**Player Value Prediction**

**Team id: CS\_28**

**Team Members:**

|  |  |  |  |
| --- | --- | --- | --- |
| Sec | DEP | ID | Name |
| 4 | **CS** | **20191700627** | **مريم حسام الدين حمدي** |
| 4 | **CS** | **20191700625** | **مريم اسماعيل علي محمد** |
| 5 | **CS** | **20191700671** | **مي محمد علي حسن** |
| 2 | **CS** | **20191700253** | **روان أحمد يوسف أحمد** |
| 1 | **CS** | **20191700100** | **اسراء عماد الدين سعيد أحمد** |

**Milestone 1**

# **Preprocessing techniques applied on the dataset:**

1. **Get dummies:** used on body\_type column to separate the categorical data into 3 columns; body type lean, body type normal, body type stocky.
2. **Replace:** replaces string values with numeric values that help describe the degree.
3. **Fillna:** replaces all null values with more suitable numeric values.
4. **Date\_Preprocessing:** replaces all values with the last 2 characters in the original string.
5. **LabelEncoder:** converts any string value to a numeric value.
6. **SimpleImputer:** replaces null values found in columns with numeric values using a certain strategy (ex: Median).
7. **FeatureScaling:** scales large numeric values.
8. **Body\_type Preprocessing:** a function to replace null values with a calculated numeric value that represents the body type by using BMI rules.

# **Dataset Analysis:**

We used the heat map function to know how each feature affects the other.

# **Regression Techniques Used:**

1. **XGBoost:** XGBoost is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library. It provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems.
2. **Polynomial Regression:** Polynomial Regressiona form of Linear regression known as a special case of Multiple linear regression which estimates the relationship as an nth degree polynomial. Polynomial Regression is sensitive to outliers so the presence of one or two outliers can also badly affect the performance.
3. **Random Forest:** Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

# **Differences Between Each Model and The Acquired Results:**

1. **XGBoost:**

* Accuracy: 0.9951267345309258 (99%)
* MSE: 145785361670.41998
* Training Time: about 4.5 sec

1. **Polynomial Regression:**

* Accuracy: 0.9863949342990068 (98%)
* MSE: 461662358925.458
* Training Time: about 0.3 sec.

# **Features used or Discarded:**

1. **Used:** overall rating, potential, wage, international reputation(1-5), skill moves(1-5), release clause euro, club rating, contract end year, national rating, age, crossing, finishing, short passing, volleys, dripping, curve, freekick accuracy, long passing, ball control, reactions, shoot power, stamina, long shots,positioning, vision, penalties, composure, traits.
2. **Discarded:** id, name, fullname, birthdate, hight(cm), weight(kg), national team, nationality, workrate, club team, club positions, club jersey number, club join date, national team position, national jersey number, tags, LS, ST, RS, LW, LF, CF, RF, RW, LAM, CAM, RAM, LM, LCM, CM, LB, LCB, CB, RCB, RB, RCM, RM, LWB, LDM, CDM, RDM, RWB, sliding tackle, GK diving, GK handling, GK kicking, GK positioning, GK reflex, body type lean, body type normal, body type stocky, preferred foot left, preferred foot right, workrate defense, positions, weak foot(1-5), heading accuracy, acceleration, sprint speed agility, balance, jumping, strength, aggression, interceptions, marking, standing tackle, workrate attacking.

# **Sizes of Training, Testing and Validation Sets:**

1. **XGBoost(Decision Tree):**

* Training Size: 0.8
* Testing Size: 0.2

1. **Polynomial Regression:**

* Training Size: 0.7
* Testing Size: 0.3

# **Data Visualization:**

# **Heat map:**

# **Conclusion:**

We got to use many different models to assess which is best like Decision Tree, Polynomial Regression, Random Forest, XGBoost and settled with the best two.

We tried to drop different features and see the relation between them and the error to get the least MSE.

We had many null values and we used different techniques to be able to fit it better with the data which helped more with the MSE and Accuracy.

**Milestone 1**

# **Classification Techniques Used:**

1. **AdaBoost with Decision Tree:** AdaBoost is an ensemble learning method (also known as “meta-learning”) which was initially created to increase the efficiency of binary classifiers. AdaBoost uses an iterative approach to learn from the mistakes of weak classifiers, and turn them into strong ones.

AdaBoost, short for Adaptive Boosting, is a statistical classification meta-algorithm.

* **Hyper Parameters of Decision Tree:**
* **Max\_depth**: 3 🡪 92.37730595196658

5 🡪 95.47511312217195

6 🡪 95.63974939088061

10 🡪 95.85798816568047

the value 5 gives us the highest testing accuracy with the minimum complexity

* **Hyper Parameters of Adaboost:**
* **learning\_rate:** 0.2 🡪 95.370692655760521

1. 🡪 94.67455621301775
   1. 🡪 95.23146536721198

The value 0.2 gives the testing accuracy

* **Training time** = 2.1542320251464844 sec.
* **Testing time** = 0.06086564064025879 sec.

1. **SVM:** The idea of SVM is that the algorithm creates a line or a hyperplane which separates the data into classes.

* **Hyper Parameters:**
* **C:** 1 🡪89.87121475809259

2 🡪89.73198746954402

3 🡪 90.01044204664113

The value 3 gives the highest testing accuracy.

* **Kernel Functions:** SVC with linear kernel 🡪 90.01044204664113

Linear SVC 🡪 43.543334493560735

SVC with rbf kernel 🡪 86.42533936651584

SVC with poly kernel 🡪 89.3491124260355

* **Training time** = 28.94691491127014 sec.
* **Testing time** = 0.33500170707702637 sec.

1. **Logistic Regression:** one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

* **Hyperparameters:**
* **Intercept\_scaling:** 1 🡪 72.43299686738601

2 🡪 75.67003132613992

4 🡪 77.06230421162547

The value 4 give the testing accuracy

* **C:** 4 🡪 78.48938391924817

2 🡪 77.75844065436826

1 🡪 77.27114514444831

The value 4 give the testing accuracy

* **Training time** = 0.6350951194763184 sec.
* **Testing time** = 0.009612798690795898 sec.

# **Feature Selection process:**

We used Information gain to select the top features, we droped the features that have information gain <0.39 as we found that it gives us the best accuracy.

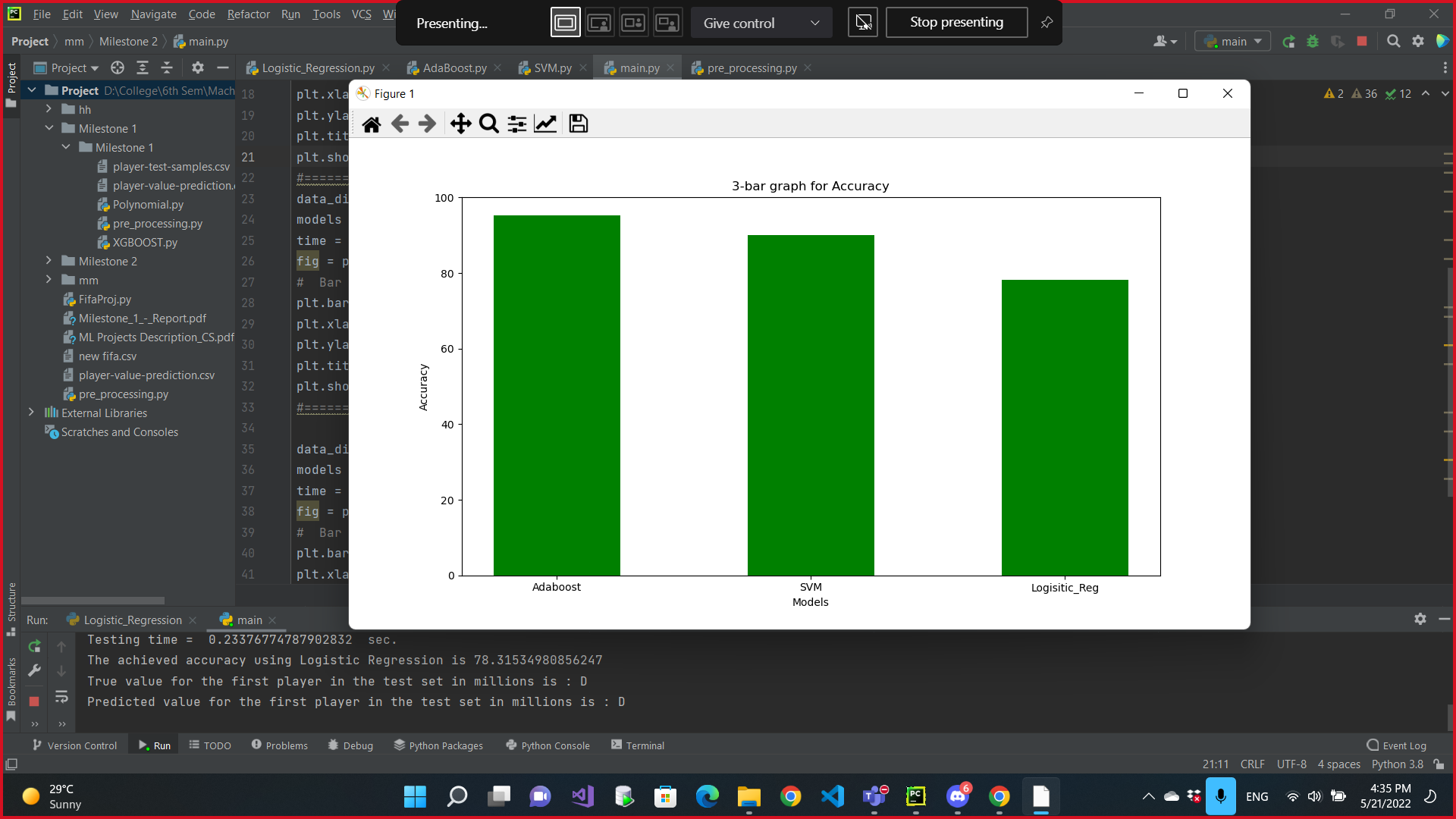
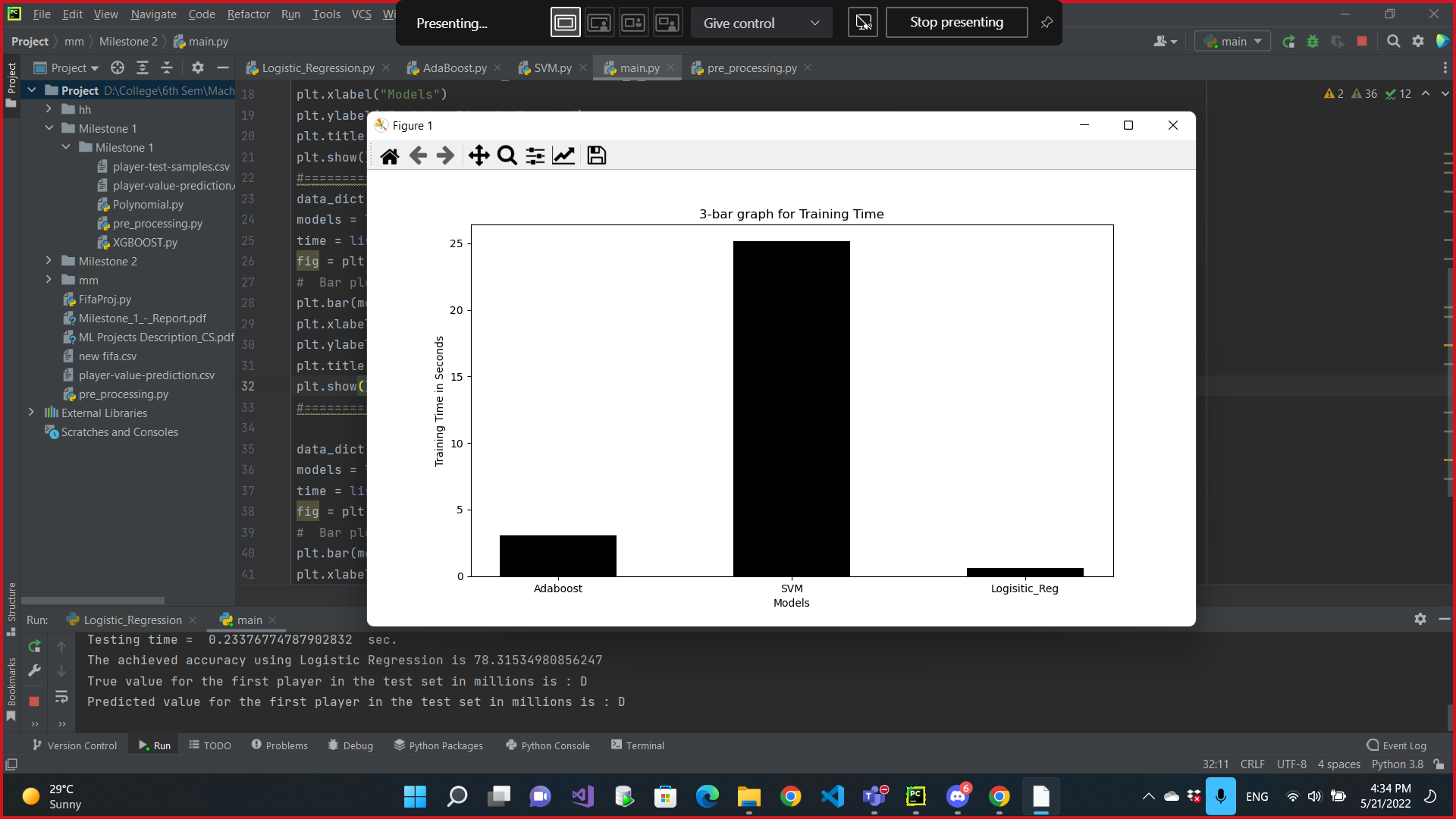
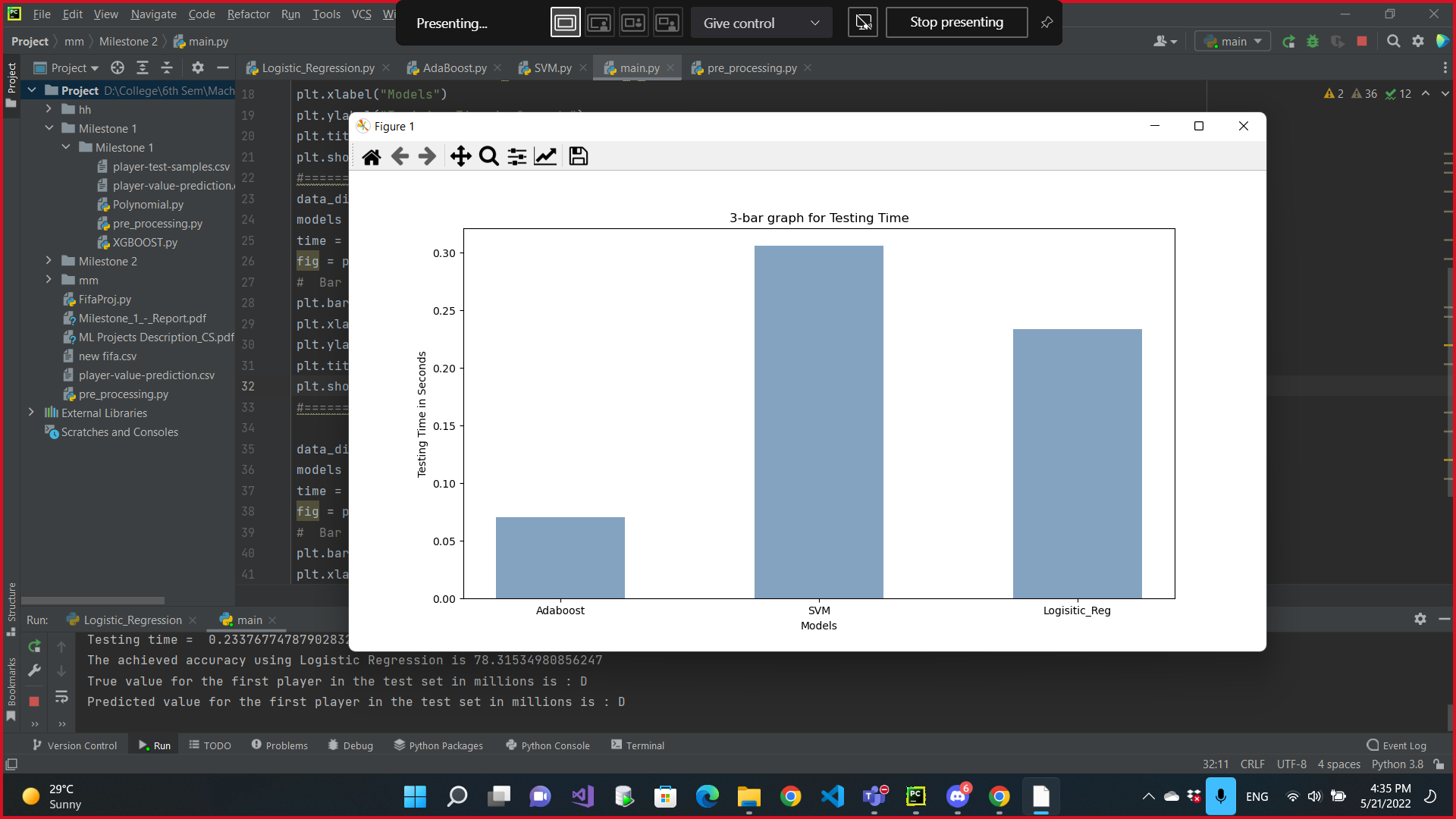
**The features that have information gain < 0.39 are: :** id, name, full\_name, birth\_date, height\_cm weight\_kgs, positions, nationality, preferred\_foot, weak\_foot(1-5), work\_rate, body\_type, club\_team, club\_rating, club\_position, club\_jersey\_number, club\_join\_date, national\_team, national\_rating, national\_team\_position, national\_jersey\_number, crossing, finishing,heading\_accuracy, short\_passing, volleys, dribbling, curve, freekick\_accuracy, long\_passin, acceleration, sprint\_speed, agility, balance, shot\_power, jumping, stamina, strength, long\_shots, aggression, interceptions, positioning, vision, penalties, composure, marking, standing\_tackle, sliding\_tackle, GK\_diving, GK\_handling, GK\_kicking, GK\_positioning, GK\_reflexes, tags, traits, LS, ST, RS, LW, CF, RW, LM, LCM, CM, RCM, LWB, LDM, CDM, RDM, RWB, LB, LCB, CB, RCB, RB

Aslo we dropped some feature manually as we found that this increases the accuracy.

**The manually dropped features are:** age, international\_reputation(1-5), contract\_end\_year, CF, LM

**So the used features are:** age, overall\_rating, potential,wage, international\_reputation(1-5), skill\_moves(1-5), release\_clause\_euro, contract\_end\_year, ball\_control, reactions, LF, RF, LAM, CAM, RAM, RM

# **Bar Graphs**

1. **Accuracy Bar Graph:**
2. **Training Time Bar Graph:**
3. **Testing Time Bar Graph:**

# **Conclusion:**

We got to use many different models to assess which is best and finally settled with Decision Tree (with adaboost), Logistic Regression, SVM with linear kernel function as they are the best three.

Also we Tried different hyper parameters with each model and settled with the best values for them.

We tried to drop different features and see the relation between them and the accuracy.

We also conclude that the used features differ from those of the regression models.