****

**PROGRAMMING FOR AI (LAB)**

**Lab Task No 02**

**Submitted To :- Sir Rasikh Ali**

**Submitted By :- Aiman Ijaz**

**Roll No :- SU92-BSAIM-F23-010**

**Department :- Software Engineering**

**Section :- BSAI-4A**

**Kaggle Spaceship Titanic Competition Report**

**Introduction**

The **Spaceship Titanic** competition on Kaggle is a binary classification problem where the objective is to predict whether a passenger was transported to another dimension. This classification is based on various features such as demographic information, onboard spending, and travel details.

This report provides a structured approach, covering data exploration, preprocessing, model training, and predictions based on the dataset provided

**Dataset Overview**

The dataset consists of two main files:

* **train.csv**: Contains labeled data, including a Transported column, which serves as the target variable.
* **test.csv**: Contains unlabeled data, where predictions must be made based on the trained model.

**Features in the Dataset**

1. **PassengerId**: A unique identifier assigned to each passenger.
2. **HomePlanet**: The planet from which the passenger embarked.
3. **CryoSleep**: A boolean variable indicating whether the passenger was in cryosleep.
4. **Cabin**: The cabin number assigned to the passenger.
5. **Destination**: The final destination of the passenger.
6. **Age**: The age of the passenger.
7. **VIP**: A boolean variable indicating whether the passenger was a VIP.
8. **RoomService, FoodCourt, ShoppingMall, Spa, VRDeck**: The amount spent on different onboard facilities.
9. **Transported**: The target variable indicating whether the passenger was transported to another dimension (1 = Yes, 0 = No).

**Data Exploration**

The dataset was first loaded into a Pandas DataFrame for further analysis.

**Basic Inspection**

1. **Displaying first and last few rows** to understand the structure of the dataset.
2. **Checking data types and missing values** using train.info() and test.info().
3. **Statistical summary** of numerical columns using describe().
4. **Unique value count** for categorical features to understand their distribution.
5. **Counting missing values** to determine necessary preprocessing steps.

**Data Preprocessing**

**Handling Missing Values**

Several columns contained missing values, which were filled using the most appropriate method:

1. **Categorical Features** (PassengerId, HomePlanet, CryoSleep, Cabin, Destination, VIP, Name) → Filled using the **mode** (most frequently occurring value).
2. **Numerical Features** (Age, RoomService, FoodCourt, ShoppingMall, Spa, VRDeck) → Filled using the **mode**.

**Data Type Conversion**

1. **Boolean Features (CryoSleep, VIP, Transported)** were converted to integers (0 for False, 1 for True).
2. **Numerical Features (Age, RoomService, FoodCourt, ShoppingMall, Spa, VRDeck)** were explicitly converted to int64 to ensure consistency in data types.

**Encoding Categorical Variables**

Categorical features were converted into numerical values using **Label Encoding**, which assigns each unique category a number:

* **Columns Encoded**: HomePlanet, Cabin, Destination, Name
* **Label Encoding** was applied separately to the train and test datasets to maintain consistency.

After preprocessing, the cleaned train and test datasets were saved as encoded-train-data.csv and encoded-test-data.csv for further use.

**Model Training**

**Preparing the Data**

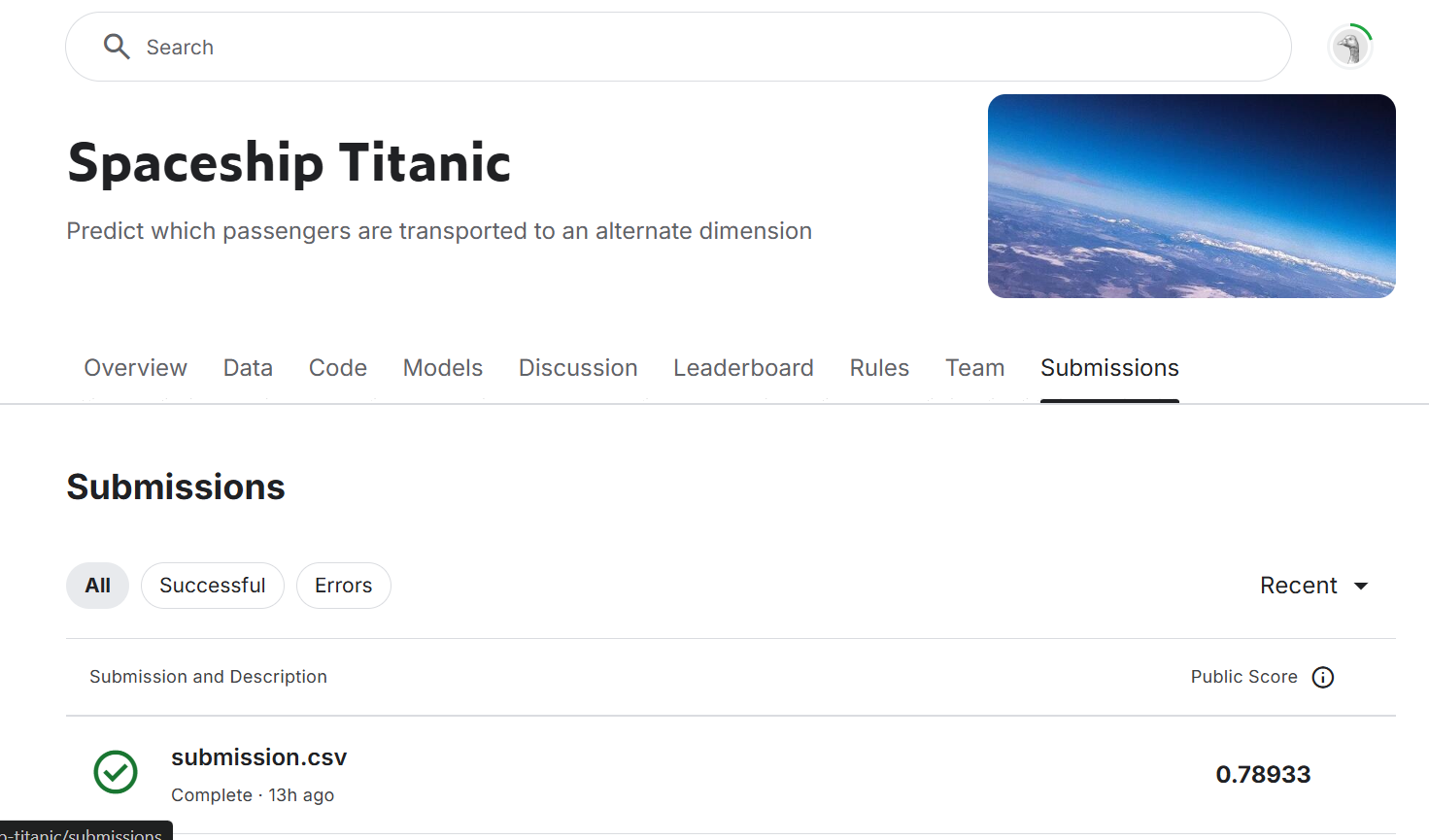
1. **Feature Selection**:
   * X\_train was created by dropping the target variable Transported.
   * y\_train was set as the Transported column.
   * X\_test was prepared by ensuring it contained the same features as X\_train.
2. **Model Selection**:
   * A **Random Forest Classifier** was chosen due to its robustness in handling structured data.
   * n\_estimators=100 was set to ensure sufficient trees in the forest.
   * random\_state=42 was used for reproducibility.
3. **Training the Model**:
   * The model was trained on X\_train and y\_train.
   * Predictions were made on X\_test.

**Submission File Creation**

After generating predictions, the results were stored in a submission file with the required format.

1. A DataFrame was created with two columns:
   * PassengerId: Taken from the test dataset.
   * Transported: Model predictions converted back to boolean values (True or False).
2. The DataFrame was saved as submission.csv, making it ready for submission to Kaggle.

**Output:**

****