**Title:** CodTech IT Solutions Internship - Task Documentation: Credit Card Fraud Detection, Titanic Survival Prediction using Python

**INTERN INFORMATION:**

**Name:** Panuganti Karthik

**ID:** ICOD5620

**Project Title: Titanic Survival Prediction**

**Objective:** The primary objectives of this code are:

* **Preprocessing:** Cleaning and imputing missing data to prepare it for further analysis and modeling.
* **Exploratory Analysis:** Understanding the relationships between various features (Sex, Pclass, Age) and the target variable (Survived). This will likely inform feature engineering and model selection in subsequent stages of the project.

**Problem Background:**

**The Titanic Disaster**

* In 1912, the RMS Titanic sank on its maiden voyage, resulting in the tragic loss of over 1500 lives.

**The Dataset**

* The Titanic dataset on Kaggle (<https://www.kaggle.com/c/titanic>) provides a passenger list with features including:
  + **Survival Status** (Target Variable)
  + **Socioeconomic Factors:** Passenger Class ('Pclass'), Sex
  + **Family Relations:** Number of siblings/spouses aboard ('SibSp'), Number of parents/children aboard ('Parch')
  + **Age**
  + **Ticket Fare**
  + **Cabin** (may reflect location on the ship)
  + **Port of Embarkation**

**The Problem**

The primary problem this code aims to tackle is:

* **Prediction of Survival:** Can we predict whether a passenger would have survived the Titanic disaster based on their available characteristics?

**Key Questions the Code Explores:**

* **Missing Data:** How to handle missing values ('Age', 'Embarked') in a way that preserves information.
* **Feature Relationships:** How do features like sex, passenger class, or age relate to a person's chance of survival?
* **Potential for Modeling:** The code suggests that subsequent steps will involve building a machine learning model to predict the "Survived" outcome.

**Why This Matters**

* **Historical Insights:** Analyzing the factors that contributed to survival offers insights into the social dynamics and potential biases present during the disaster.
* **Machine Learning Practice:** The Titanic dataset is a classic "starter" dataset for practicing classification modeling for machine learning.

**Approach Overview:**

1. **Setup**
   * **Import Libraries:** Loads necessary libraries (NumPy, Pandas, Matplotlib, Seaborn) for data manipulation and visualization.
   * **Suppress Warnings:** May streamline the output.
   * **Load Datasets:** Reads in "train.csv" and "test.csv"
2. **Initial Exploration**
   * **Dimensions and Overview:** train.shape, train.head(), train.info(), and test.info() provide an understanding of the dataset's structure, features, and potential missing data.
3. **Data Preprocessing**
   * **Handle Missing Values:**
     + **'Embarked':** Imputed with the most frequent value ('mode').
     + **'Cabin':** Filled with a placeholder 'U0' (presumably meaning 'Unknown').
     + **'Age':**
       - Data split into records with and without 'Age' values.
       - A Random Forest Regressor model is trained on passengers with known ages to predict missing ages based on other features.
       - Predicted ages are used to fill the gaps in the 'Age' column.
4. **Exploratory Data Analysis (EDA)**
   * **Survival Analysis:**
     + Pie chart visualizes the distribution of survival vs. non-survival.
     + Calculations and bar plots examine survival rates grouped by sex and passenger class.
     + Multi-variable analysis provides insights into how survival varies across combinations of sex and passenger class.
   * **Age Analysis:**
     + Histogram and box plot visualize the age distribution and potential outliers.
     + Kernel density plots (KDE Plots) explore the age distribution further, faceted by whether a passenger survived or not.

**Key Points**

* **Focus on Preprocessing and EDA:** The code primarily addresses data cleaning, imputation of missing values, and visualizing relationships between variables and the target ('Survived').
* **Sets the Stage for Modeling:** While a predictive model is not explicitly built within this code snippet, the data preparation and insights gained are essential prerequisites for selecting and training a machine learning model to predict passenger survival.

**Next Steps**

The logical progression from this stage would likely involve:

* **Feature Engineering:** Creating new features (e.g., extracting titles from 'Name', and categorizing ages or fares into bins) that might improve model performance.
* **Model Selection:** Choosing a classification algorithm (e.g., Logistic Regression, Decision Trees, Support Vector Machines, or ensemble methods).
* **Training and Evaluation:** Training the model on the 'train' set, and evaluating performance metrics on the 'test' set.

**Dependencies:**

* **Python**: Python 3.12
* **Libraries**:
  + NumPy: "For numerical computations and array manipulation."
  + Pandas: "For data loading, manipulation, and analysis."
  + Matplotlib and Seaborn: "For creating informative data visualizations."
  + Scikit-learn: "For building and evaluating machine learning models."
  + **Dataset:**
  + Source: Kaggle (<https://www.kaggle.com/c/titanic>)
  + Size: (891, 12)
  + Features: List key features, noting the 'Class' target variable

**Implementation**

1. **Data Loading and Initial Exploration**
   * **Loading Datasets:** train = pd.read\_csv('./Dataset/train.csv') and test = pd.read\_csv('./Dataset/test.csv') read the data into Pandas DataFrames.
   * **Basic Insights:**
     + train.shape provides dataset dimensions.
     + train.head() shows the first few rows for a quick preview.
     + train.info() and test.info() summarize data types, columns, and any missing values.
2. **Missing Value Handling**
   * **'Embarked' Imputation:**
     + Missing values in the 'Embarked' column are filled using the most frequent value (train['Embarked'].dropna().mode().values).
   * **'Cabin' Imputation:**
     + Missing values in 'Cabin' are filled with 'U0' (likely indicating "unknown").
   * **'Age' Imputation:**
     + The DataFrame is split into subsets with and without 'Age' values.
     + A Random Forest Regressor model is trained on the data containing known ages, using 'Survived', 'Fare', 'Parch', 'SibSp', and 'Pclass' as predictors.
     + Predicted ages from the model are used to fill in the missing 'Age' values.
3. **Exploratory Data Analysis (EDA)**
   * **Survival Distribution:** A pie chart visualizes the proportion of passengers who survived vs. those who did not.
   * **Survival by Sex:**
     + train.groupby(['Sex','Survived'])['Survived'].count() calculates grouped counts.
     + train[['Sex','Survived']].groupby('Sex').mean().plot.bar() creates a bar plot of average survival rate by sex.
   * **Survival by Class:** Similar grouping and bar plot visualization are used to examine survival rates across passenger classes ('Pclass').
   * **Multi-Variable Analysis (Sex & Class):** Survival rates are explored across combinations of sex and passenger class.
   * **Age Distributions:**
     + train['Age'].hist(bins=70) creates a histogram of age distribution.
     + train.boxplot(column='Age', showfliers=False) displays a box plot for age, aiding in outlier visualization.
   * **KDE Plots of Age by Survival:** Seaborn's FacetGrid and sns.kdeplot are used to generate Kernel Density Estimation plots of age distributions split by 'Survived' status.

**Code Explanation**

**1. Setup and Data Loading**

* **Import Libraries:** Necessary libraries are imported for data manipulation, statistical analysis, and visualization.
* **Suppress Warnings:** warnings.filterwarnings('ignore') potentially hides warning messages, making the output less cluttered.
* **Load Datasets:** The code loads the training and testing datasets from CSV files using pd.read\_csv().

**2. Initial Exploration**

* train.shape**:** Reveals the dimensions (rows, columns) of the training DataFrame, giving an idea of the dataset's size.
* train.head()**:** Displays the first few rows, providing a quick look at the data structure and column values.
* train.info()**and**test.info()**:** Summarize each DataFrame, including column names, data types, and the presence of any missing values.

**3. Missing Value Handling**

* **'Embarked' Imputation:**
  + Finds any missing values in the 'Embarked' column.
  + Replaces missing 'Embarked' values with the most frequent value (mode()).
* **'Cabin' Imputation:** Replaces missing 'Cabin' values with the placeholder 'U0' (presumably denoting "Unknown").
* **'Age' Imputation:**
  + Splits the DataFrame into records with known ages (age\_df\_notnull) and those with missing ages (age\_df\_isnull).
  + Trains a Random Forest Regressor model (RFR) on the data with known ages to predict ages based on other features ('Survived', 'Fare', 'Parch', 'SibSp', and 'Pclass').
  + Uses the trained model to predict the missing ages.
  + Updates the 'train' DataFrame with the predicted ages.

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

* **Survival Distribution:**
  + train['Survived'].value\_counts().plot.pie(autopct = '%1.2f%%') Generates a pie chart visualizing the proportions of passengers who survived and those who did not.
* **Survival Analysis by Sex:**
  + Calculates the number of survivors grouped by sex.
  + Creates a bar plot showing the average survival rate for each sex.
* **Survival Analysis by Class:** Performs similar analysis as above but focused on the passenger class ('Pclass') variable.
* **Multi-Variable Analysis (Sex & Class):** Examines how survival rates vary across combinations of sex and passenger class, visualized with bar plots.
* **Age Analysis**
  + **Histogram:** train['Age'].hist(bins=70) creates a histogram to visualize the distribution of ages.
  + **Box Plot:** train.boxplot(column='Age', showfliers=False) displays a box plot, helping to identify potential outliers in the age data.
  + **KDE Plots with Faceting:**
    - Uses Seaborn's FacetGrid to create subplots of Kernel Density Estimation (KDE) plots for the age distribution.
    - Facets the plot by survival status ('Survived'), visualizing the age distribution differences between survivors and non-survivors.

**Usage**

* **Data Preparation and Cleaning:**
  + Real-world datasets often contain missing values or inconsistencies. This code demonstrates basic techniques for:
    - Identifying missing values.
    - Imputing missing values with simple strategies (mode, placeholders). These provide a starting point; more sophisticated imputation might be needed in practice.
* **Exploratory Analysis for Insight Generation:**
  + The calculations and visualizations highlight how a dataset can be explored to uncover relationships between variables and potential factors impacting survival:
    - The importance of sex and passenger class in determining survival outcomes.
    - Differences in age distributions between survivors and non-survivors.
* **Precursor to Modeling:** While this code doesn't explicitly build a predictive model, it showcases essential steps that prepare you for the modeling stage:
  + **Handling missing values:** Most machine learning algorithms require complete datasets.
  + **Understanding feature relationships:** The EDA aids in potential feature engineering and model selection.

**How This Could Be Applied**

1. **Historical Analysis and Simulations:**
   * Researchers or those interested in the Titanic disaster could use these techniques to further examine factors contributing to survival and simulate "what-if" scenarios.
2. **Dataset Foundation for Model Building:** This code offers a strong base for these next steps in a survival prediction project:
   * **Feature Engineering:** Consider creating new features (e.g., family size, binning ages or fares) that might boost the prediction accuracy.
   * **Model Selection:** Train classification models (Logistic Regression, Decision Trees, etc.) on the 'train' set.
   * **Evaluation:** Evaluate the model performance on the 'test' set using metrics like accuracy, precision, recall, and F1-score.

**CONCLUSION**

1. **Addressing Missing Data:** Machine learning models generally cannot handle missing values. This code establishes methods to clean the data, making it suitable for modeling.
2. **Guiding Feature Engineering and Modeling:** The insights gained from visualization guide informed decisions about:
   * **Creating New Features:** EDA might inspire new features like family size or age categories.
   * **Selecting an Appropriate Model:** Understanding the relationships observed suggests classification algorithms as suitable choices for predicting survival.

**Important Notes**

* **Imputation Refinement:** Consider the potential biases introduced by using simple imputation techniques. More advanced methods (e.g., imputation based on class and sex) could be explored.
* **This is Not a Complete Prediction Model:** This code lays the groundwork for training and evaluating machine learning models that would predict passenger survival.

**Next Steps**

Building upon this foundation, the logical next steps would include:

* **Feature Engineering**
* **Model Selection and Training (e.g., Logistic Regression, Decision Trees, Support Vector Machines)**
* **Evaluation using the 'test' dataset**

**OUTPUT**











