**Data Science Project Report: Titanic Dataset Analysis**

**Objective**

The objective of this project is to apply data science concepts on the Titanic dataset to predict whether a passenger survived or not. The tasks involved include acquiring, cleaning, and preprocessing the data, performing exploratory data analysis (EDA), visualizing key insights, and building and evaluating a machine learning model.

**1. Introduction to the Dataset**

The dataset used for this project is the Titanic dataset, which contains information about passengers aboard the Titanic. This dataset has been widely used in data science and machine learning practice. It contains 891 rows and 12 columns, with attributes related to the passengers, such as their age, sex, class, fare, and whether they survived the tragic sinking of the ship.

The dataset is available at:

* Titanic Dataset - Kaggle

Key columns in the dataset:

* **PassengerId**: A unique ID for each passenger.
* **Pclass**: Passenger class (1, 2, 3).
* **Name**: Name of the passenger.
* **Sex**: Gender of the passenger.
* **Age**: Age of the passenger.
* **SibSp**: Number of siblings or spouses aboard.
* **Parch**: Number of parents or children aboard.
* **Ticket**: Ticket number.
* **Fare**: Fare paid for the journey.
* **Cabin**: Cabin number.
* **Embarked**: Port of embarkation (C = Cherbourg; Q = Queenstown; S = Southampton).
* **Survived**: Whether the passenger survived (1) or did not survive (0).

The target variable is **Survived**, which is a binary classification problem.

**2. Data Cleaning and Preprocessing**

**2.1 Handling Missing Values**

The dataset contains missing values in a few fields, most notably the Age column. The most frequent value in the column was utilized to fill in these missing values using a SimpleImputer. This keeps the dataset intact and prevents information loss from dropped rows.

**2.2 Dropping Unnecessary Columns**

In this research, columns like Name, Ticket, and Cabin were eliminated because they had no bearing on survival prediction. The machine learning model does not receive structured, predictive data from these columns.

**2.3 Encoding Categorical Variables**

The category data in the Sex and Embarked columns must be transformed into numerical form. The Sex (male = 0, female = 1) and Embarked (C = 0, Q = 1, S = 2) columns were encoded using a LabelEncoder.

**2.4 Feature Scaling**

To ensure that every feature contributed equally to the model, StandardScaler was used to scale the continuous variables, such as Age, Pclass, and Fare. Feature scaling standardizes the data and improves the performance of machine learning models like Random Forest by removing the mean and scaling the data to unit variance.

**2.5 Splitting the Data**

Using an 80-20 split, we divide the dataset into training and testing sets, with 20% going to testing and 80% going to training. This guarantees that the model can be assessed using data that hasn't been seen yet.

**3. Exploratory Data Analysis (EDA)**

**3.1 Distribution of Age**

We used a histogram with a Kernel Density Estimate (KDE) overlay to display the Age variable's distribution. With a few outliers, the plot displays an age distribution that is comparatively normal.

sns.histplot(df['Age'], kde=True, bins=30, color='blue')

plt.title('Age Distribution')

plt.show()

**3.2 Survival Rate**

To show the number of people who survived vs those who did not, we employed a count plot. The plot indicates that a greater proportion of passengers perished than survived.

sns.countplot(x='Survived', data=df, palette='Set2')

plt.title('Survival Count')

plt.show()

**3.3 Correlation Heatmap**

To comprehend the connections between the features, a correlation heatmap was employed. We were able to determine that the target variable, Survived, has a moderate association with both Age and Fare.

sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt='.2f')

plt.title('Correlation Heatmap')

plt.show()

**4. Machine Learning Model Building and Evaluation**

**4.1 Model Selection**

We select the Random Forest Classifier as the machine learning model for this classification task. To improve prediction accuracy and avoid overfitting, the Random Forest ensemble learning technique employs a number of decision trees.

**4.2 Model Training**

We trained the Random Forest Classifier using the training data (X\_train and y\_train). We used 100 estimators (trees) in the forest and set the random state for reproducibility.

model = RandomForestClassifier(n\_estimators=100, random\_state=42)

model.fit(X\_train, y\_train)

**4.3 Model Evaluation**

After training the model, we used the test set to make predictions and evaluated the model using several evaluation metrics.

**4.3.1 Accuracy**

The accuracy of the model on the test set is 0.8045, which means the model correctly predicted the survival of 80.45% of the passengers.

accuracy = accuracy\_score(y\_test, y\_pred)

print(f"Accuracy: {accuracy:.4f}")

**4.3.2 Confusion Matrix**

A confusion matrix was generated to visualize the performance of the classification model. It shows that the model predicted 177 survivors correctly, but misclassified 42 non-survivors as survivors, indicating room for improvement in handling the negative class.

cm = confusion\_matrix(y\_test, y\_pred)

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', cbar=False)

plt.title('Confusion Matrix')

plt.show()

**4.3.3 Classification Report**

The classification report provides key metrics such as precision, recall, and F1-score. The model performed well with a high recall (0.78) for survivors, ensuring that most of the passengers who survived were correctly identified.

print(classification\_report(y\_test, y\_pred))

**5. Conclusion**

In this project, we applied various data science concepts to the Titanic dataset. We cleaned and preprocessed the data by handling missing values, encoding categorical features, scaling numerical features, and splitting the data into training and test sets. We performed exploratory data analysis (EDA) to understand the dataset better and visualized key insights, such as the distribution of Age and the survival rate.

A Random Forest Classifier was trained on the dataset, and we evaluated the model using accuracy, confusion matrix, and classification report. The model achieved an accuracy of 80.45% on the test set. Although the model performed well, there is potential for improvement, particularly in predicting non-survivors.

This analysis highlights the importance of feature engineering, model selection, and evaluation metrics in solving real-world classification problems.

**6. References**

* Kaggle Titanic dataset: <https://www.kaggle.com/c/titanic/data>
* Scikit-learn documentation for Random Forest and evaluation metrics: <https://scikit-learn.org/stable/modules/ensemble.html>