UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Machine Learning (APA)

Laboratory Project: SkillCraft

Group 12
David Nogales Pérez
Álex Ochoa Blasco

Barcelona January 6, 2022

Contents

1 Introduction 3

2 Data Exploration 4

3 Models Trained 8

4 Results Obtained 9

5 Model Chosen 14

6 Interpretability Analysis 15

7 Self-Assessment 16

8 References 17

Color codes:

- CT-Generic Competence evaluation criteria= 10%
- Task Evaluation Criteria= 40%
- Tasks that must be included in each section.

1 Introduction

In this Project, given some data, we are trying to predict the level of expertise of a player of the famous RTS game "Starcraft". Firstly we are making the pre-proccesing or the data, secondly we will re-sample the data for our models, and to finish with, we will compare the results of the models and chose the one that we consider to be the best. We are going to train 5 different models (3 Lineals/Quadratic and 2 non-linears) with the best hyperparameters for each one.

We will use this dataset that comes from the UCI dataset repository, It contains the data of 3395 game players of the famous RTS game "Starcraft" from 7 levels of mastery. Most of the variables are related to in-game performance of players, but we also have external variables like age, hours spent in the game. The target variable is League Index, which has 7 categories that represent the level of expertise of the player.

Related Previous Work

The investigation of previous work, has been done after our study, in order to not be influenced by it. Our dataset has been used in a Paper by Thompson JJ, Blair MR, Chen L, Henrey AJ, see in References. Their primary finding was that variables do change in importance across the skill levels. They also concluded that RTS game replays can track abilities of interest to cognitive science

2 Data Exploration

Balancing Data and Dealing with Null values

Observing the histogram plot of the target variable, one can see that the dataset is heavily unbalanced mainly due to the extreme categories. Given that we can't obtain more data we should study if we can afford to drop these categories, or try another methods such as merging them given that some categories in the Starcraft Ladder aren't too different in terms of skill.

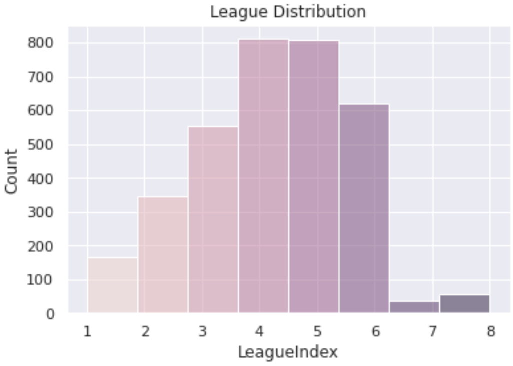

Fig. 2.1: Histogram Plot of LeagueIndex

Looking at the information of the dataframe, there are some variables which are numeric but appear as object and also there seemingly aren't any null values. After converting the variables to numeric we found that there are indeed null values in the dataset. We can see that most of our missing data is located in the 'Professional' category (8) and only 2 rows in the 5th category have missing data. A solution to this would be imputing all the missing data in the corresponding categories instead of dropping them since we could be dropping valuable data. But given that our data sample of these categories are to low there is no way to know if the values we would be imputing will be correct, so we will drop them.

To solve the problem regarding the unbalanced data we will merge categories 1,2 and 6,7 given that according to the global statistics of ranks of the population of active players in the game through different seasons, these categories have a smaller number of players in general so by merging them we will be able to better predict the remaining ranks.

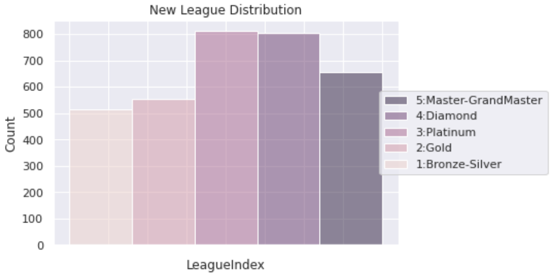

Fig. 2.2: Plot of New League Distribution

Detecting Outliers

Looking at the description of the variables one can rapidly tell the presence of outliers. For example in the variable *TotalHours* we have that someone played the game a million hours which is roughly 114 years. Similarly in *HoursPerWeek* the maximum of hours played in a week is 168 hours which would mean a week of playing the game without rest. Analyzing the violin plots of the dataset we can clearly observe the presence of outliers in different variables.

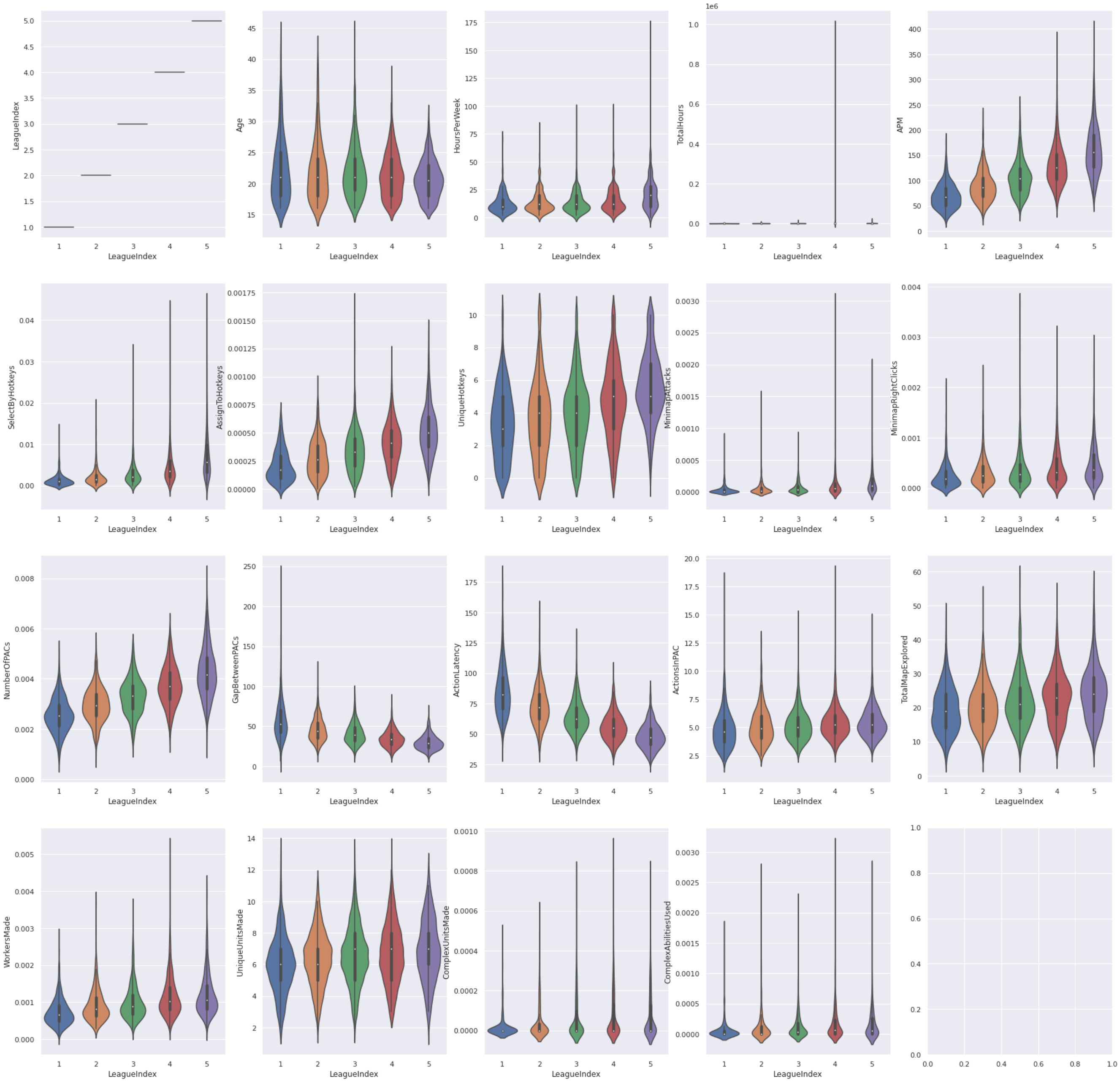

Fig. 2.3: Violin Plots of the dataset

Since most of the extreme values appear to be in the upper points of the violin plots we will erase values greater to the 99 percentile. Although outliers are still present in our data we can't erase or modify them since those data points may pertain to players who are close to climb the classification ladder.

Discretizing Age

Since our data does not have a discrete variable we are tasked to create one. To do that we chose Age which will be automatically discretized by the KBinsDiscretizer, which will create 3 new categories of same size.

Feature Selection

We will perform Feature Selection instead of Feature Extraction avoiding the change of variables' space performed by applying PCA. Observing the Correlation Heatmap one can tell that the target variable has a relatively high correlation with *APM* and *NumberOfPACs*, in contrast, we have that is inversely correlated with *ActionLatency* and *GapBetweenPACs*

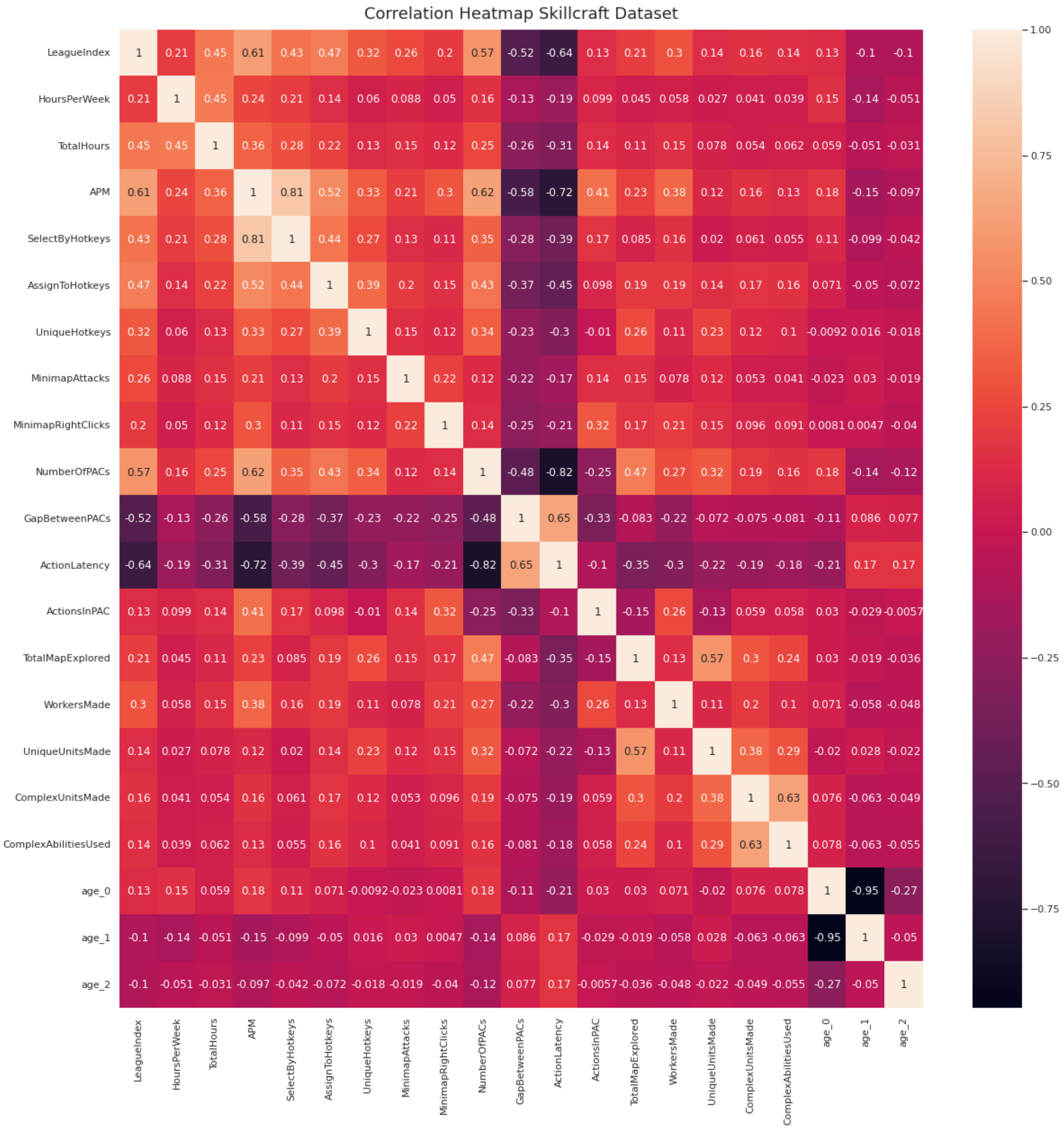

Fig. 2.4: Heatmap of the dataset

Looking at the absolute value of the correlations we have that at least 9 features have a relatively big impact on the target variable, meanwhile the rest does not seem to be that important.

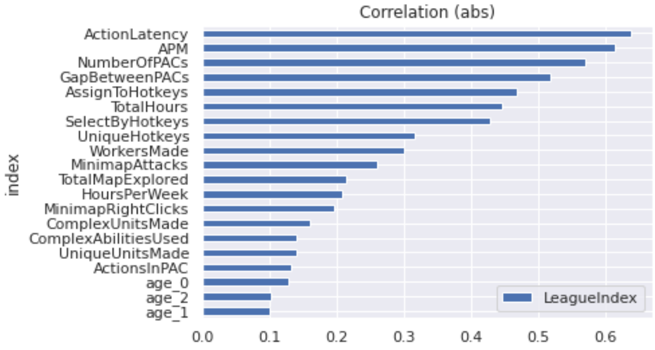

Fig. 2.5: Absolut value of the correlations

Feature Importance

According to the Random Forest Regressor we can explain aproximately 60% of the variance using only *APM*, *ActionLatency* and *TotalHours* variables.

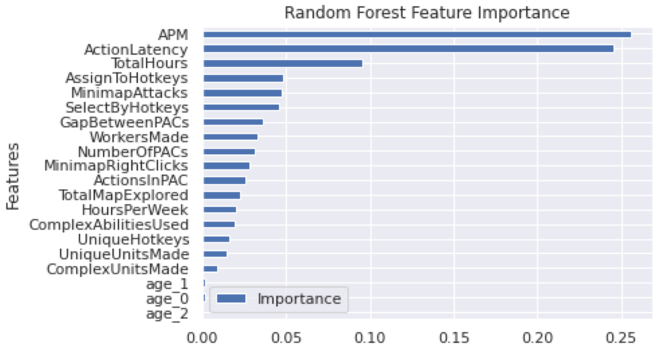

Fig. 2.6: Random forest features importance

Looking at the cumulative sum of the importances we can see that we can explain 90% of the variance using 12 features instead of 20.

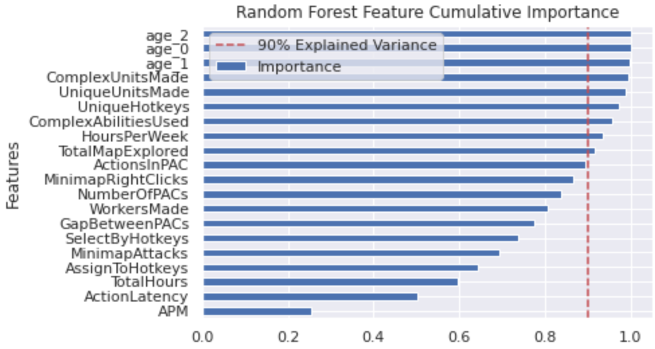

Fig. 2.7: Random forest features Cumulative importance

By intersecting the 10 most important features found with the correlation matrix and feature importances of the Random Forest we end up with 9 features instead of 20, which is a reduction of more than half of the variables.

3 Models Trained

Our target variable is Ordinal so the models chosen are regression models in order to avoid losing the inherent relationship between the ranks of the players and also lose prediction power. First we split the training and test data. Being 70% for training and 30% for testing. We use K-Fold Cross-Validation, having proved different values for K we decided to chose K = 5. We are using the following Models:

Linear/Quadratic

- Linear Regression
- Linear SVM
- Quadratic SVM

Non-linear

- SVM with RBF kernel
- Random Forest

4 Results Obtained

Linear Regression

First method we are using is the Linear Regression

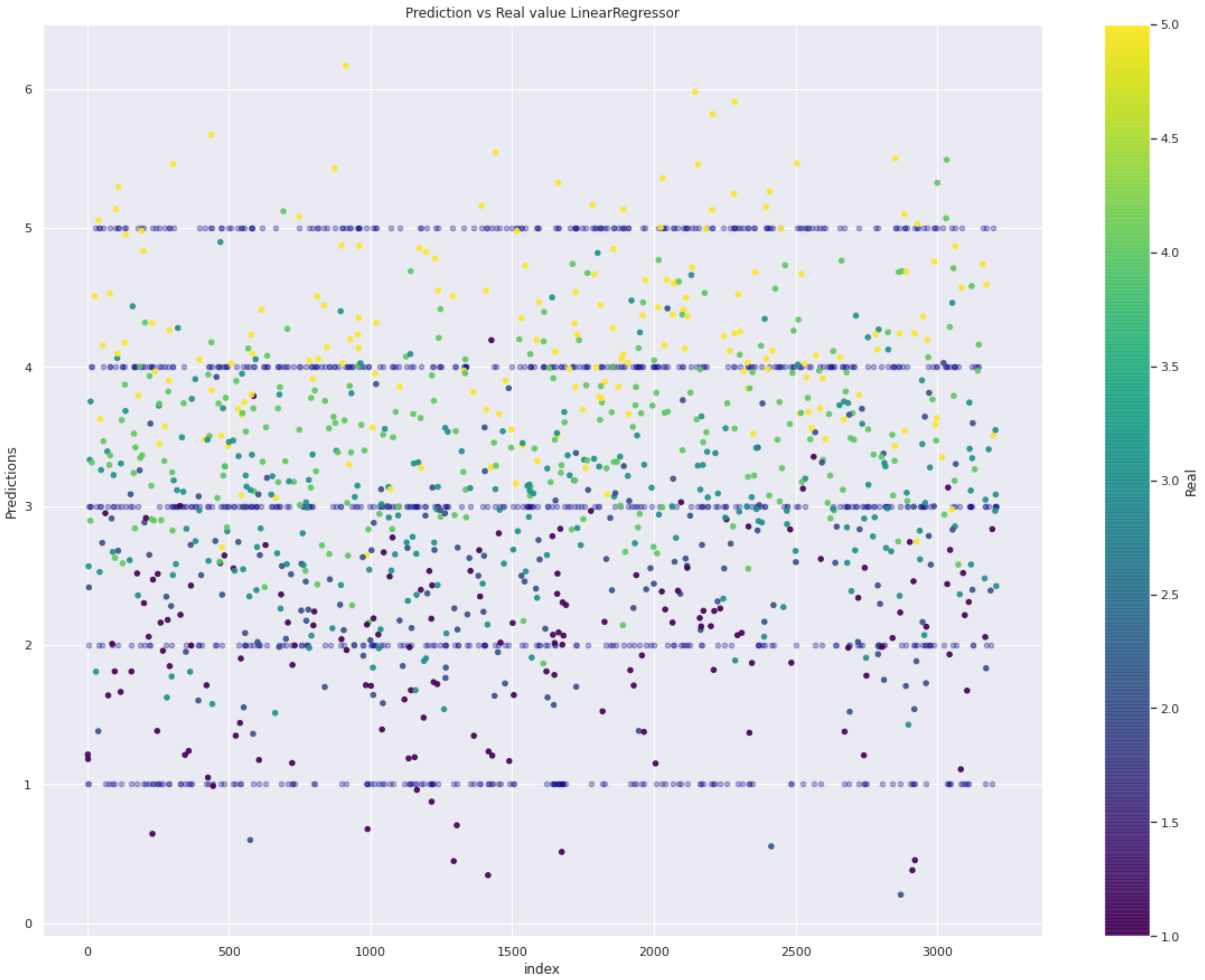

Fig. 4.1: Prediction vs Real value Linear Regression

Using this model we archieved a R2 test score of 0.56, and a MAE test score of 0.73.

Linear SVM

Second method we are using is the Linear SVM. The best hyperparamaters we found for this model are the following:

```
{'random_state': [42], 'max_iter': [100000], 'epsilon': [0.8], 'C': [25]}
```

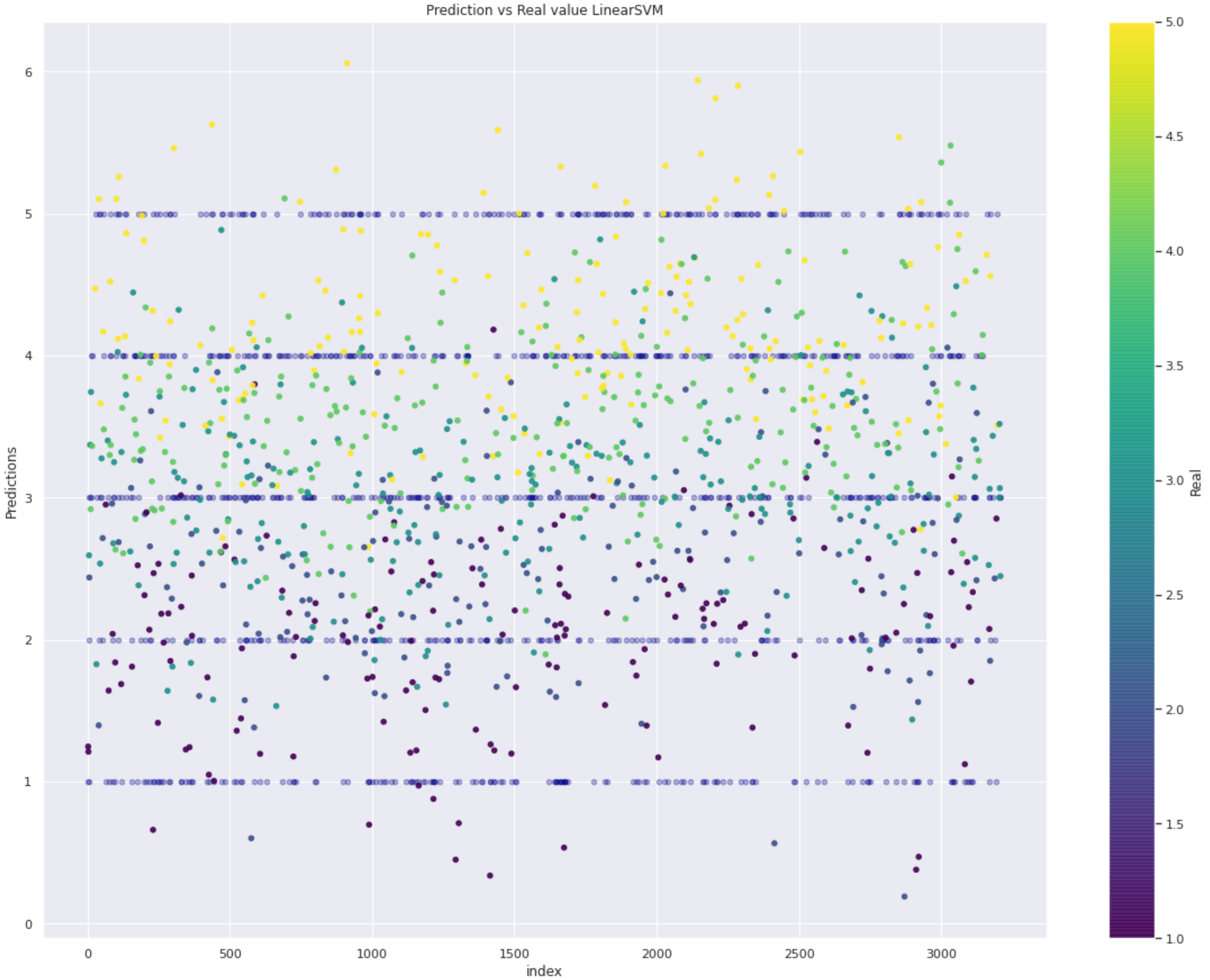

Fig. 4.2: Prediction vs Real value Linear SVM

Using the Linear SVM model with best hyperparameters, we archieved a R2 test score of 0.56, and a MAE test score of 0.73.

Quadratic SVM

Third method we are using is the Quadratic SVM. The best hyperparamaters we found for this model are the following:

```
{'kernel': ['poly'], 'epsilon': [1e-05], 'degree': [2], 'coef0': [50], 'C': [1]}
```

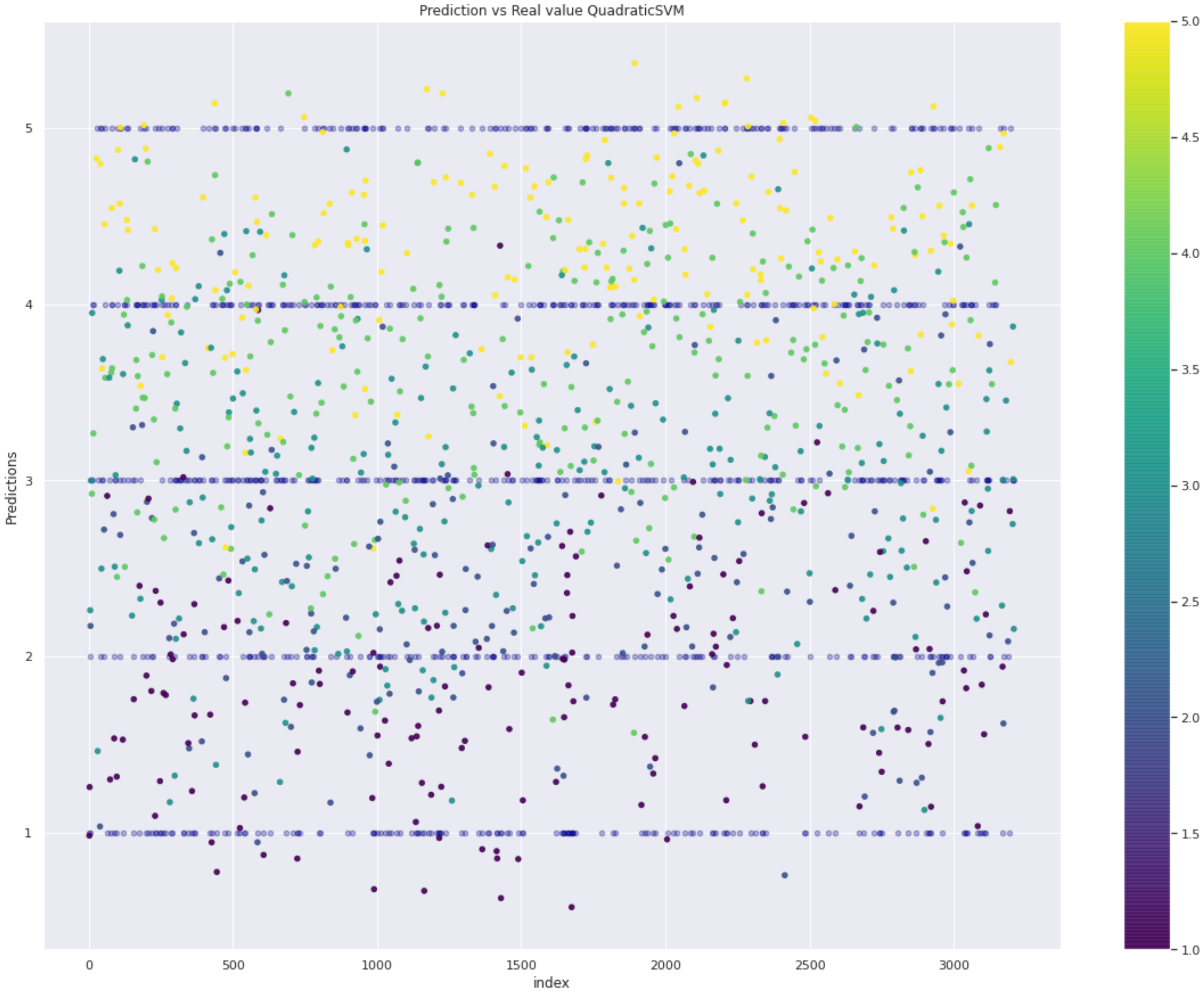

Fig. 4.3: Prediction vs Real value Quadratic SVM

Using the Quadratic SVM model with best hyperparameters, we archieved a R2 test score of 0.58, and a MAE test score of 0.69.

RBF SVM

First of the non-linear methods we are using is the SVM with RBF kernel. The best hyperparamaters we found for this model are the following:

```
{'kernel': ['rbf'], 'gamma': [0.001], 'epsilon': [0.0001], 'C': [1000]}
```

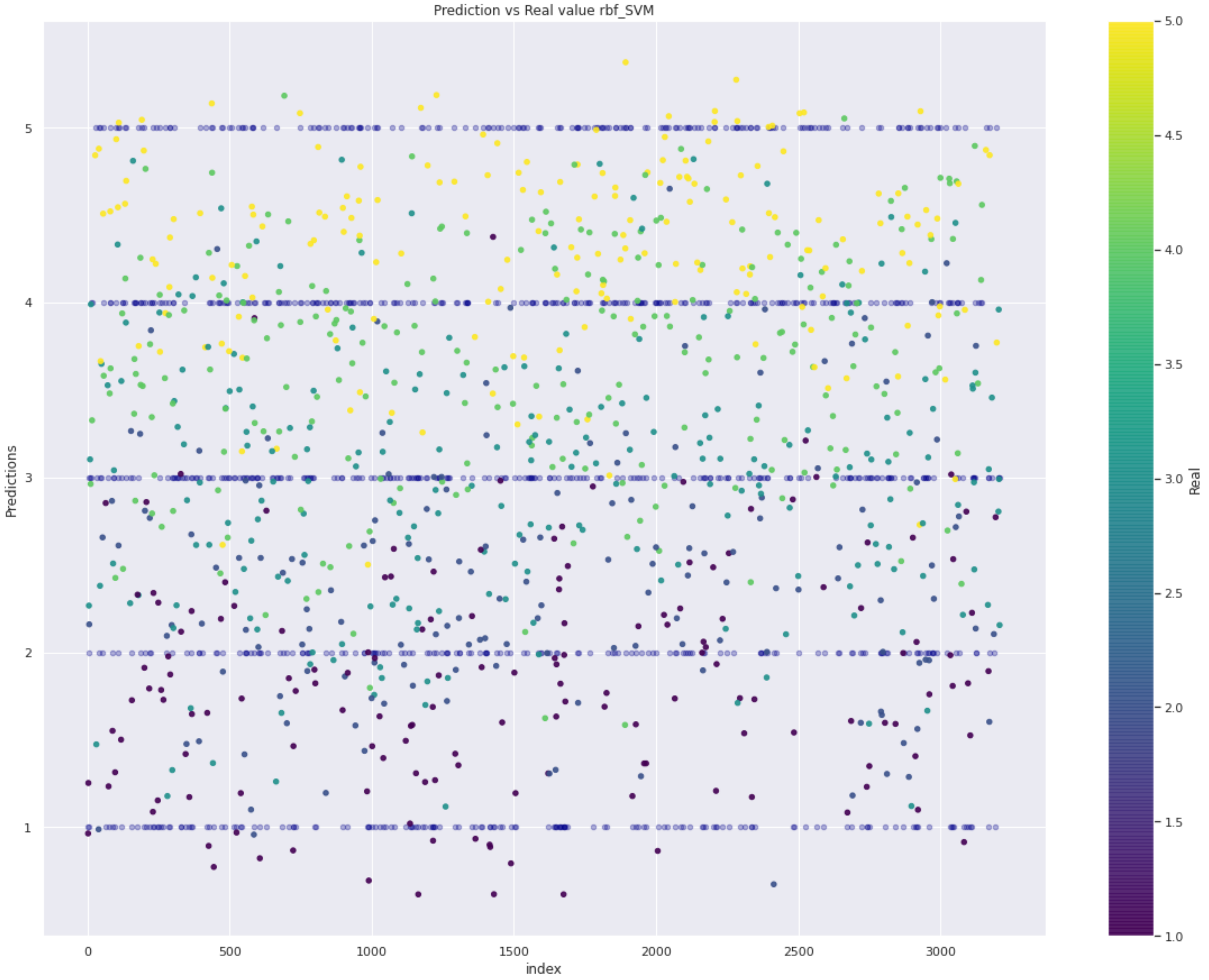

Fig. 4.4: Prediction vs Real value SVM with RBF kernel

Using the SVM with RBF kernel model with best hyperparameters, we archieved a R2 test score of 0.59, and a MAE test score of 0.68.

Random Forest

Last method we are using is the Random Forest. The best hyperparamaters we found for this model are the following:

```
{'random_state': [44], 'n_estimators': [225], 'max_features': ['log2'], 'max_depth': [23]}
```

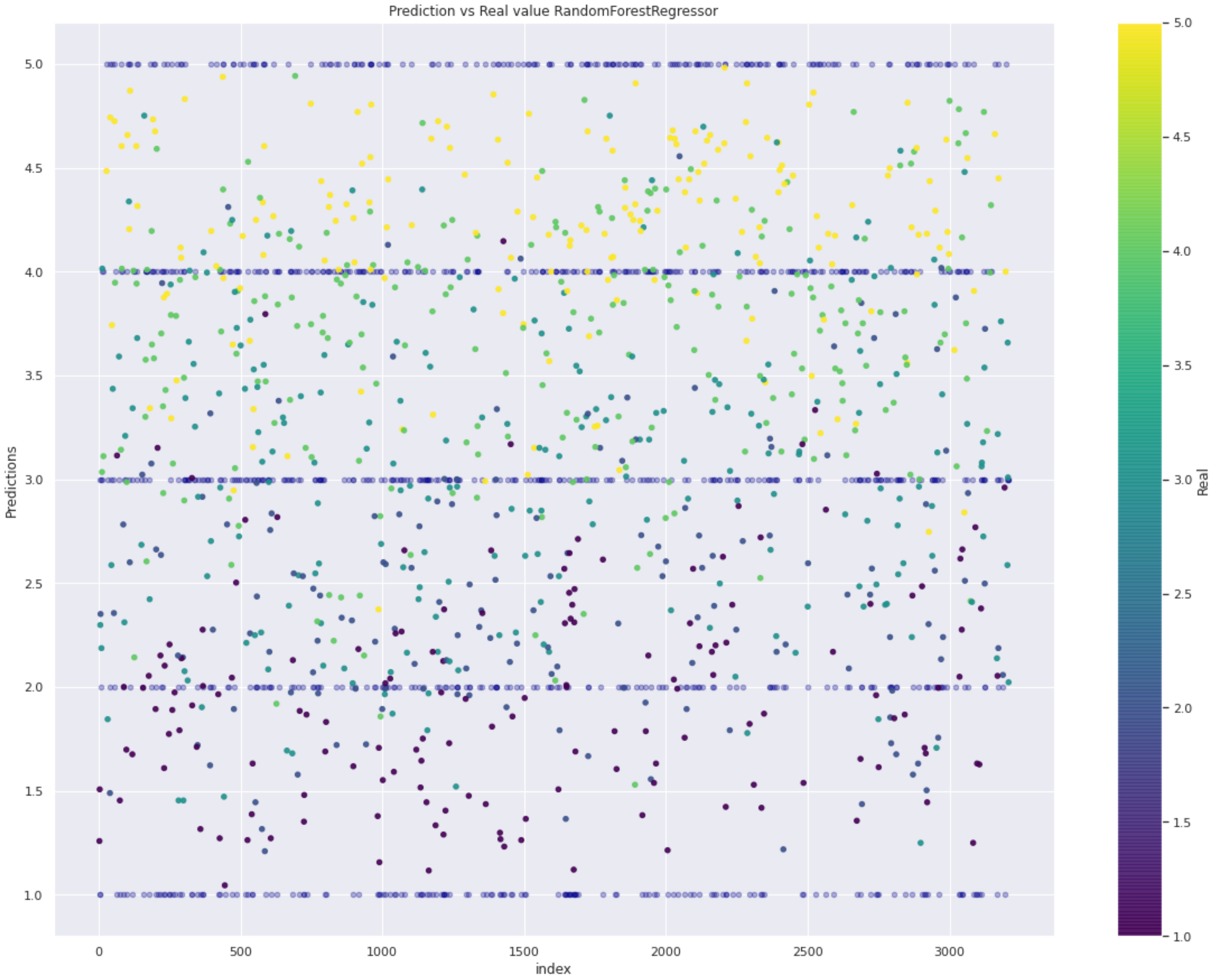

Fig. 4.5: Prediction vs Real value Random Forest

Using the SVM with RBF kernel model with best hyperparameters, we archieved a R2 test score of 0.58, and a MAE test score of 0.70.

5 Model Chosen

Analyzing the final results, we can see that all 5 models act pretty similar in terms of score, despite that, the SVM with RBF kernel produces the best score with the least absolute mean error. Although it offers the best results, by a low margin compared to the others, it also has the greatest prediction time which wouldn't be desirable in real-time applications given that one can get similar results with simpler models. As a final model, and following the Ockham's Razor principle, we are inclined to pick the simplest model which is the LinearRegression as it has an acceptable prediction score and is the fastest model regarding prediction time.

	Test score R2	Test score MAE	Train score MAE	Prediction Time
RbfSVM	0.587767	0.682977	-0.692864	1.352230
QuadraticSVM	0.582313	0.689182	-0.698299	0.567482
RandomForestRegressor	0.579276	0.702062	-0.709131	0.545400
LinearRegressor	0.559078	0.725275	-0.724747	0.001171
LinearSVM	0.558238	0.725961	-0.724471	0.002330

6 Interpretability Analysis

7 Self-Assessment

Looking back at the overall performance obtained with the models, the poor performance could be explained simply because the motor abilities of the players can't fully predict their corresponding rank in the league. But before arriving to such conclusions further studies could be performed, such as:

- Instead of performing Feature Selection one could analyze the performance doing Feature Extraction.
- Choosing different models like K-Nearest Neighbours or a Neural Network.
- Using the models provided by the libraries mord or statsmodels which allow Ordinal Regression in Python.
- Instead of training models with all the categories at once, training the models by pairing ranks may increase the separability between categories (As seen in the original paper of the dataset) and by doing so arrive to our own conclusions.

All in all, this project provided an enlightening experience by looking at extracted data from such a popular game as Starcraft 2 and see the different characteristics of the game that could be analyzed and experience first-hand the problems that one can encounter with data in the 'wild'.

8 References

[1] 1v1 league distribution. `https://www.rankedftw.com/stats/leagues/1v1/#v=2&r=-2&sx=a`. Data actualized daily.

[2] Ethem Alpaydin. *Introduction to machine learning.* 2020.

[3] Jason Brownlee. How to remove outliers for machine learning. `https://machinelearningmastery.com/how-to-use-statistics-to-identify-outliers-in-data`, 2018. Last Updated on August 18, 2020.

[4] Thompson JJ, Blair MR, Chen L, and Henrey AJ. Video game telemetry as a critical tool in the study of complex skill learning. *PLoS ONE*, 2013.

[5] Leslie Lamport. *LaTeX: a Document Preparation System.* Addison Wesley, Massachusetts, 2 edition, 1994.