Player value prediction report

الاسم	رقم الجلوس	السكشن	
اندرو ناصف امين يعقوب	20191700144	1	
اندريه عادل ابراهيم متى	20191700145	1	
ماريا موريس ايوب عبد المسيح	20191700473	3	
مايفى ايهاب رسمي مكين	20191700485	3	
مينا سامى انور سويحه	20191700676	5	

Feature Analysis:

- Preprocessing techniques applied:
    1. Feature encoding: Changed all columns with text features to numerical values using “Feature_Encoder” function by passing it the data and the columns with string values to be replaced with integer values.

    2. Missing values: Calculated the percentage of missing values in each column using “isnull().sum()” function on each column, then deleted columns with 50% or more missing values using “data.dorp(column)” function, then removed any missing values in the remaining columns using “data.dropna()” function, We tried to take the mean value or average of the missing values instead of discarding it but it increased the MSE so, we just removed it instead.

    3. Data handling: Changed the dates in “club_join_date” and “contract_end_year” to simple integers, also changed the values of “LS, ST, RS, LW, LF, CF, RF, RW, LAM, CAM, RAM, LM, LCM, CM, RCM, RM, LDM, CDM, RDM, RWB, LB, LCB, CB, RCB, RB, LWB" columns to integers.

    4. Feature scaling: Standardized the independent features present in the data in a fixed range to handle highly varying magnitudes or values or units, Applied Min-Max Normalization on the data.

- Feature selection:
    - Applied Data Correlation algorithm to find the relationship between multiple variables and attributes in the dataset with respect to the “value” column
    - After testing we found that 0.4 correlation was the best value that gave the least MSE and after applying correlation some of the features was discarded as they were not related enough to the “value”

    - Discarded features: “national_team, national_rating, national_team_position, national_jersey_number, tags, traits, birth_date, age, height_cm, weight_kgs, positions, nationality, preferred_foot, weak_foot(1-5), skill_moves(1-5), work_rate, body_type, club_team, club_position, club_jersey_number, club_join_date, contract_end_year, national_team, national_rating, national_team_position, national_jersey_number, crossing, finishing, heading_accuracy, volleys, dribbling, curve, freekick_accuracy, long_passing, acceleration, sprint_speed, agility, balance, shot_power, jumping, stamina, strength, long_shots, aggression, interceptions, positioning, vision, penalties, marking, standing_tackle, sliding_tackle, GK_diving, GK_handling, GK_kicking, GK_positioning, GK_reflexes, tags, traits, LS, ST, RS, LW, LF, CF, RF, RW, LAM, CAM, RAM, LM,

LCM, CM, RCM, RM, LWB, LDM, CDM, RDM, RWB, LB, LCB, CB, RCB, RB”

- Used features: “overall_rating, potential, wage, international_reputation(1-5), release_clause_euro, club_rating, short_passing, ball_control, reactions, composure”

Algorithms used:

- Polynomial Regression:
  - First, we split the data into training and testing sets using “train_test_split()” function, we split the data 20% test and 80% training

  - Then we choose the degree of the Polynomial the best degree was “3” as if we increased it more than that not only it will increase the MSE, but it will increase the complexity of the regression also, and if we tried to decrease the degree then the MSE will increase.

  - Then we transform the existing features to higher degree features using “fit_transform()” function

  - We fit the transformed features to Linear Regression model using “LinearRegression()” and “fit()” functions

- Then we predict on the training and test datasets using “predict()” function
- Finally, We calculate the MSE using the “metrics.mean_squared_error()” function giving it the actual Y and the predicted Y.

- Multi Linear Regression:
  - First, we call the Linear Regression model using “LinearRegression()”function

  - Then we fit the model using the training data we got before using “fit()” function

  - We get the predicted Y using the “predict()” function

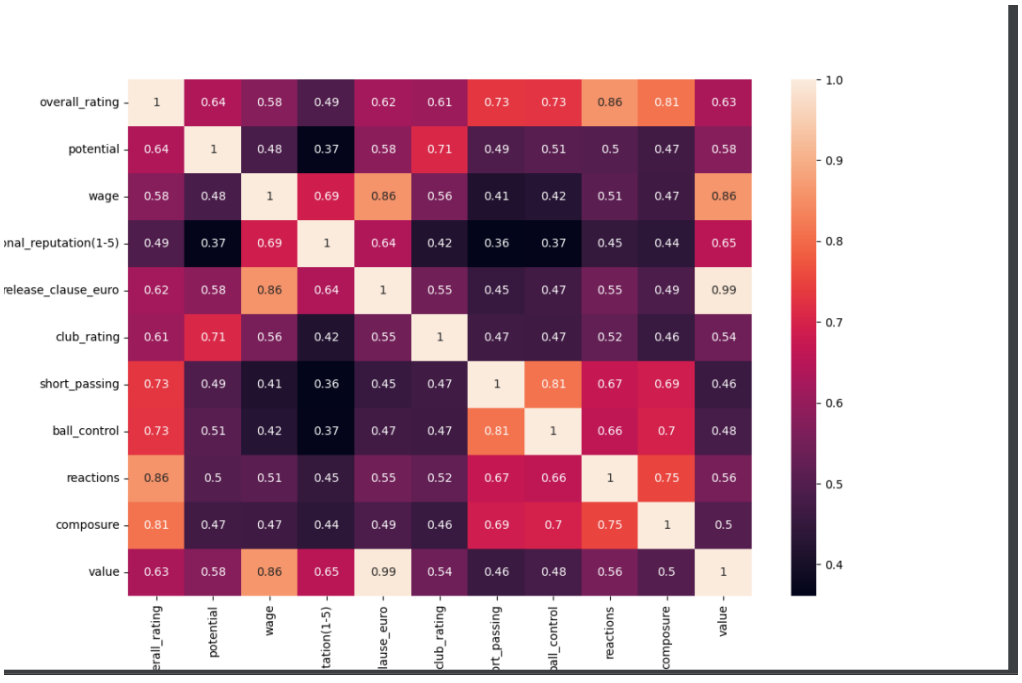
  - Finally, We calculate the MSE using the “metrics.mean_squared_error()” function giving it the Y test and the predicted Y.

Polynomial Regression	Multi linear Regression	Diff.
- It provides a great defined relationship between the independent and dependent variables - It can be more accurate than Multi linear regression but also more complex	- considers the influence of multiple independent variables on a dependent variable Y - looks at the relationships within a bunch of information. Instead of just looking at how one thing relates to another thing	Usage
-MSE = 48 243 709 936 -More accurate	-MSE = 97 810 814 096 -Less accurate	Results

High	Low	Training Time

- Difference between the algorithms used:
- Techniques used to improve results:
  - Used Train Validation Test split with test size equal to 90% and 10% train and validation

- Screen shots results:



Conclusion:

- This phase of the project made us realize the difference between the machine learning algorithms that can be used to solve the same problem
- First, we tried the multi linear Regression algorithm but, we had the intuition that the polynomial regression will give us better results without too much cost and that was proved when we tried the polynomial regression at degree “3” which gave us much better result than the multi linear and with little cost and complexity
- Also, we thought that using train validation test split will be better and it was also proved by the results
- We also learned how to handle missing values and how to handle different datatypes to apply different pre-processing techniques

- And, We applied feature selection using correlation algorithms to discard unnecessary features