Titanic Data analysis

Problem Description :

The Titanic Data Analysis project aims to explore and analyze the dataset containing information about passengers on board the ill-fated RMS Titanic, which sank on its maiden voyage after colliding with an iceberg on April 15, 1912. This project seeks to gain insights into the factors that influenced passenger survival and build predictive models to determine the likelihood of survival based on various features.

Dataset Information:

- The dataset consists of two files: "train.csv" and "test.csv."
- "train.csv" contains the training data with the target variable 'Survived,' indicating whether a passenger survived (1) or not (0).
- "test.csv" contains the test data without the 'Survived' column, which will be used for making predictions.

Each row in the dataset represents information about an individual passenger and includes various features, such as age, sex, ticket class, fare, number of siblings/spouses aboard, number of parents/children aboard, embarked port, and cabin number.

Background Information: The RMS Titanic was a British passenger liner deemed "unsinkable" and one of the largest and most luxurious ships of its time. Tragically, during its maiden voyage from Southampton to New York City, the Titanic collided with an iceberg in the North Atlantic Ocean and sank, resulting in the loss of more than 1,500 lives out of around 2,224 passengers and crew on board.

This maritime disaster was one of the most infamous and devastating events in history, prompting extensive investigations, inquiries, and analyses. Researchers have conducted numerous studies to understand the factors that influenced passenger survival during this tragic event.

Objective: The main objective of this Titanic Data Analysis project is to perform a comprehensive exploratory data analysis (EDA) and build predictive models that can determine the probability of passenger survival based on various factors such as age, gender, ticket class, and family size.

Data Exploration and Analysis: The project will start by loading and exploring the dataset to gain insights into the distribution of various features and the proportion of survivors among different groups. The analysis will include visualizations such as histograms, bar plots, and heatmaps to identify correlations and patterns in the data.

Feature Engineering and Data Preprocessing: To prepare the data for modeling, feature engineering will be performed to extract relevant information from certain columns, handle missing values, and transform categorical variables into numerical representations. Additionally, numerical features may be binned into categories to better capture underlying relationships.

Model Building and Evaluation: Several classification models will be trained on the training dataset, including logistic regression, support vector machines, random forest, and others. The performance of each model will be evaluated using appropriate metrics, and the best-performing model will be selected for making predictions on the test dataset.

Predictions and Submission: The chosen model will be used to predict passenger survival on the test dataset, and the results will be submitted in the required format for evaluation. The goal is to achieve the highest accuracy possible in predicting survival outcomes for the test data.

Significance: The Titanic Data Analysis project holds historical significance as it aims to uncover insights into the factors that played a role in determining passenger survival during the tragic sinking of the Titanic. It also demonstrates the application of data analysis and machine learning techniques to a real-world dataset, providing valuable insights into the importance of data-driven decision-making. The project may also serve as a learning resource for data analysts and aspiring data scientists seeking to enhance their skills in exploratory data analysis and predictive modeling.

Possible Framework :

1. Introduction:

- Introduce the Titanic Data Analysis project and provide a brief overview of the dataset and its features.
- Mention the main objective of the project, which is to analyze the data, explore patterns, and build predictive models for passenger survival.
- Explain the significance of the project and its historical context related to the sinking of the RMS Titanic.

2. Data Loading and Exploration:

- Load the "train.csv" and "test.csv" datasets using pandas.
- Display basic information about the datasets, such as the number of rows, columns, and data types.
- Perform initial data exploration to understand the distribution of features and the target variable 'Survived'.
- Create visualizations like histograms and bar plots to analyze the distribution of different features and the proportion of survivors.

3. Data Cleaning and Preprocessing:

- Check for missing values in the datasets and decide on the appropriate strategy to handle them (e.g., imputation or dropping rows/columns).
- Perform feature engineering to extract relevant information from the 'Name' column, such as titles (Mr., Mrs., Miss, etc.), and create a new feature 'Title'.
- Convert categorical variables like 'Sex', 'Embarked', and 'Title' into numerical representations for model training.
- Bin numerical features like 'Age' and 'Fare' into appropriate categories for better model performance.

4. Exploratory Data Analysis (EDA):

- Further explore relationships between features and survival using visualizations like heatmaps, box plots, and point plots.
- Analyze the impact of variables like 'Pclass', 'Sex', 'Age', 'FamilySize', and 'Fare' on passenger survival.

- Draw insights from the EDA and identify interesting patterns and correlations in the data.

5. Model Building:

- Prepare the data for model training by splitting the training dataset into features (X_train) and target variable (Y_train).
- Implement various classification algorithms like Logistic Regression, Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Random Forest, and Naive Bayes for model building.
- Train each model using the training data and evaluate its performance using suitable metrics like accuracy, precision, recall, and F1-score.

6. Model Evaluation and Selection:

- Compare the performance of each model to determine which one provides the best predictive power for passenger survival.
- Select the model with the highest accuracy and most suitable for the dataset.

7. Predictions and Submission:

- Use the selected model to predict passenger survival on the test dataset ('test.csv').
- Prepare the predictions in the required format for submission.
- Submit the results to the evaluation platform for scoring and comparison with other participants.

8. Conclusion:

- Summarize the findings of the Titanic Data Analysis project.
- Recap the insights gained from the data exploration and analysis.
- Discuss the predictive model's performance and its significance in determining passenger survival.

9. Future Enhancements:

- Suggest possible future improvements to the analysis, such as trying different modeling techniques, feature engineering, or feature selection methods.

- Highlight other areas of exploration, such as the impact of different age groups, fare bands, or embarkation ports on survival.

10. References:

- Provide references to any external sources, articles, or papers used during the project for data analysis or model evaluation.

Code Explanation :

*If this section is empty, the explanation is provided in the .ipynb file itself.

1. Data Loading and Exploration: The code begins by importing necessary libraries like pandas, numpy, seaborn, and matplotlib. These libraries are widely used in data analysis and visualization. Next, it loads the Titanic dataset from two CSV files: 'train.csv' and 'test.csv'. The 'train.csv' file contains data for training the predictive models, while the 'test.csv' file is used for evaluation and prediction.

2. Data Exploration: Once the dataset is loaded, the code starts exploring it. It displays the column names using print(train_df.columns.values) and prints the first few rows of the 'train_df' DataFrame using train_df.head(). These steps help to get an overview of the available data.

3. Data Information: The code then prints information about the datasets using train_df.info() and test_df.info(). This provides a summary of the dataset, including the number of non-null values in each column, data types, and memory usage. It helps to identify missing values and understand the nature of the dataset.

4. Descriptive Statistics: The code calculates descriptive statistics for the numerical columns in the dataset using train_df.describe(). This gives important insights like the count, mean, standard deviation, minimum, and maximum values for each numerical feature.

5. Grouping and Aggregation: The code performs grouping and aggregation operations to analyze the impact of various features on passenger survival. It groups the data by 'Pclass' (passenger class), 'Sex', 'SibSp' (number of siblings/spouses aboard), and 'Parch' (number of parents/children aboard) and calculates the average survival rate for each group using train_df[['Pclass', 'Survived']].groupby(['Pclass'], as_index=False).mean(). This helps to identify trends and patterns in the data.

6. Data Visualization: The code uses visualization techniques to represent data visually. It creates various plots like histograms, bar plots, and point plots using seaborn and matplotlib libraries. For example, it uses the sns.FacetGrid to create grids of multiple plots based on different factors like 'Pclass', 'Sex', and 'Embarked'. These visualizations help in understanding relationships between features and passenger survival.

7. Data Cleaning and Feature Engineering: Next, the code performs data cleaning and feature engineering. It removes irrelevant columns like 'Ticket' and 'Cabin' from the dataset, as they may not contribute significantly to the analysis. It also creates a new feature called 'Title' by extracting titles from passenger names. This new feature can provide additional information about a passenger's social status.

8. Data Transformation: The code transforms categorical variables like 'Sex', 'Embarked', and 'Title' into numerical representations using mapping and one-hot encoding techniques. This conversion is necessary for the machine learning models to understand and process the data.

9. Age Imputation: The code handles missing values in the 'Age' column by imputing the median age based on 'Sex' and 'Pclass'. It calculates the median age for each combination of 'Sex' and 'Pclass' and fills the missing 'Age' values accordingly.

10. Age Binning: After imputing missing 'Age' values, the code bins the age values into categories ('AgeBand') to group passengers into different age groups. This can help in capturing age-related patterns in survival.

11. Family Size and IsAlone: The code creates a new feature called 'FamilySize' by summing up 'SibSp' and 'Parch', which represents the total number of family members aboard for each passenger. It then derives another feature called 'IsAlone' to indicate whether a passenger is traveling alone or with family.

12. Fare Binning: The code bins the 'Fare' column into categories ('FareBand') to group passengers based on the ticket fare they paid. This can help in identifying fare-related trends in survival.

13. Model Building and Evaluation: Finally, the code builds predictive models using various machine learning algorithms like Logistic Regression, Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Random Forest, Naive Bayes, Perceptron, Stochastic Gradient Descent (SGD), Linear SVC, and Decision Tree. It splits the 'train_df' dataset into features (X_train) and target variable (Y_train) for model training.

Each model is trained using the training data and evaluated on its performance using accuracy as the evaluation metric. The code then stores the accuracy scores of each model in a DataFrame.

14. Model Comparison: The code compares the performance of all the models by creating a DataFrame with model names and their respective accuracy scores. It sorts the DataFrame in descending order of accuracy to identify the best-performing model.

15. Conclusion: The code does not provide a formal conclusion, but based on the accuracy scores, we can conclude which model performed the best on the given dataset.

16. Future Work: The code does not include future work suggestions, but possible improvements could involve hyperparameter tuning, feature selection, and ensemble methods to further enhance the predictive performance.

Overall, the code demonstrates the entire data analysis process, from data loading and exploration to feature engineering, model building, and evaluation. It leverages data visualization and descriptive statistics to gain insights into the dataset and make informed decisions during the analysis. The ultimate goal is to build a predictive model that can accurately predict passenger survival based on their characteristics.

Future Work :

Step 1: Collect More Data (If Possible) Collecting additional relevant data can provide more insights and improve the accuracy of predictive models. For example, information on passenger occupations, deck levels, or specific cabin locations could be valuable in understanding survival patterns.

Step 2: Feature Engineering Explore and create new features that might have significant impact on passenger survival. For instance, combining 'SibSp' and 'Parch' to create a 'FamilySize' feature, or analyzing the ticket number to extract information about group bookings.

Step 3: Handle Missing Data In real-world datasets, missing data is common. Apply more sophisticated imputation techniques or use machine learning models to predict missing values based on other features.

Step 4: Advanced Data Visualization Use advanced data visualization techniques to visualize complex relationships and correlations in the dataset. Tools like Plotly or Tableau can help create interactive and informative visualizations.

Step 5: Feature Selection Not all features contribute equally to the predictive model. Implement feature selection techniques to identify the most relevant features for training the model and discard irrelevant ones, reducing overfitting.

Step 6: Hyperparameter Tuning Tune hyperparameters of machine learning models to optimize their performance. Grid Search or Random Search can be used to find the best combination of hyperparameters.

Step 7: Cross-Validation Perform cross-validation to ensure the model's generalization ability. Divide the training data into multiple folds and train the model on different subsets to obtain a more robust estimate of the model's performance.

Step 8: Model Ensemble Combine predictions from multiple models using ensemble methods like Voting Classifier, Stacking, or Boosting. This can improve the overall accuracy and robustness of the final prediction.

Step 9: Model Interpretability Explore model interpretability techniques to understand how the predictive model makes decisions. Techniques like SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-Agnostic Explanations) can provide insights into model predictions.

Step 10: Deploy and Monitor Once a satisfactory model is achieved, deploy it in a real-world environment. Monitor the model's performance over time and update it as needed. Consider automation and integration with other systems for seamless use.

Step-by-Step Guide for Future Work:

- **Collect Additional Data (if possible**): Look for other datasets that might complement the existing data. Make sure to clean and preprocess the new data to match the format of the current dataset.
- **Feature Engineering:** Analyze the existing features and create new ones based on domain knowledge and insights from the data. Consider feature interactions and transformations.
- **Handling Missing Data:** Implement advanced techniques to impute missing values or use machine learning models to predict missing data based on other features.
- **Data Visualization:** Use libraries like Plotly or Tableau to create more interactive and insightful visualizations.
- **Feature Selection:** Apply feature selection techniques to identify the most important features for training the model.
- **Hyperparameter Tuning:** Use Grid Search or Random Search to find the best combination of hyperparameters for each model.
- **Cross-Validation:** Perform cross-validation to assess the model's generalization performance.
- **Model Ensemble:** Combine predictions from multiple models using ensemble techniques.
- **Model Interpretability:** Use SHAP or LIME to gain insights into how the model makes predictions.
- **Deployment and Monitoring:** Deploy the final model in a real-world environment and monitor its performance. Update the model as needed to maintain accuracy and relevance.

Remember that future work is an iterative process, and each step may require experimentation and fine-tuning. Continuously analyze the results and explore new approaches to improve the overall performance of the predictive model.

Exercise Questions :

Exercise 1: What is the distribution of passenger ages in the Titanic dataset?

Answer: To understand the distribution of passenger ages, we can create a histogram with age bins and count the number of passengers falling into each bin. This will give us an idea of how the ages are distributed across the dataset.

Exercise 2: How does the survival rate vary across different passenger classes (Pclass)?

Answer: We can calculate the survival rate for each passenger class separately. To do this, we group the data by Pclass and compute the average survival rate for each group.

Exercise 3: What is the correlation between passenger age and fare?

Answer: We can calculate the correlation coefficient between passenger age and fare to measure the strength and direction of their relationship. A positive correlation indicates that as one variable increases, the other also tends to increase, and vice versa.

Exercise 4: How does the survival rate differ between male and female passengers?

Answer: We can calculate the survival rate for male and female passengers separately. By grouping the data by sex and computing the average survival rate for each group, we can compare the survival rates between genders.

Exercise 5: Which embarked port (S, C, or Q) has the highest survival rate?

Answer: We can calculate the survival rate for each embarked port (S, C, Q) and determine which port has the highest survival rate.

Exercise 6: Are passengers with higher socio-economic status (Pclass=1) more likely to survive?

Answer: We can calculate the survival rate for passengers with Pclass=1 and compare it to the overall survival rate to determine if higher socio-economic status is associated with higher survival rates.

Exercise 7: Does having family members onboard (SibSp and Parch) affect the survival rate?

Answer: We can calculate the survival rate for passengers with different family sizes (e.g., alone, with siblings/spouse, with parents/children) and compare the survival rates to understand if having family members onboard influences survival.

Exercise 8: What is the average age of male and female passengers in each passenger class (Pclass)?

Answer: We can group the data by Pclass and sex and calculate the average age for male and female passengers in each class to understand the age distribution across different groups.

Exercise 9: Are passengers with missing age values more or less likely to survive compared to those with known age values?

Answer: We can analyze the survival rate for passengers with missing age values and compare it to the survival rate for passengers with known age values to determine if missing age values have any impact on survival.

Exercise 10: Can we predict passenger survival using machine learning algorithms like Logistic Regression or Random Forest?

Answer: We can split the data into a training set and a test set, then use machine learning algorithms like Logistic Regression or Random Forest to train a model on the training set. Finally, we can evaluate the model's performance on the test set to see if it can accurately predict passenger survival.