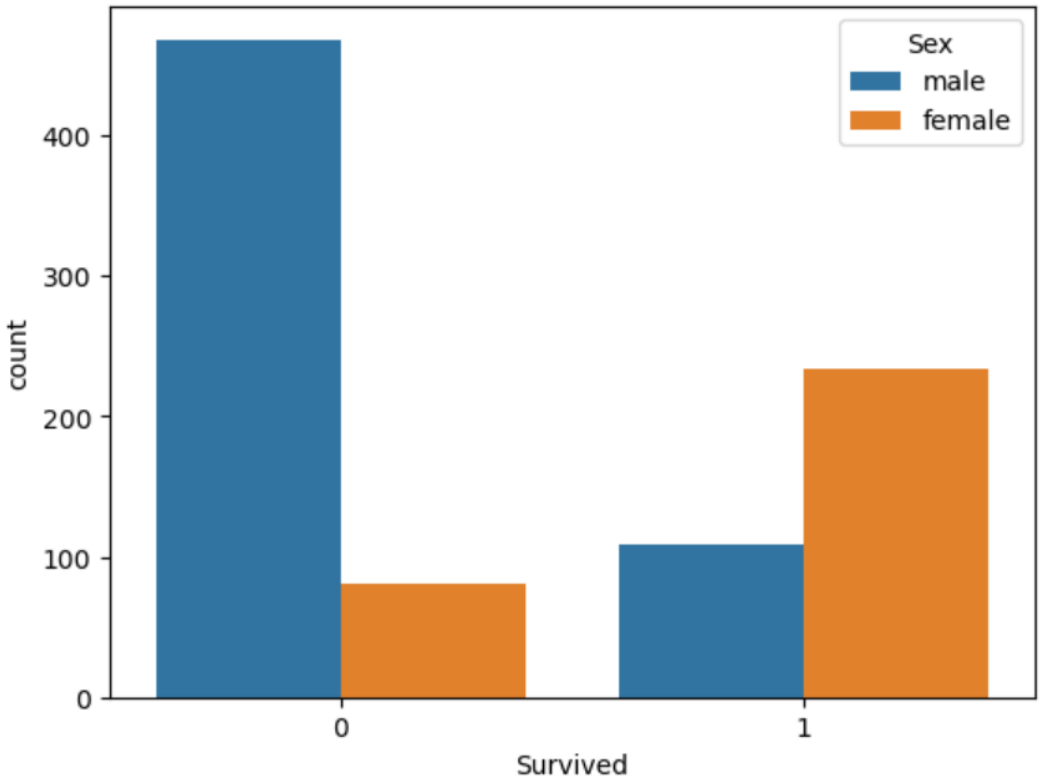
```
Sex
male      577
female    314
Name: count, dtype: int64
```

Once the data was cleaned, I proceeded with feature engineering to enhance the dataset. I created FamilySize by adding SibSp and Parch and including the passenger, IsAlone to indicate passengers traveling without family, and Has Embarked to flag whether cabin data was present.

```
import seaborn as sns
import matplotlib.pyplot as plt

sns.countplot(x='Survived', hue='Sex', data=train_df)
plt.show()
```

In this I started approaches of statistical ploting of graphs for easier visualization of data :



The graph below illustrates the relationship between sex and survival in the dataset. It shows that survival rates vary notably between males and females, suggesting that gender may have played an important role in determining the likelihood of survival. This insight helps us understand demographic patterns in the data.

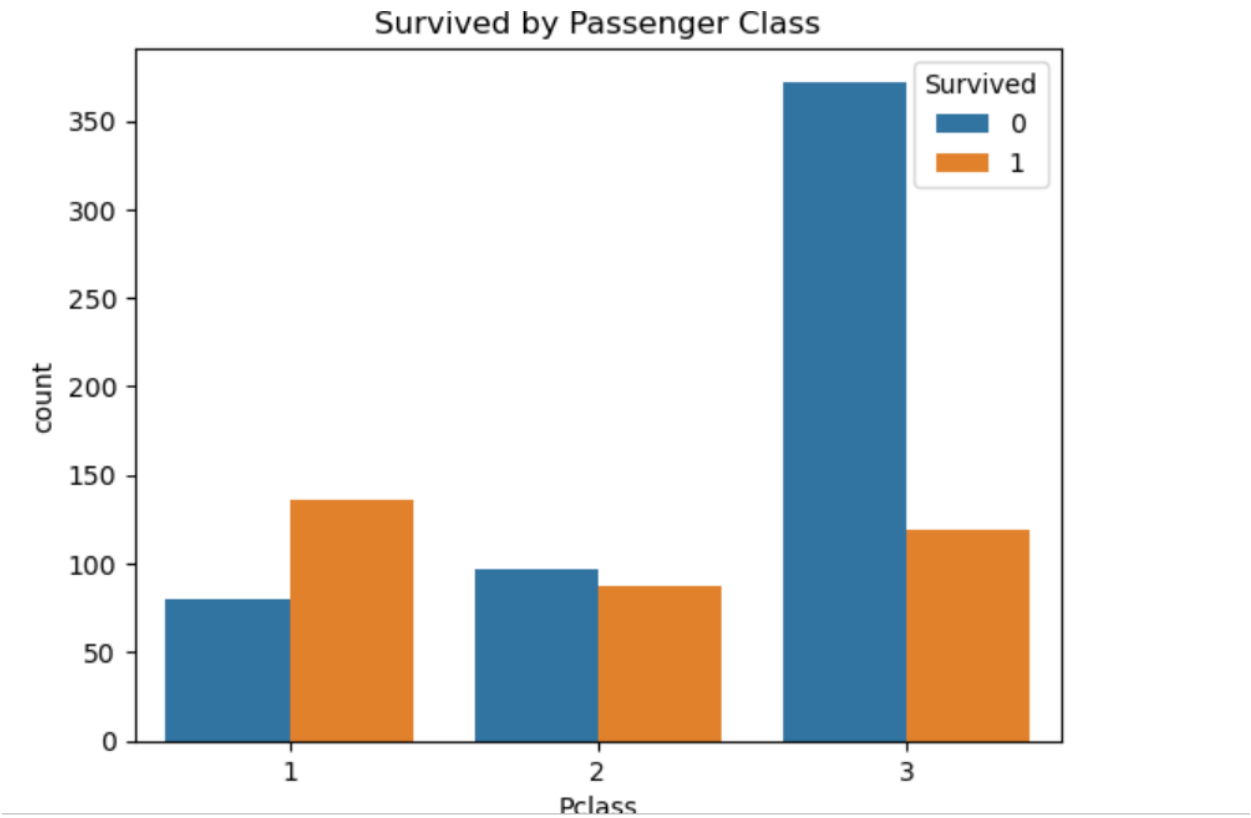
```
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt


train_df = pd.read_csv("train.csv")

train_df['Survived'] = train_df['Survived'].astype(str)

sns.countplot(x='Pclass', hue='Survived', data=train_df)
plt.title('Survived by Passenger Class')
plt.show()
```

The graph illustrates the relationship between passenger class (Pclass) and survival. It reveals that survival rates were higher in the upper classes and declined as class level decreased, indicating a possible link between socio-economic status and chances of survival:



In addition to analyzing sex and passenger class, I conducted further exploratory analyses on other variables and built predictive models to better understand the factors affecting survival. These additional steps helped validate the insights and improve the overall accuracy of the findings.

Exploratory Data Analysis (EDA).

In this section, I performed a detailed exploratory analysis to uncover key patterns and relationships in the data. Some of the main insights include:

- Sex and Survival: Females had a notably higher survival rate compared to males, indicating gender was an important factor.
- Passenger Class (Pclass) and Survival: Passengers in higher classes had better survival chances, suggesting socio-economic status impacted outcomes.
- Other Variables: Additional analysis was conducted on age, fare, and embarked location, among others, to better understand their influence on survival.

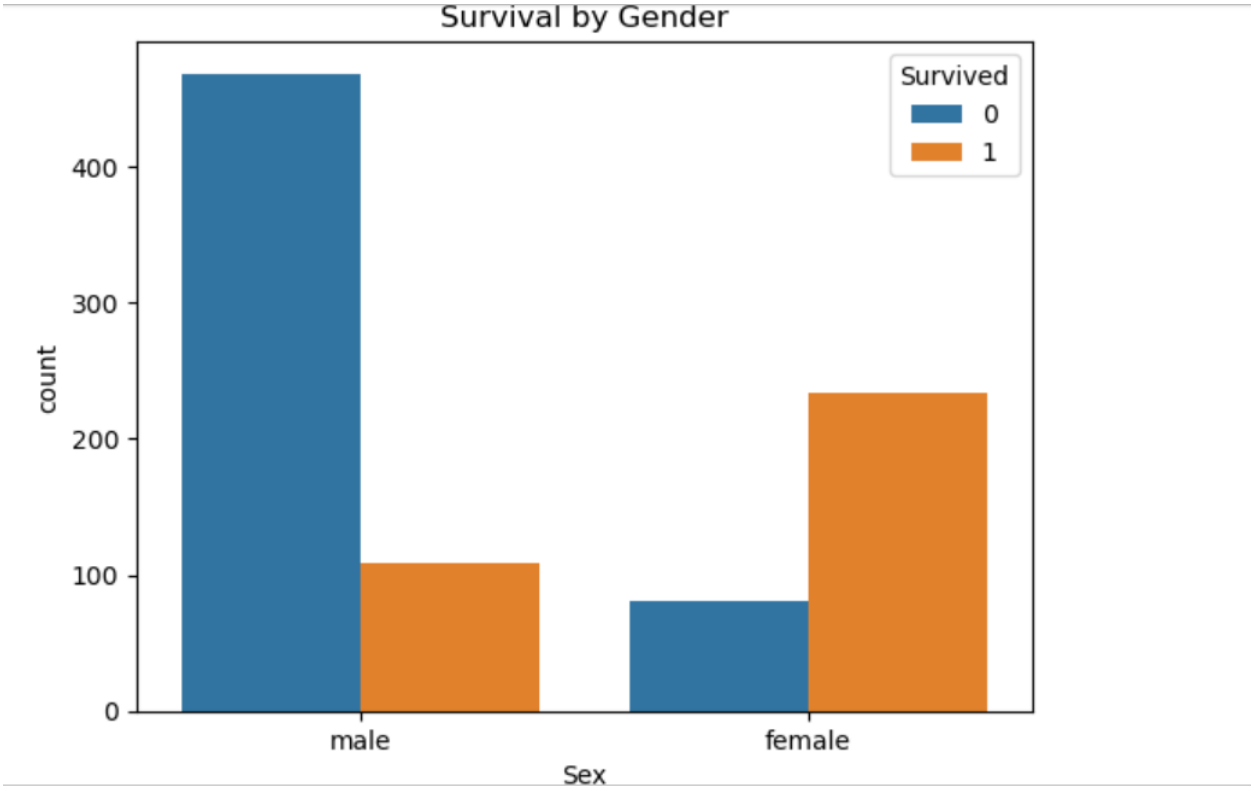
These insights helped guide the development of predictive models and deeper analysis.

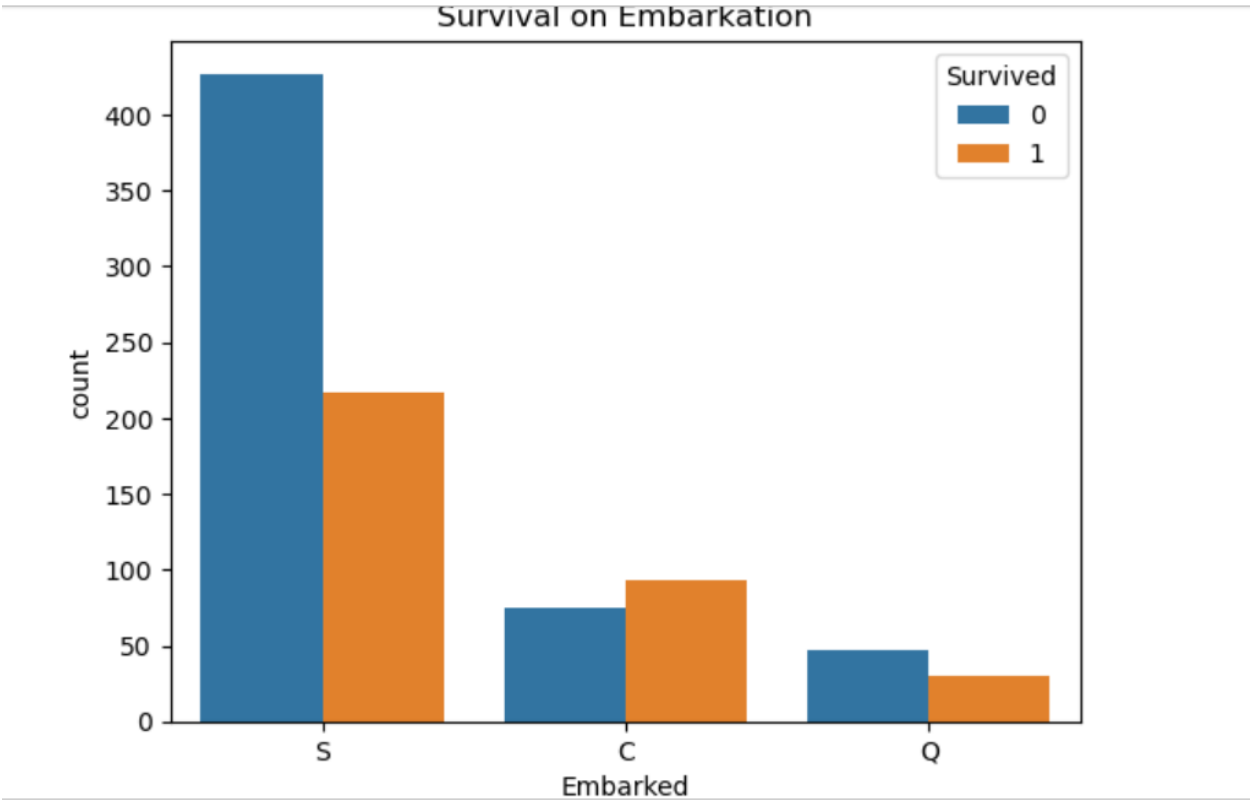
Feature Engineering.

- ✓ Family Size: Created by combining the number of siblings/spouses (SibSp) and parents/children (Parch) aboard to capture the effect of traveling with family on survival chances. Larger families might have different survival patterns compared to individuals.
- ✓ Is Alone: A binary feature derived from Family Size, indicating whether a passenger was traveling alone. This helps assess if traveling solo influenced survival outcomes.
- ✓ Title Extraction: Extracted titles (Mr., Mrs., Miss, etc.) from passenger names to capture social status or age group differences that might affect survival.
- ✓ Age Groups: Categorized continuous age values into groups (e.g., child, adult, senior) to better handle non-linear relationships with survival.

Each new feature was created to enhance model performance by incorporating meaningful information not directly available from the original variables.

- Categorical variables such as Sex and Embarked were encoded into numerical values to be compatible with machine learning algorithms. I then split the cleaned dataset into training and validation sets to allow for model evaluation by use of graphs and other various techniques :





Model Training.

I first split the dataset into training and validation sets to properly train and evaluate the models. After training the initial model, the validation accuracy obtained was 0.8156424581005587 (approximately 81.56%). Building on this, I proceeded to apply more advanced models for better prediction.

```
from sklearn.model_selection import train_test_split

x_train, x_val, y_train, y_val = train_test_split(
    x, y, test_size=0.2, random_state=42
)

print('Training Set Shape', x_train.shape)
print('Validation Set Shape', x_val.shape)
```

```
Training Set Shape (712, 11)
Validation Set Shape (179, 11)
```

For modeling, I started with Logistic Regression because it is easy to understand and works well for predicting two possible outcomes. After that, I used a Random Forest Classifier, which is a more advanced model that combines many decision trees to make better predictions. I checked how well both models performed by looking at their accuracy on the validation data, and both models scored about 81.56% accuracy.

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score


log_reg_model = LogisticRegression (max_iter=200)

log_reg_model.fit(x_train, y_train)

y_pred = log_reg_model.predict (x_val)

accuracy = accuracy_score(y_val, y_pred)
print("Validation Accuracy:", accuracy)
```

```
Validation Accuracy: 0.8156424581005587
```

- The Model gave us a validation accuracy of 0.8156424581005587 which was actually accurate.

Finally, I looked at the Random Forest model to see which features were most important in predicting survival. The results showed that Sex, Age, and Fare were the top factors that influenced whether a passenger survived or not.

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

rf_model = RandomForestClassifier(n_estimators=100, random_state=42)

rf_model.fit(x_train, y_train)

y_pred_rf = rf_model.predict(x_val)

accuracy_rf = accuracy_score(y_val, y_pred_rf)
print("Random Forest Validation Accuracy: ", accuracy)
```

```
Random Forest Validation Accuracy:  0.8156424581005587
```

- I confirmed the previous Logistics regression if my model was actually correct in the training, prediction and validation stage and the model predicted accurately with the same validation accuracy of 0.8156424581005587.

Later, I performed a feature importance analysis using the Random Forest model to better understand which variables had the strongest impact on predicting survival. Random Forest works by building many decision trees and combining their results, which makes it very effective at capturing complex relationships in the data. One of its advantages is that it provides a measure of how much each feature contributes to the model's decisions.

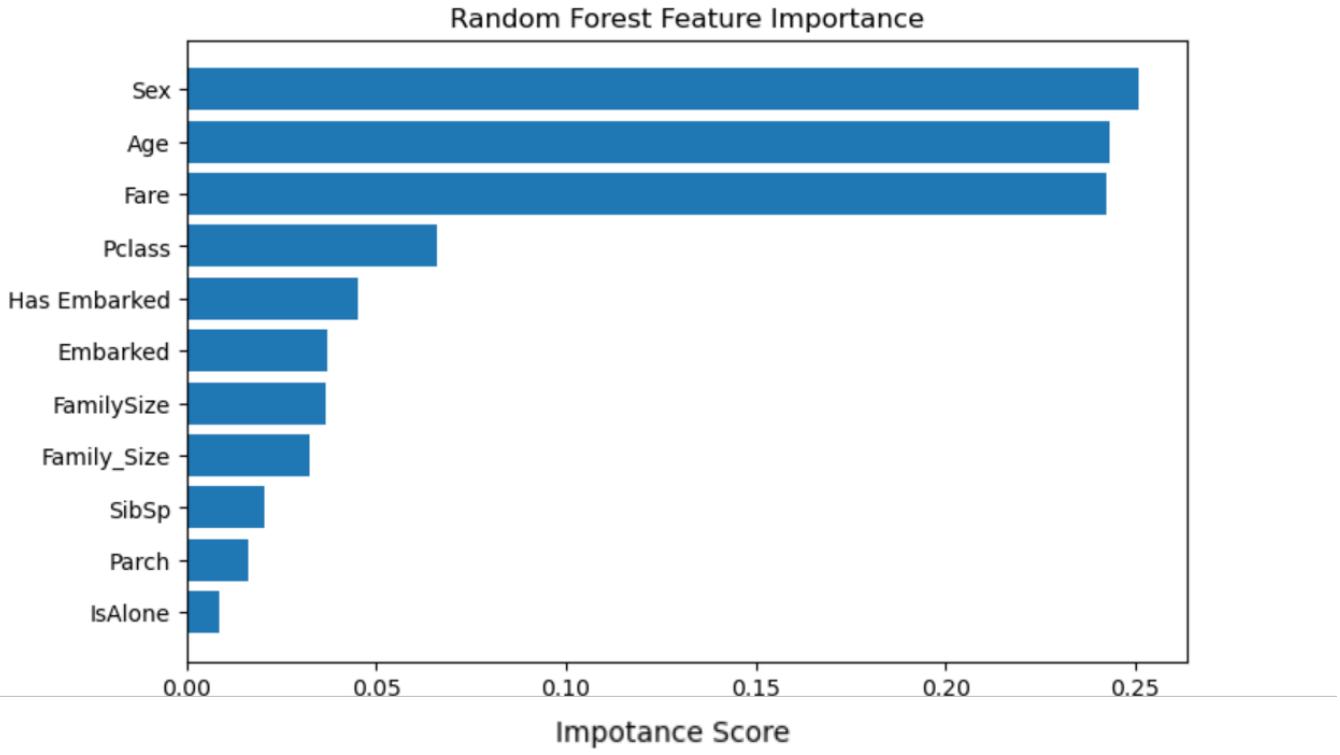
```
importances = rf_model.feature_importances_

feature_names = x_train.columns
feature_impotance_df = pd.DataFrame({
    'Feature' : feature_names,
    'Importance' : importances
}).sort_values(by='Importance', ascending = False)

print(feature_impotance_df)

plt.figure(figsize=(8,5))
plt.barh(feature_importance_df['Feature'],feature_importance_df['Importance'])
plt.gca().invert_yaxis()
plt.xlabel('Impotance Score')
plt.title('Random Forest Feature Importance')
plt.show()
```

```
          Feature  Importance
1             Sex    0.250914
2             Age    0.243240
5            Fare    0.242501
0          Pclass    0.065950
7   Has Embarked     0.045317
6       Embarked    0.037286
9     FamilySize    0.036737
8    Family_Size    0.032417
3           SibSp    0.020501
4           Parch    0.016388
10       IsAlone    0.008751
```



To carry out the feature importance analysis, I extracted the importance scores assigned by the Random Forest to each feature after training. These scores indicate how much each feature helped reduce uncertainty or "impurity" when making splits in the decision trees. The higher the score, the more important the feature is for predicting survival.

From the analysis, I found that Sex, Age, and Fare were the top three features influencing survival predictions. This means that the model relied heavily on these factors to differentiate between passengers who survived and those who did not. For example, gender differences in survival chances and the effect of passenger age on vulnerability were clearly reflected in the importance rankings. The Fare paid also indicated socio-economic status, which impacted survival odds.

This analysis helped validate previous findings from the exploratory data analysis and provided insight into the predictive power of each feature. It also guided further model refinement by focusing on the most relevant variables.

Results and Evaluation

Model Performance:

Both the Logistic Regression and Random Forest models achieved a validation accuracy of approximately 81.56%, indicating that they performed similarly well in predicting passenger survival. This accuracy shows that the models correctly classified survival status in about 8 out of 10 cases.

Comparison:

While Logistic Regression is simpler and easier to interpret, the Random Forest model can capture more complex patterns because it combines many decision trees. Despite their similar accuracy scores, the Random Forest offers additional advantages, such as identifying the relative importance of each feature in making predictions.

Interpretation:

The feature importance analysis from the Random Forest model highlighted **Sex**, **Age**, and **Fare** as the most influential factors affecting survival. This aligns with the exploratory data analysis, which showed that females, younger passengers, and those who paid higher fares had better chances of survival. These insights confirm the models' ability to capture meaningful relationships within the data.

Conclusion

Summary:

This project thoroughly analyzed factors affecting passenger survival through data exploration and predictive modeling. By preparing the data and creating meaningful features, the models were trained to identify patterns related to survival. Both Logistic Regression and Random Forest classifiers achieved a strong validation accuracy of about 81.56%, demonstrating their effectiveness in solving the survival prediction problem. The Random Forest's feature importance analysis confirmed that Sex, Age, and Fare were the most critical variables influencing survival outcomes.

Future Work:

To build on these results, future efforts could focus on hyperparameter tuning, exploring more advanced algorithms, and including additional data sources. Investigating variable interactions and performing more robust validation methods would strengthen model reliability. Ultimately, deploying the model for practical use could assist in decision-making processes related to passenger safety.

References

Kaggle Titanic Dataset. Available at: https://www.kaggle.com/c/titanic/data

Kamau Johnson. Documentation and analysis blog post on Medium. Available at: https://medium.com/@Kamau_Johnson