### **PPML PROJECT- PREDICTING FOOTBALLER’s VALUE** **Naveen ganesh KM-220901062**

### **Aim:**

To build a machine learning model using the provided files and libraries such as numpy, pandas, and scikit-learn, we can follow these general steps:

### **Objective:**

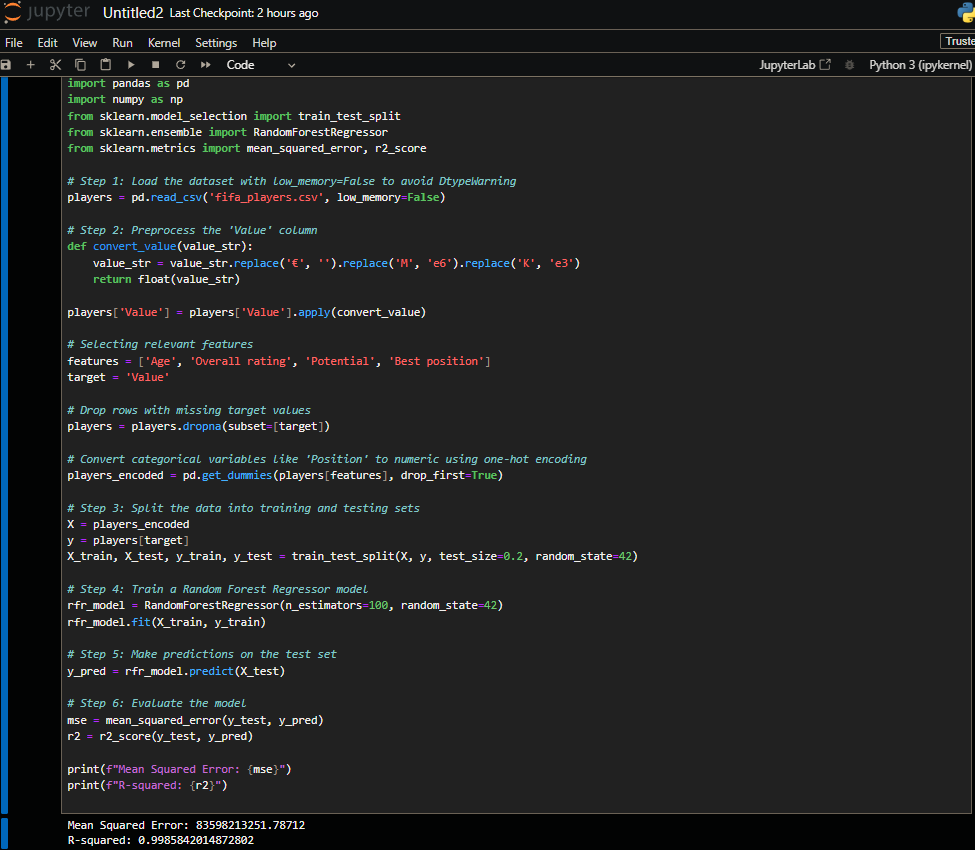
Based on the provided FIFA datasets, we'll create a model to predict a player's market value (Value) based on various attributes such as player rating (Overall), potential, age, and other features

**Algorithm:**

### **Algorithm: Predicting Player Value Using Linear Regression**

1. **Load the Data**:
   1. Import the necessary libraries: pandas, numpy, and scikit-learn.
   2. Load the dataset using pandas.read\_csv() with low\_memory=False to avoid mixed data type warnings.
2. **Preprocess the 'Value' Column**:
   1. Define a function convert\_value() to clean the 'Value' column.
      1. Remove the '€' symbol and convert abbreviations like 'M' (millions) and 'K' (thousands) into numbers.
   2. Apply the convert\_value() function to the 'Value' column to convert string values into numeric format.
3. **Select Features and Target**:
   1. Choose the features that are expected to impact player value, such as 'Age', 'Overall', 'Potential', and 'Position'.
   2. Define the target variable as 'Value'.
4. **Handle Missing Values**:
   1. Remove any rows that have missing values in the target column ('Value').
5. **Convert Categorical Features**:
   1. Apply one-hot encoding to the categorical features (e.g., 'Position') using pandas.get\_dummies().
   2. This converts the categorical column into binary indicator columns (0/1), which the model can handle.
6. **Split the Data**:
   1. Split the dataset into training and testing sets using train\_test\_split() from scikit-learn.
   2. Set 80% of the data for training and 20% for testing.
7. **Train the RandomForest Regression Model**:
   1. Initialize the RANDOM FOREST REGRESSOR() model from scikit-learn.
   2. Fit the model on the training data (X\_train and y\_train) using the fit() method.
8. **Make Predictions**:
   1. Use the trained model to predict player values on the test data (X\_test) using the predict() method.
9. **Evaluate the Model**:
   1. Evaluate the model’s performance using **Mean Squared Error (MSE)** and **R-squared (R²)** metrics.
   2. Calculate MSE using mean\_squared\_error() and R² using r2\_score() from scikit-learn.

**Implementation:**



* **Feature Engineering**: Transform the categorical variable 'Position' into numerical binary columns.
* **Linear Regression Model**: Predicts continuous target values based on a linear relationship between the features.
* **Model Evaluation**: MSE indicates prediction error, while R² shows the proportion of variance explained by the model.