Data Analysis and Mining Report: Titanic Dataset

## Introduction to the Titanic Dataset

For my data analysis project, I worked with the Titanic Dataset, which is a popular dataset often used for machine learning and data mining exercises. The dataset contains information about passengers aboard the RMS Titanic, including whether they survived or not. The main goal of this analysis was to explore the data and build a model to predict passenger survival based on certain attributes.

The dataset consists of 891 rows (passengers) and 12 columns (features). Some of the key features in the dataset include:

**PassengerId**: A unique identifier for each passenger.

**Survived**: Whether or not the passenger survived (1 = survived, 0 = did not survive).

**Pclass**: The class of the passenger (1st, 2nd, or 3rd).

**Name**: The passenger's full name.

**Sex**: The gender of the passenger (male or female).

**Age**: The age of the passenger (many values are missing).

**SibSp**: The number of siblings or spouses aboard.

**Parch**: The number of parents or children aboard.

**Ticket**: The ticket number.

**Fare**: The amount paid for the ticket.

**Cabin**: The cabin number assigned to the passenger (many missing values).

**Embarked**: The port where the passenger boarded (C = Cherbourg, Q = Queenstown, S = Southampton).

## Exploratory Data Analysis (EDA)

The first step I took was exploratory data analysis (EDA) to better understand the structure and quality of the data. Here are some key findings from the initial analysis:

Missing Values

The dataset had some missing values, especially in the Age and Cabin columns. About 20% of the Age values were missing, and the Cabin column had a lot of missing data (over 70%). Fortunately, the Embarked column only had one missing value.

Data Types

The dataset contains both numerical and categorical features. Numerical features like Age and Fare were easy to analyze, but the categorical features (like Sex and Embarked) needed to be converted into a numerical format before I could use them in a model.

Summary Statistics

The average age of passengers was around 29.7 years.

The Fare feature had a skewed distribution, with a few passengers paying much more for their tickets than others.

Around 38% of the passengers survived the Titanic disaster, which means the dataset is imbalanced—there are more non-survivors than survivors.

## Data Cleaning and Preprocessing

Next, I moved on to data cleaning to get the dataset ready for modeling. Some of the key preprocessing steps I performed include:

Handling Missing Data

For the missing Age values, I decided to fill them in with the median age of the passengers, as this is a common approach for dealing with missing numerical data.

The missing value in Embarked was filled with the most frequent value, which was 'S' (Southampton).

Since a lot of the Cabin data was missing, I filled those entries with 'Unknown' instead of removing the entire column.

Feature Encoding

The Sex column was converted into numerical values (0 = male, 1 = female) using label encoding.

For the Embarked column, I used one-hot encoding to create binary columns for each possible value (Cherbourg, Queenstown, Southampton).

Data Splitting: I split the data into two sets: 80% of the data for training the models and 20% for testing the models. This split helps to evaluate the model’s performance on unseen data.

## Model Building

I implemented three different machine learning models to predict passenger survival: Decision Tree, Random Forest, and Support Vector Machine (SVM). Here’s why I chose these models:

**Decision Tree:** This is a simple and interpretable model that splits the data into smaller subsets based on feature values. It’s easy to understand and visualize.

**Random Forest:** An ensemble of decision trees, this model is more robust and often performs better than a single decision tree by averaging the predictions of multiple trees.

**SVM:** A powerful model that finds the best boundary (or hyperplane) to separate the classes in the data. It’s useful when the data is complex and high-dimensional.

## Model Evaluation and Results

Once I trained the models, I evaluated them using a few different metrics: accuracy, precision, recall, and F1-score. Here’s how the models performed:

Decision Tree:

Accuracy: 79.88%

The Decision Tree model had decent accuracy, but it showed signs of overfitting—it performed better on the training data compared to the test data.

Random Forest:

Accuracy: 82.12%

The Random Forest model performed the best out of the three. It was able to handle the complexities of the data and showed better generalization to the test data.

Support Vector Machine (SVM):

Accuracy: 65.36%

The SVM model didn’t perform as well as the other two models. This was mainly due to the imbalanced classes (more non-survivors than survivors) and the fact that SVM can struggle with imbalanced data without additional preprocessing (like scaling or class weighting).

## Analysis of Results

Challenges: One of the biggest challenges was dealing with the missing data and the imbalanced dataset. The missing values in the Age and Cabin columns required some careful handling, and the imbalance in the Survived column meant that models like SVM didn’t perform as well without specific adjustments.

Model Selection: The Random Forest model turned out to be the best for this dataset because it combines multiple decision trees, which helps improve performance and reduces overfitting. The Decision Tree was also a solid choice but overfit the data, and the SVM struggled with class imbalances.

Performance Interpretation: The Random Forest’s higher accuracy and balanced predictions show that it was the best model for this dataset. The Decision Tree was easy to interpret but didn’t generalize as well to new data. The SVM’s poor performance highlights how important it is to preprocess data properly, especially when dealing with class imbalance.

## Conclusion

In conclusion, this analysis demonstrates how machine learning can be used to predict passenger survival on the Titanic. The Random Forest model was the most effective, with the best overall performance in terms of accuracy and class balance. The Decision Tree provided a good starting point but overfitted, while the SVM struggled with class imbalance.

Some future improvements could include exploring feature engineering to create new features (like family size from SibSp and Parch), handling the class imbalance more carefully, and trying other models like Logistic Regression to see if they perform better. Overall, this project was a great opportunity to apply various machine learning techniques and gain hands-on experience with data preprocessing and model selection.