## LAB 3: DATA CLEANING

## STUDENT NAME: ENYONG ABIGAIL

## Task 1: Introduction

## Brief explanation of architecture of the solution.

This piece of code sets up the foundation for understanding how data is being collected, cleaned and being transformed. The dataset used for this work was the TITANIC DATASET.

## TASK 2: set up infrastructure.

The IDE used for the development of this project was PyCharm in which I was able to install and import libraries such as;

- **pandas:** For data manipulation and analysis.

**-seaborn and matplotlib. pyplot:** For data visualization.

**-numpy:** For numerical operations.

**-SMOTE from imblearn:** For handling class imbalance by oversampling.

- **RandomForestClassifier from sklearn:** For building the classification model.

**-Various metrics from sklearn:** For model evaluation.

**-train\_test\_split from sklearn:** For splitting data into training and testing sets.

- **StandardScaler from sklearn:** For feature scaling.

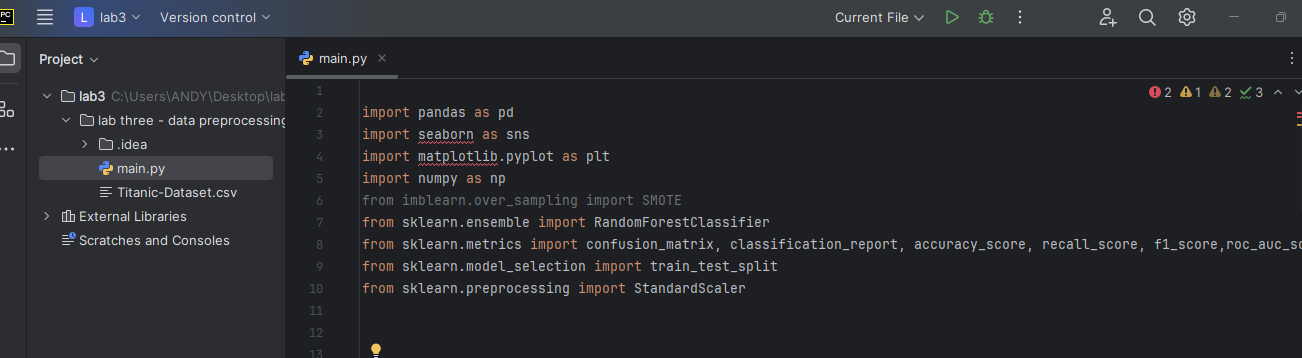


Fig 1. Importation of libraries

After importing the various dependencies needed, I went ahead to clean the data. This code carries out the following function

**Convert columns to numeric:** This handles non-numeric values by setting them to NaN.

**Fill missing values:** For 'Age' with median, 'Embarked' with mode, 'Fare' with median, and 'Parch' and 'SibSp' with 0.

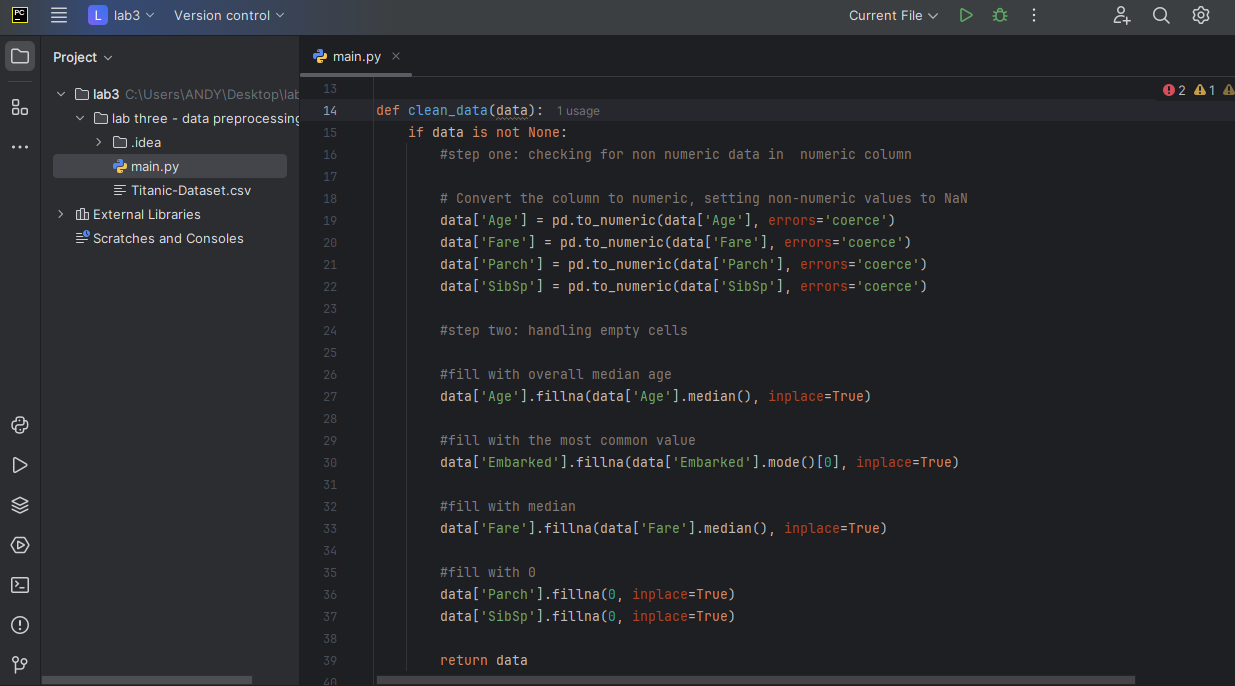


Fig 2. Checking for non-numeric values and filling missing values

The next task to be done is to clean the data. This process is done by the code below

**It Load dataset:** Reads the dataset into a pandas DataFrame and also it **Clean dataset:** Applies the clean data function to handle missing values and non-numeric data.

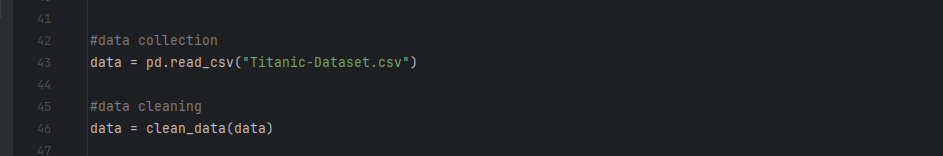


Fig 3. Showing data collection and cleaning

I went ahead to handle outliers I created **Box plots to** Visualize outliers in 'Fare' and 'Age'.

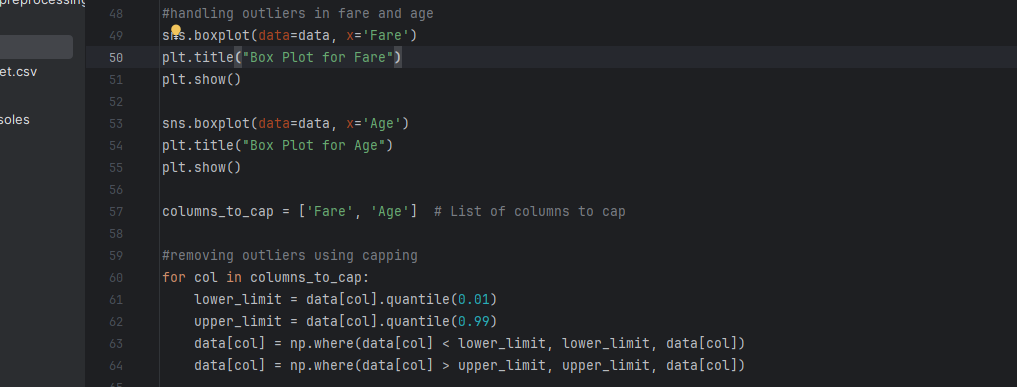


Fig 4. Handling of outliers

**The data was been normalized by standardizing the scales** 'Fare' and 'Age' to have mean 0 and variance

**I also demonstrated feature engineering by using**

**Family size:** Combines 'SibSp' and 'Parch'.

**Title extraction:** Extracts titles from names.

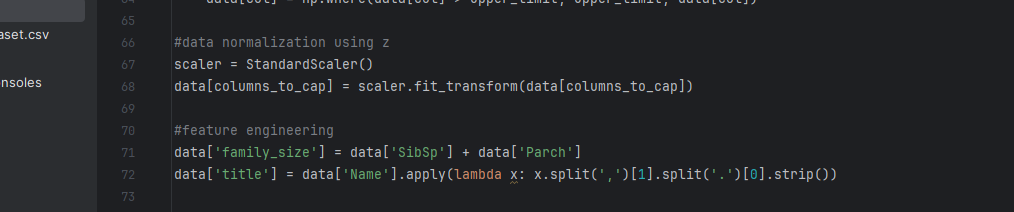
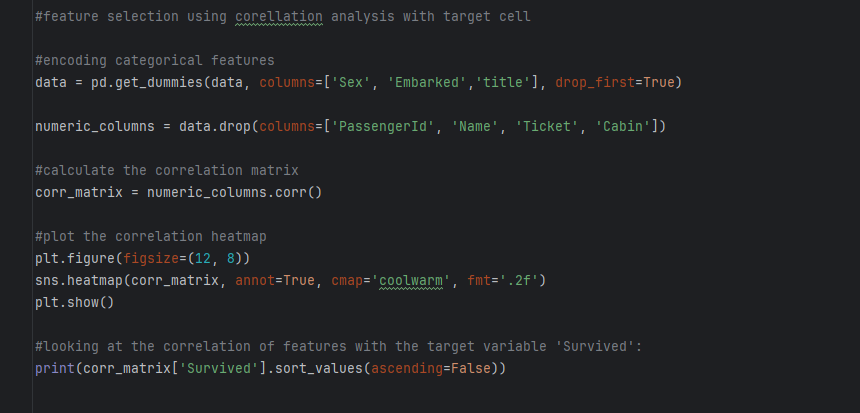


Fig 5. Showing data Normalization

I went ahead to select the various features calculate the correlation matrix and plot the correlation heatmap. I also looked at the correlation of features with the target variable 'Survived'

****

**Fig 6**

**Lastly, I move to building the model and evaluating it by;**

* **Drop irrelevant columns:** Removes unnecessary columns.
* **Feature and target separation:** Splits data into features (x) and target (y).
* **Train-test split:** Splits data into training and testing sets.
* **Random Forest model:** Initializes and trains the model on the training data.
* **Predict:** Makes predictions on the test data.



Fig 7

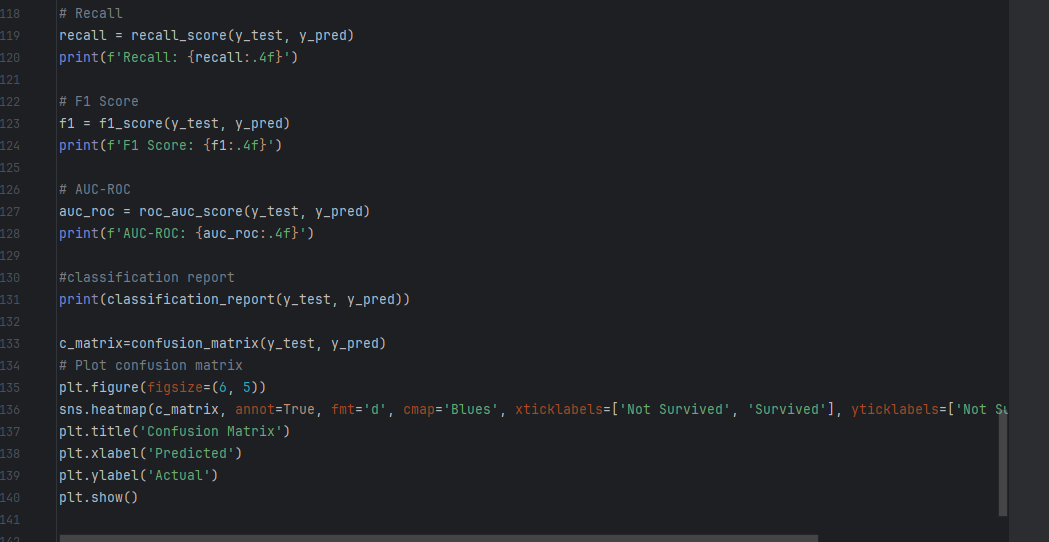


Fig 8