**Lab 02**

****

**Name Syed M Haider Jafir**

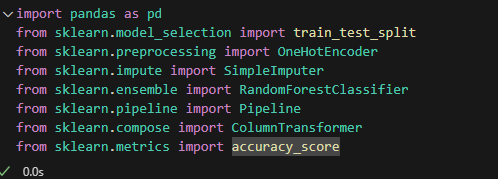
**Roll no SU92-BSAIM-S24-004**

**Section BSAI-4A**

**Subject PF (AILab)**

**Spaceship Titanic**

**Import Libraries**

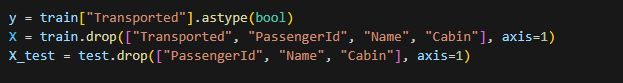


* **pandas**: for loading and manipulating the CSV files.
* **train\_test\_split**: to split the training data into training and validation sets.
* **OneHotEncoder**: converts categorical columns into numeric format for the model.
* **SimpleImputer**: fills in missing values.
* **RandomForestClassifier**: the machine learning model we’re using.
* **Pipeline**: lets us chain preprocessing steps and the model into one object.
* **ColumnTransformer**: applies different preprocessing to numerical vs. categorical columns.
* **accuracy\_score**: checks how accurate the model is on the validation set.

**Load the Data**



**Separate Target & Features**

****

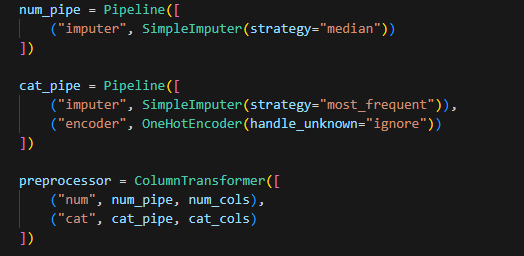
* y: the target column we want to predict (Transported).
* X: all features used to train the model.
* We drop:
  + - Transported (target column, not needed in features).
    - PassengerId (only an ID, not predictive).
    - Name and Cabin (text fields; we skip them for a basic model).

**Identify Column Types**

****

* Finds which columns are **categorical** (object dtype) and which are **numerical**.
* This is important because we treat these types differently during preprocessing.

**Preprocessing Pipelines**

****

* num\_pipe:
  + - * Uses the median to fill missing numeric values.

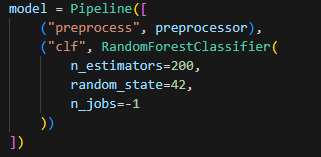
cat\_pipe:

* Fills missing categorical values with the most frequent category.
* Converts categories to numbers using One-Hot Encoding.

**Combine them:**

* Ensures that numeric columns go through num\_pipe and categorical columns through cat\_pipe in one unified step**.**

**Model Definition**

****

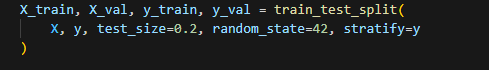
**Pipeline** runs:

1. preprocess → apply imputations + encoding.
2. clf → fit the **Random Forest** classifier.

**RandomForestClassifier**:

* n\_estimators=200: build 200 decision trees.
* random\_state=42: ensures reproducible results.
* n\_jobs=-1: use all CPU cores for faster training.

**Train/Validation Split**

****

**Splits data into:**

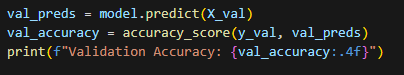
* 80% training → used to fit the model.
* 20% validation → used to check model accuracy before testing.
* stratify=y keeps the True/False ratio balanced in both sets.

**Train the Model**

****

* **Fits (learns patterns from) the training data.**

**Validate the Model**

****

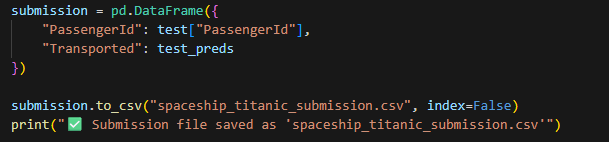
* **Predicts outcomes for the validation set.**
* **Calculates accuracy to see how well the model performs on unseen data.**

**Predict on Test Set**

****

* **Uses the trained model to predict Transported for passengers in test.csv.**

**Create Submission File**

****

**Creates a new DataFrame with:**

* PassengerId (from test set).
* Predicted Transported (True/False).

**Saves it as a CSV for Kaggle upload.**