Regression based machine learning on the FIFA Ultimate Team transfer market

Master of Science Thesis
University of Turku
Department of computing
Data analytics
2023
Jaakko Kittilä

UNIVERSITY OF TURKU
Department of computing

Jaakko Kittilä: Regression based machine learning on the FIFA Ultimate Team transfer market

Master of Science Thesis, 36 s.
Data analytics
August 2023

Ultimate Team is a highly popular game mode in the FIFA video game series developed by EA Sports. In Ultimate Team, players can buy and sell items based on real players on the transfer market with an in game currency. To combat buying and selling the currency with real money, the items have a price range set on the transfer market so that each item can only be sold for a reasonable price based on the item's abilities.

This thesis demonstrates how machine learning regression models could be used to predict the prices of new items entering the game, so that the price ranges could be set more accurately to improve player experience. As the predictions weren't good enough to improve the current situation, the thesis goes through what are biggest issues in making the predictions.

Asiasanat: regression, machine learning

Table of contents

Figures

Tables

1 Introduction, background and inspiration

1.1 The purpose of this thesis

Ultimate Team is a highly popular game mode in the FIFA video game -series produced by Electronic Arts. In Ultimate Team, players gather their teams from virtual player items based on real world players. This thesis aims to use machine learning to predict the prices of new players released to improve player experience and reduce chances of cheating in the game.

First, this thesis will explain the necessary information about Ultimate team regarding this thesis in more detail and its inspiration and after that in the second chapter, it will briefly cover what other research has been done in this area. In the third chapter it will cover what how the data for the thesis was gathered and stored and how missing values were handled and how it was decided what parts of it were filtered out. The fourth chapter will go deeper into the techniques for the predictions. Also, the setup, for example feature scaling and how the input and output values for each timestamp in the data were chosen are covered in this chapter. The fifth chapter will go through all of the features that were tested in making the predictions and which ones were chosen to be used in making the final predictions. The sixth chapter will go through the regression techniques that were used in making the predictions and their performance. Finally, the seventh chapter will cover what can be learned from the results of the regression models and the final results of thesis.

1.2 Ultimate Team explained

In Ultimate Team every player has their base item, which is meant to represent their real world performance as close as possible. The card has an overall rating of 1-99 and six stats, pace, shooting, passing, dribbling, defending, and physicality that are also ranked from 1-99. These stats also have more detailed substats, for example jumping in physicality. Player items also have 1-5 star ratings for their weak foot and skill moves. For goalkeepers, their items are rated similarly, but have separate stats, diving, handling, kicking, reflexes, speed, and positioning, due to the difference of play in their position.

On top of the base item, players can also get boosted items based on their real world performances. For example if a player scores a hat-trick in, he will most likely get into the Team Of The Week and get a boosted item released that will only be available for a week. On top of the boosted items based on the real world, there also weekly promotions, where there are players released with bigger boosts based around a theme, e.g. FUT captains that focuses on players that have been captains of their club for a long time. The clear majority of the new cards released throughout the year are boosted items, since players will only get a new base item if they move clubs but new boosted items will be released weekly.

These new items can be bought from the transfer market with the in-game currency called FIFA coins. All Ultimate Team players can list their existing player items to the transfer market where other players can then purchase them for prices ranging from a few hundred coins going all the way up to the maximum price of 15 million. Every game played earns around 300-500 coins and weekly rewards from different game modes earn about 5000 to 100000 coins and some amount of packs based on how well the player performed how much they played that game mode. In the transfer market each item has a minimum and maximum price set that they can be sold at, i.e. a price range. New items enter Ultimate Team through packs, that give a random set of items, with more highly rated players naturally being rarer. These packs can also be bought with FIFA coins or also with FIFA points that can only be bought with real money. New items can also be acquired by doing objectives or submitting player items to Squad Building Challenges (SBCs), i.e. sets that have requirements on what kind of items to submit to receive a new player item. However, these items from objectives and SBCs can't normally be sold on the transfer market so they important in the context of this thesis.

In Ultimate Team, there is a chemistry system in the teams, meaning that playing items alongside other items from the same nation and league will increase the items' chemistry score, which then boosts the items' stats. This means that players from leagues and nations with other highly rated players will be more valuable than players from lower rated leagues and nations. The chemistry system was changed from FIFA 22 to FIFA 23, but the basic premise that players from popular leagues and nations will be more expensive remains.

1.3 Inspiration

Football is the most popular sport in the world and the FIFA-series is the most popular football video game series. Football, in this thesis meaning the European version of the sport, where eleven players on both teams try to get the ball to the opposing team's goal. Across the FIFA, Madden and NHL series, Ultimate Team game modes generated around 1,6 billion US dollars in the year 2021, so Ultimate Team is a huge business nowadays, especially on FIFA [1]. Ultimate Team being a hugely profitable business model and a highly popular game mode means that people are willing to do illegitimate things as well to gain an edge over other players.

As mentioned in the previous section, each item has a price range set for it. Before price ranges were used, people could buy FIFA coins in exchange for real money for prices that were significantly cheaper than buying FIFA points and opening packs, which relies on the player getting a highly valued item from the pack and then selling it, which isn't at all guaranteed. FIFA coins were usually sold through the buyer listing any normally cheap item for an absurdly high price that the seller would then buy on their account, thus transferring the coins to the buyer. With coins being this easy to acquire, the prices of the best player items were inflated beyond the reach of players, that weren't willing to spend real money on the game.

To prevent this way of transferring coins, price ranges were set so that players could only be sold at a reasonable price relative to their rating. However, a poorly set price range brings in a new problem, if the minimum price is set too high, the item is difficult to sell, as other people aren't willing to spend that amount of coins on it, or if the maximum price is set too low, the item is difficult buy as people aren't willing to list the item at a price that feels too low and all the ones that are listed are instantly sold. This thesis aims to predict the prices of new players released, so that their price ranges could be set more accurately so that the players' experience on the transfer market could improve and reduce the chances of cheating further, with player items not having too large of a price range.

2 Similar research

I wasn't able to find similar studies to this, most likely due to the data being quite hard to collect and the Futbin terms and conditions forbidding usage of their pricing data with an exemption for students, but there are still some other studies worth mentioning in this context, either using the same data for different purposes or doing similar research in another environment.

2.1 Using FUT data for other purposes

Due to FIFA Ultimate Team having a large database of real world players with stats try to mimic their footballing abilities, the data is also possible to be used outside of the game. For this reason, some research has been made using the same kind of data as used in this thesis, but instead for predicting the real world value for the players. For example, a recent study using data from an older FIFA, FIFA 20 had promising results using random forest regression to predict the estimates of player values if they were to be sold to a new club. Naturally, real world values are determined differently from values in game, as for example player reputation can generate higher commercial income and a potential young player is more expensive, as they will likely get better and can have a longer career at their new club, unlike in the game where only the current ability is relevant and not all real world abilities can be directly translated into the game. For this reason, features like player potential and reputation were used in this research, but parts of the data were still similar.

Also, in the conclusion of this study, it was mentioned that this study and its results could be used as a basis for predicting the in game values of players as well, which this thesis is attempting to do. This study built on top of previous studies, which had used linear models in predicting the player values, but due to the non-linear relationship of some of the features, the random forest approach outperformed them, which can also be seen in the results of this thesis. [2]

Some research has also been done a few years ago using the players' individual in-game abilities in predicting the results of real world matches with good results. [3] [4] However, if the players in the game are meant to represent how they perform in real life, it's reasonable to expect that good teams in the game will also perform well in actual games.

2.2 Other FUT related studies

Since FUT is among the most popular and highest grossing game modes in gaming, it is also a good place to get an insight into player behaviour. These studies aren't necessarily directly related to the subject of this thesis, but they help provide a better overview into the game as a whole and maybe can also explain the some of the phenomena that affect the results of this thesis.

One popular field of study in Fifa Ultimate Team is its microtransactions, which account for most of its income. Loot boxes (packs in FUT) have been a hot topic even in political conversations recently, due to them basically being gambling, but right now they still aren't as restricted as gambling in a legal context in most countries. An interesting perspective into this was acquired from message threads gathered from the EA FIFA forums, where analyzing players' opinions found that animosity towards the producers and developers of the game didn't prevent players from spending money on it. Also, the players who are most profitable to EA, either from buying packs or promoting pack sales to other players or playing the game professionally, tend to also be more likely breaking the rules regarding selling coins or entire accounts with good teams on them, which spawns all kinds of conspiracies among the other players that these players are given preferential treatment by EA. [5] Based on this study, another study looked deeper into the relationships between the players, producers and also streamers and Youtubers who produce content on the game. They found out that the players on the forums tend to be frustrated towards the contect creators and shift some of the blame from the producers and developers to them. The players see that the content creators who have large audiences use broken game mechanics in their content that gives them an advantage in the game and also promote them to other players, which can lower the gameplay experience as a whole. Also, them promoting buying packs to other players, especially young players is perceived negatively and enforces the opinions that spending money is required if you wish to succeed in the game. [6]

Obviously not all players feel negatively towards streamers, it is also reasonable to expect that players who are willing to spend time writing on forums regarding the game are more invested into the community and have a better overall image of the state of the game as a whole, which may lead to more negative opinions towards EA and content creators. Still, this shows that around FUT there is an intricate set of different parties who don't necessarily always get along with each other but still care for the game itself.

Another study took the research into the mechanisms that make a player purchase packs further and also studied how closely buying packs in the game is related to gambling and internet gaming disorders through a survey filled out to by approximately 1100 FUT players. The study found that even though spending money on the game can help give the player access better items faster, the time spent on the game was a better predictor of success than how much money the player spent, which still had some success in predicting success. The relatedness of buying packs to gambling could be seen in reward sensitivity having a significant role in predicting how much a player spends real money on packs, i.e. getting a rare item from a pack gives the same instant gratification as winning a bet and will lead to the player attempting to replicate the feeling buy spending more. Also, compared to traditional gambling where a losing bet will lead to the player losing everything they spent, in FUT a pack will always something, which can lead to the loss of money feel less significant. This was also seen looking at buying packs from the perspective of Self Determination Theory (SDT), where the motivations for purchasing packs where observed through competence, autonomy and relatedness. In FUT, autonomy was the biggest reason for spending money on packs, meaning that buying packs enabled players to build the teams they wanted, either through placing the rare items they packed into their team or selling the less valuable items. [7]

3 Gathering data, data format and first data filtering

3.1 Gathering data

The data for this thesis was gathered from the website futbin.com. Futbin is a website that focuses on maintaining a database for all the player items in Ultimate Team and also their current prices as well as a daily price graph for all the items throughout their existence.[8] The current FIFA out now is FIFA 23 but for this thesis the players and prices for FIFA 22 were used, since that allowed for more data to be used and testing the performance of the predictions throughout the whole year. In FIFA 23 the Playstation and Xbox platforms have a shared transfer market, but in FIFA 22 they are still separate, so the prices from the Playstation platform were used for the predictions.

For parsing the player data from Futbin, an open-source Github project was used as the basis for gathering all of the required data. There wasn't any way to get the player item data directly through an API so the player data had to be parsed from the player table HTML based on row identifiers and table cell titles. The project that was used initially allowed the player table HTML to be parsed and added to the data. With small additions to the code it could also gather unique item identifiers that allowed more detailed player data to be gathered from the individual player item pages. With this ID player pricing data could also be gathered. The pricing data was available as Javascript Object Notation (JSON) [9] data through an API.

3.2 Data files and format

The data that was gathered was split into two files. One that included all of the player items and all of the features that related to them. The original amount of players gathered was around 23000 so the first stage of data filtering was done at this point to reduce the time and storage space required for getting the more detailed item data, as well as the daily pricing data for all the items. The player prices were stored into a second file, where each row contained the price, the timestamp and the player ID that the price was related to.

Both files were stored in the Comma Separated Values (CSV) format.[10] In CSV the data does not contain any extra formatting or styles, it just has column values separated by commas. This allowed the data to be easily opened and manipulated and many programming languages have ready-made libraries that make these tasks easy. The simple structure also made manual editing possible, which came in useful many times.

3.3 Data filtering

To make the next stages of gathering and fixing missing values less time-consuming, the filtering out items was done at this point. Firstly, all bronze items (rating <= 64) were dropped from the data, as their prices didn't go over 10 000 coins at all and there were so few bronze items released during the year, that they would just add noise in the data. After dropping the bronze items, all Squad Building Challenge and Objective items were dropped as well, as they aren't able to be bought or sold through the transfer market and weren't useful for this thesis. After this filtering there was 14400 items left in the data, with 10184 silver (rating <= 74) and 4216 gold items (rating >= 75), where 2105 gold items and 1913 were released after the release of the game, i.e. the items whose prices would be predicted. To decide whether or not to keep silver items in the data, boxplots of the average prices of all gold and silver items were compared.

The boxplots revealed that the scales of the prices of gold and silver items significantly different, with the most expensive silver items being over 20 000 coins and the most expensive gold items being worth millions of coins. Due to this difference in scales it was decided to also filter out the silver items from the data. Even with just the 4000 gold items, the prices are heavily skewed to the smaller values and adding 10000 more items that are exclusively in the lowest prices would make it worse, while adding a whole lot of extra noise to the data. Also, even though the accuracy of the predictions could increase from adding in all the silver items, it wouldn't be beneficial for the purpose of this thesis, as it would likely make it harder for the models to evaluate the prices of the most expensive gold players and predicting prices between 0 and 20000 coins for all players wouldn't be useful, even though the performance would seem high.

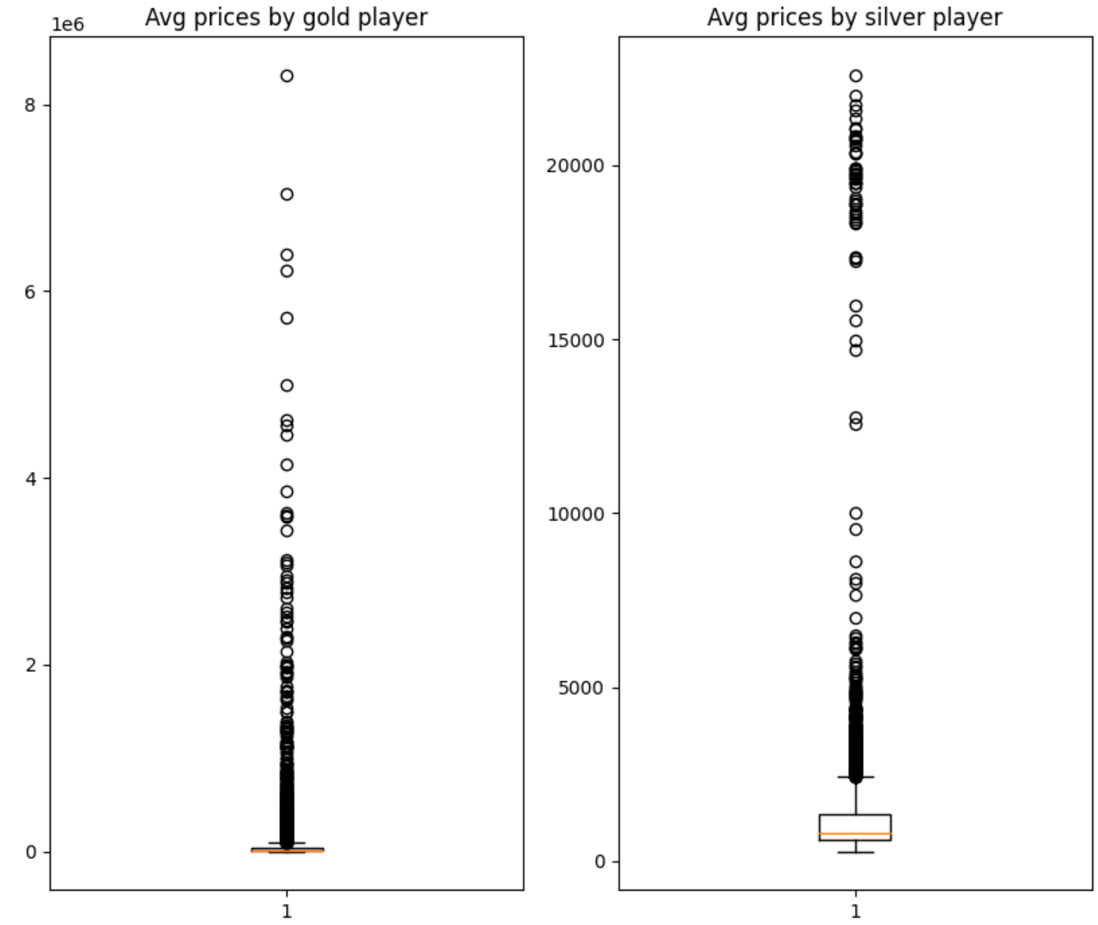

Figure 3.1: Boxplots of average prices of all gold players (left) and all silver players (right)

3.4 Missing values and fixes to data

There were some missing values in the player data, especially in the height, weight and body type features, that had to be gathered from the same table cell with regular expressions. Weight and body type weren't deemed to be important features to start with, so these features were dropped because of the large amount of missing values and expected small amount of use in the predictions. The missing height values were filled in from various sources, as height could potentially be a useful feature.

If an item wasn't consistently available to buy in a particular day of price data, its' price had a value of 0 for that day. If the items that had a price of 0 were in the training data, they were omitted from that training to stop missing values from interfering with the training. If an item had a missing value in the group of items whose prices are to be predicted, the next price available in the price data, was used for that item's true price.

Some player items, mostly icons originally had a price on the date of the game's release then followed by a few months of 0 values in their price data. These prices and the missing values that followed them were manually removed from the data as well.

3.5 Live items

Some of the player items in the data are live items, which means that they could be receive further boosts to their stats after their release. For example Ones To Watch items received all the performance based boosts that their player received and Champions League items received boosts if their player's team progressed to the next stage in the competition. This meant that these items were in their final version in the data gathered and not in the version that they were released at. To keep the data realistic, these players had to be changed to their original form in the data. This was done manually, as their original versions had to be sourced from a few different places. To further enhance the data, all the boosts and their timestamps were added to a third CSV file. Then, depending on the timestamp of the data, the original versions of the items were dropped from the data and the most recently boosted versions were put in their place.

4 Approach to making predictions and features used

4.1 Techniques used

As the thesis is about predicting continuous, numerical values, it is a regression problem, i.e. giving the regression model a number of independent, numerical features as input X and an output value of Y, the model then tries to generalize the relationship between the input and output values so that it could be used to predict values on items outside the training dataset.

Python is a popular programming language in data and machine learning related fields and it was also used in this thesis. It had ready-to-use libraries such as Pandas and Sklearn that made it easy to read and manipulate the data and test multiple different machine learning models without needing to do extra work with every new model.

4.2 Performance metrics

4.2.1 Regression performance metrics

The performance metrics used for evaluating the accuracy of different machine learning models in this thesis are the **Symmetric Mean Absolute Percentage Error**

(SMAPE), which is defined as

$$\mathrm{SMAPE}(y,\hat{y}) = \frac{100\%}{N}\sum_{i=0}^{N-1}\frac{2*|y_i-\hat{y}_i|}{|y|+|\hat{y}|}$$

and the **Mean Absolute Percentage Error**, which is calculated

$$\mathrm{MAPE}(y,\hat{y}) = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i-\hat{y}_i}{y_i}\right|$$

where y is the predicted value and $\hat{y}$ is the actual value[11] [12]. Percentage errors were chosen, as the prices range from almost 300 to 15000000 so just a mean error would be hard to interpret in the context of this study. For MAPE, the error values go from 0 to 100 for when the model predicts below the true value, and 0 to infinity when it overpredicts. For SMAPE, the error values go from 0 to 200 regardless of over or underprediction.

These two values were chosen to get a wider understanding of the performance in this study, as both have their own weaknesses in this context. As we find out as we test different models, all of them tend to overpredict, which results in poor MAPE scores. Also, due to the large scale of the prices we're predicting, errors in smaller priced items are going to have a larger prevalence in determining the MAPE performance. For example, predicting 4000 for an item with the price of 1000 will result in MAPE of 300, but when we consider the range being 300-15000000, the prediction is quite acceptable. SMAPE mitigates these issues that MAPE has, but on the other hand it gives smaller errors to underpredictions, so it's not symmetric either.

4.2.2 Other performance metrics

As we later find out in this thesis, the regression model performances weren't ideal, so other performance metrics were also tested to see that even though the regression performance wasn't what was hoped, the models still managed to learn something from the data.

The first metric chosen was the **Somers' D metric**, which is defined as

$$D = \frac{\text{Number of Concordant Pairs} - \text{Number of Discordant Pairs}}{\text{Total Number of Pairs}}$$

Rather than comparing the numeric output values to the true values, Somers' D is an ordinal metric, meaning that when both the predicted and true values are put into order, it measures how well the orderings match. Somers' D values are between -1 and 1, -1 meaning that none of the items are in the correct order and 1 meaning, that everything matches. [13]

The second metric tested was the **coefficient of determination** or R^2, which is defined as

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

In R^2 the metric measures how well the model does compared to just predicting the average for each item in the data. In R^2 value 1 means that there is no error in the predictions, 0 means that the model has equal performance to predicting the average and values below 0 mean that it actually performs worse. [14]

4.3 Setup for making predictions

After the filtering, the final dataset had 4215 items. As mentioned in chapter 2, the pricing data has item IDs alongside the price and the timestamp of said price. From this data, the release date of all items could be sourced by choosing the lowest value timestamp available in each items pricing data. In total there were 127 different release dates in the data, with 2111 items being released on the first day and the 2105 other items being released throughout the year.

With the release dates sourced, it was possible to divide the data into training and prediction data based on the timestamp, with items released before the timestamp being the training data and the items released on the timestamp being predicted.

The predictions were then in practice done by using the 2111 items released on the first day as the initial training data and then iterating through all the other release dates and adding more items to the training data as the iteration moves on during the year, re-training the model on each individual timestamp on the current amount of training data and their prices on that particular timestamp.

Due to goalkeepers having different stats on their items, the data was finally split into two parts, one having just goalkeepers and the other having all of the outfield players.

For the Y values for fitting the models, the average of the items price on the timestamp and the day before the timestamp were used. The value the model is trying to predict is the average of the item's price nine and ten days after its' release. That was chosen as most new items are available in packs for seven days, so after ten days, its' price should be settled down as no new versions of the item are no longer entering the game and there most likely won't be any new spikes in its' demand. Averages were used to reduce random variation in the data.

4.4 Feature scaling

As the models are fitted with numerical variables with different scales, such as rating [74, 99] and skill moves [1, 5], it's wise to re-scale all the numerical variables to the same scale, so that the variables with larger variance don't overpower the models. For this reason all numerical variables were centered and scaled by removing the feature mean and then dividing it by its standard deviation. This scaling was done at every timestamp individually before splitting the data to training and testing data to account for the current state of all the items released and their stats. It is not always a good practice to do the scaling before splitting the data, as it can lead to the training data leaking to the testing data. However, here where there are not that many items released at once and their distribution most often isn't the same as in the training data, scaling them individually would significantly reduce the performance.

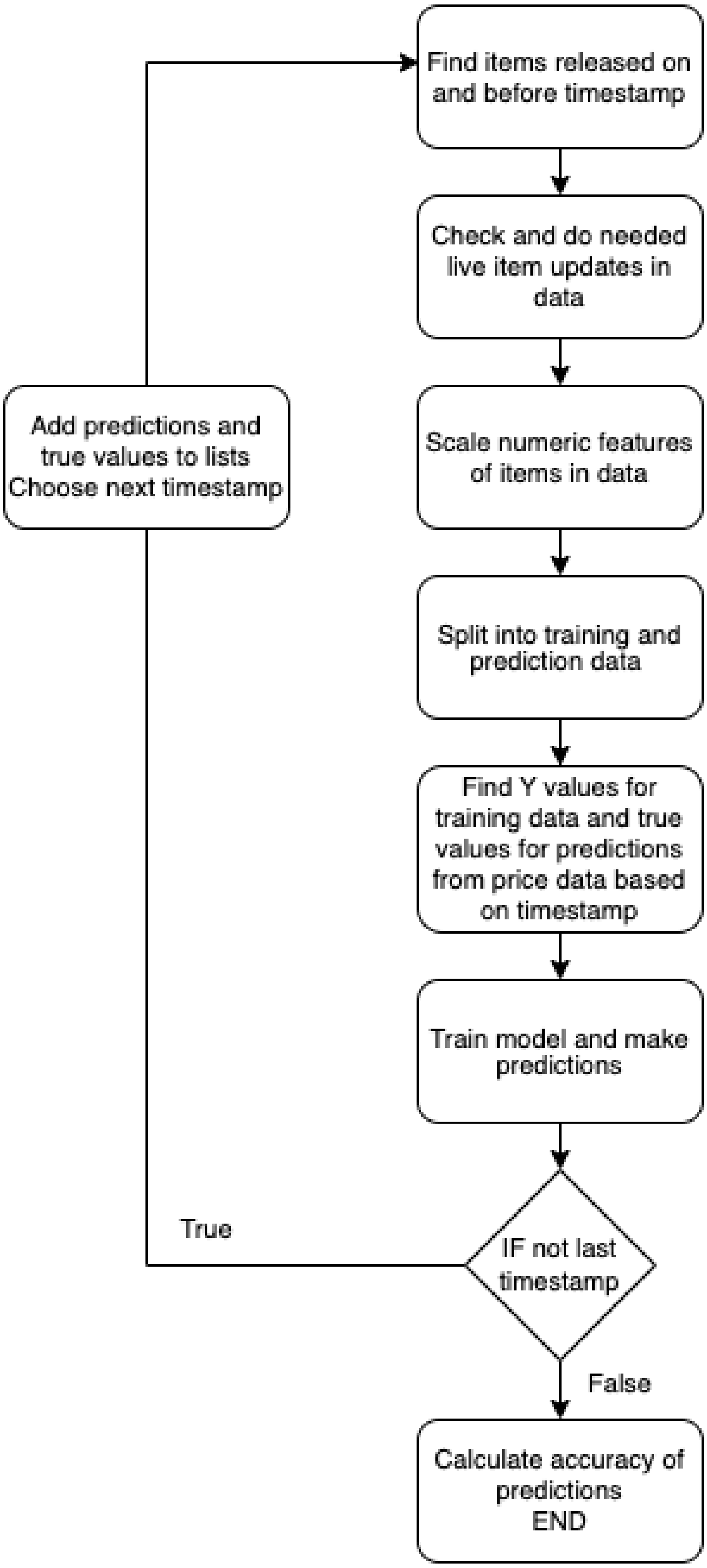
Find items released on and before timestamp
Check and do needed live item updates in data
Scale numeric features of items in data
Split into training and prediction data
Find Y values for training data and true values for predictions from price data based on timestamp
Train model and make predictions
IF not last timestamp
True
False
Add predictions and true values to lists Choose next timestamp
Calculate accuracy of predictions END

Figure 4.1: Flowchart of how predicting the values of all new items was done.

4.5 Prediction minimum and maximum values

As regression models try to fit their outputs to the given input features, using the models on items outside the training dataset can make the model give outputs that aren't sensical. In Ultimate Team, each item can be quicksold, meaning that the item is permanently removed from the game and the player immediately receives an amount of coins. The items can't be sold on the transfer market for less than their quick sell value, which gives each item type an absolute minimum price, which their minimum price in their price range can then increase. For non-rare base items, their minimum prediction value was set to 350, for rare base items their minimum prediction was set to 650, and special items their minimum prediction was set to 10000. A maximum price of 15000000 was set for all predictions.

If a player item had a lower rated version in the dataset and the predicted price of the new, higher rated item was lower than the current price of the lower rated item, the lower rated item's price was used, as it is reasonable to expect that a better version of the same item will be more expensive than the previous version.

5 Features tested and used

5.1 Performance based features

How well an item plays in the game is naturally very important on determining how it should be valued. These features try to capture that as well as possible:

Rating: The overall rating an item is given for its' performance is naturally taken to the predictions as its' the quickest way of telling how the item performs in-game. However, an item's price still doesn't linearly grow as its' rating grows and an item with a much lower rating can still have a higher price than a higher rated item. Still, due to the average rating requirements in squad building challenges, an item's rating gives it a minimum price for that particular rating, as players try to buy the cheapest cards for completing the squad building challenges, while still fulfilling the rating requirements.

Pace: How fast an item is, is arguably the most important factor for most players on whether or not to add to add it to their team. Typically, a pace of at least 85 is considered adequate for items in wide and attacking positions and 75 for items in central and defensive positions.

Adjusted pace: As some positions typically have items with more pace, the same amount of pace is valued differently at different positions. To account for this, items in central positions (CAM, CM, CDM, CB) used their original pace stat and items in wide and attacking positions (ST, CF, RF, LF, RW, LW, RM, LM, RWB, LWB, RB, LB) had their pace stat decreased by five.

Dribbling: How well the player can move with the ball. Even though dribbling isn't as important on defenders as they don't move that much with the ball, dribbling also covers agility and balance, which are deemed important in all positions.

Physicality: Physical features of an item, e.g. their strength, stamina and jumping.

Position stat: To reduce dimensionality and noise in the data, not all six stats from every item were used in the models. On top of pace, dribbling and physicality for all items, the final stat used was chosen based on the item's position. As the position stat, defenders used defending, midfielders used passing, and attackers used shooting.

Weak foot: Each item has a preferred foot specified. Weak foot indicates how well the item can control the ball on their weaker foot. Especially on attackers, the ability to shoot the ball on both feet is considered very important.

Skill moves: The higher the skill moves rating, the more advanced tricks an item can do and get around opponents easier.

Stars: As the skill moves rating isn't as important on defensive positions and as an attempt to reduce dimensionality, the stars feature was engineered. For stars, defensive items used just their weak foot rating, and midfielders and attackers used an average of their weak foot and skill moves ratings.

5.2 Chemistry based features

Due to the chemistry system, even if an item's stats suggest it should play well in the game, if the item is from a league and a nation that aren't easy to link to other good items, that item's price will be lowered significantly.

The mean and median prices for all items in the different leagues in the data were checked and icons stood out as by far the most expensive league. After that came Premier League (England), Ligue 1 (France), LaLiga (Spain), Bundesliga (Germany) and Serie A (Italy) and the rest of the leagues came behind.

The mean and median prices for nations were checked as well, and similar features for them were tested as well, but because there are even more nations and leagues in the data, the results weren't as useful as for leagues. Some nations had only one or two players and if one them were highly priced, that gave a false impression of how valuable that nation is.

League icon: If an item is an icon item. Icons are among the highest rated items in the game, along with the added benefit that they get chemistry with all other items.

League top 5: If an item is from the top 5 leagues (England, Germany, Spain, Italy, France).

League other: If an item isn't an item or from the top 5 leagues.

League tier: As an attempt to reduce the three different league features into one and create a larger gap between icons, top 5 leagues and other leagues, league tier was tested. As league tier, icons got 2, top 5 got 0 and others got -2.

League median: As an attempt to get more separation for each individual league in the data without adding more features, a median price of all items from that league was used.

League amount: Same logic as in league median, but instead of the median price, the amount of players in the data from each league was used.

Nation median and **Nation amount:** Same as league amount and league median, but for the nations of each item.

5.3 Others

Pos def: If the item is a defender.

Pos mid: If the item is a midfielder.

Pos att: If the item is an attacker.

Updateable: For live items, their price isn't only determined by their current rating and stats, but by also the fact that they may get boosted in the future, which raises their price above other similar items that can't get boosted. To reduce the chance of the price of live items affecting other items, the updateable feature was created. For each timestamp in the data, each live item was checked and if they can still get boosted, they will get 1 for updateable, otherwise 0. The problem with this is that there aren't that many items in the data at the same time that can get boosted. On top of that, not every live item is as likely to get boosted and can get boosted as much and so the amount of price increase depends on how players evaluate, that how likely each item can get boosted.

Special: To try to get more accuracy in the lower end of items, special was used to separate the base items from boosted items. Even though cheaper boosted items and base items have similar stats, due to the higher quick sell value of boosted items, they are a little more expensive.

5.4 Goalkeepers

As goalkeepers have different stats from outfield players, their own stats (**diving, handling, kicking, reflexes, speed,** and **positioning**) were tested, as well as their **height**.

5.5 Features used

From all of the features tested, the following gave the best performance for outfield players, when used together: **Rating, Adjusted pace, Dribbling, Physicality, Position stat, League icon, League top 5, League Other, Pos Att, Pos Mid,** and **Pos def**. For goalkeepers, the separate features tested just for them weren't found to improve performance, so just **Rating, League icon, League top 5,** and **League other** were used.

6 Models tested and model performances

6.1 Models chosen

For the regression models used, models of different complexities were chosen to be tested. Here are the models tested, organised by their type.

6.1.1 Neighbors regression

Regression based on the nearest neighbors is one of the easiest places to start, when starting to implement a regression model. When predicting based on the nearest neighbors, the model is given the training features X and output values Y and when given new data to predict on, the model calculates the items from the input data that are the closest to the new data and uses the average of their output values as the prediction. Due to this nature neighbors regression is efficient with non-linear data, but doesn't do well with items that are outside the boundaries of the training data. [15]

As neighbors models the following two were tested:

k-nearest neighbors: In k-nearest neighbors the model is given a value k for how many nearest neighbors it calculates from the data to use for predicting.

Radius nearest neighbors: In radius nearest neighbors, the model isn't given a fixed amount of neighbors to calculate from the data, but is instead given a distance, and each item in the training data that is within that distance of the item to predict on, is taken into calculating the prediction.

6.1.2 Linear regression

In **Linear regression**, each of the features in input X are given a coefficient and the coefficients are set so to minimize the least squares error in the output Y, i.e. the model is trying to fit a linear equation to the input. In this thesis, there are quite a few features in the input data, and that can lead to the linear regression model creating overly complicated coefficients and overfitting to the training data. [16] For this reason **Ridge regression** was also chosen to be tested. Ridge regression follows the same premise as linear regression but adds a regularization parameter alpha, that penalizes too complex models. How large alpha is set, then controls the overfitting, but too high values will lead to the model underfitting. [17]

6.1.3 Decision tree:

Decision tree regression is easy to setup for most regression situations, as is doesn't require variable scaling or creating dummy variables, as it doesn't do calculations on the variables on the input features X, but instead tries to find different tests to perform on the data, that have the most accurate effect on predicting the output variable Y. It's called a tree as it starts by selecting the feature that provides the biggest reduction in variance, and then splits the data into two subsets based on its values, called leaves. This is then repeated for every subset until an implicitly set max depth of the tree, or a minimum amount of data in a leaf is met or there are no more subsets to split. Then predictions can be made by traversing the tree from top to bottom, choosing the correct leaf based on the comparison set by the model. [18]

6.1.4 Ensemble methods

Ensemble methods are machine learning methods that use multiple algorithms to obtain a better accuracy, than using just one algorithm. There are many different ways of going about ensemble learning, e.g. which group of algorithms to use, but for this thesis the following two were chosen.

Random forest: In random forest regression, multiple decision trees are trained with different samples of the training data, and then an average of the predictions of the different trees is used as the final prediction. Doing it this way and dividing the data between different trees is a good way to control overfitting. [19]

Gradient boosting: Gradient boosting is also based on training multiple decision trees, but instead of training a predetermined amount of decision trees, the trees are built additively, meaning that new trees are trained to improve on the weaknesses of previous trees. [20]

6.1.5 Neural networks

Neural networks are meant to mimic the way human brains learn and process information. For neural networks, **Multilayer perceptron** (MLP) was chosen for testing. MLPs consist of connected perceptrons, which are organised into at least three layers, with one input, one output and one hidden layer, the multilayer approach making it possible for MLP to fit to non-linear data as well. The perceptron connections are weighted, meaning that some inputs can be given higher values than others. The perceptrons each have an activation function, meaning that if the inputs exceed some value, the perceptron fires. The training of the MLP then comes down to calculating the error of the initial predictions and then readjusting the weights of the connections, to improve the performance. This can be repeated as many times as it's necessary. [21] [22]

6.2 Model performances

6.2.1 Regression model performances

The models tested received the following SMAPE and MAPE scores for their total accuracy. To get a better understanding of the strengths of each model, SMAPE scores for every quartile based on their true price were calculated as well, i.e. the first quartile (Q1) is the bottom 25 percent of items based on their true value and Q2 is bottom 25 to 50 percent of items and so forth.

Model	MAPE	SMAPE	Q1	Q2	Q3	Q4
k-nearest neighbors	116.249	51.616	38.249	40.597	70.547	57.096
Radius nearest neighbors	225.219	79.265	66.064	77.922	109.478	63.622
Linear regression	556.908	109.068	106.813	129.435	133.303	66.723
Ridge regression	560.743	109.818	106.923	130.746	134.921	66.687
Decision tree	171.193	57.795	39.833	43.134	75.063	73.185
Random forest	333.656	58.163	47.426	40.265	81.417	63.265
Gradient boosting	199.634	67.181	74.413	43.118	86.309	64.871
Multilayer perceptron	145.321	63.267	36.764	43.387	103.392	69.575

Table 6.1: Table of models chosen for testing and their SMAPE scores

6.2.2 Other metric performances

The models tested received the following Somers' D and R^2 scores.

Model	Somers' D	R^2
k-nearest neighbors	0.764	0.604
Radius nearest neighbors	0.749	0.242
Linear regression	0.703	0.170
Ridge regression	0.706	0.169
Decision tree	0.745	0.383
Random forest	0.764	0.616
Gradient boosting	0.704	0.445
Multilayer perceptron	0.691	0.438

Table 6.2: Table of tested models and their Somers' D and R^2 values

7 Prediction results

7.1 Results overview

As can be seen from the table of different model performances, there is some potential in predicting item prices with this data, however the results in this thesis fell short of improving the current situation. Most of the models could at least differentiate between the lower and priced items and observing individual predictions, had some sense in how they could come to their particular values.

As it was anticipated even before testing, linear models had the worst performances of the models tested. For outfield players, there were 11 features selected, which is quite a lot for fitting coefficients for each of the features. Also, the item prices don't increase linearly, but instead the same stat boosts bring much higher price increases in higher rated items, than low rated items. This meant that to minimize the error in their predictions, linear models tended to predict quite similar prices for all items, as can be seen in Table 6.2.

K-nearest neighbors had both the best SMAPE and MAPE performance, but with other models there were quite big variances between how they well performed in the two different regression metrics. Multilayer perceptron's predictions averaged lower than neighbor models, which helps explain why it performed so well in MAPE, as it's penalty for overpredicting doesn't have a limit. Another interesting model is random forest, which performs well in all other metrics, but has a MAPE of 333, but once again looking at table 7.2 gives this context, as it tends to predict higher than other models.

The R^2 from the different models are all above 0 and range between 0.169 and 0.616, so all the models do at least have better regression performance than just predicting the average, although the linear models are barely scraping by. The Somers' D are all quite close to each other, ranging between 0.691 and 0.764, so even though the regression performances vary largely, the items all ordered very similarly regardless of the model.

7.2 Third quartile

As can be seen from the table of model performances (Table 6.1), each of the models gets their worst SMAPE performance on the third quartile, i.e. the top 50-25 percent of items. To get a better of understanding of this phenomenon, a closer look was taken at what kind of items are in that and the surrounding quartiles and what kind of predictions they were getting.

Quartile	True price avg	True price median	Rating avg	Stats avg
Q4	1133661	597056	91.63	86.94
Q3	45003	36623	87.88	78.50
Q2	14305	14132	82.08	73.53

Table 7.1: Price and item stat information of second, third and fourth quartile of items by true price. Only outfield players used for stats avg. Stats avg = average of adjusted pace, dribbling, position stat, and physicality.

From table 7.1 it can be clearly seen that even though the numerical features of the items grow fairly linearly when going from lower priced quartile to higher priced quartile, the growth in price is far from linear. Going from Q2 to Q3, true price avg grows by 3.16 times and the median by 2.59 times, but going from Q3 to Q4, true price avg grows by 25.16 times and the median by 16.30 times. This same effect can be seen in Figure 2.1, where the boxplot whiskers are barely visible, but there is a large amount of outliers. It's reasonable to expect that this is the cause of the poor performance on Q3, but let's get confirmation.

Model	Q4 avg	Q4 mdn	Q3 avg	Q3 mdn	Q2 avg	Q2 mdn
k-nearest neighbors	890214	584444	124837	64581	23454	14631
Radius nearest neighbors	466735	428092	179586	166854	48577	29239
Linear regression	499913	456639	256473	263533	115466	110667
Ridge regression	497983	457924	259265	264646	116668	111550
Decision tree	1221509	615156	194977	46793	23205	13762
Random forest	1279323	781685	190524	75224	22171	14218
Gradient boosting	1318935	740145	163106	73606	23632	10144
Multilayer perceptron	964489	950983	156060	69109	22023	10000

Table 7.2: Table of SMAPE prediction averages and medians by quartile.

As it can be seen by comparing true price medians and averages (Table 7.1) and prediction averages and medians (Table 7.2), the biggest issue in prediction performance is that despite towards the highest rated items, even though the features keep growing linearly, the prices grow exponentially. This is clearest in Q3, where with most models the prediction means and averages are multiple times larger than their true values. In Q2, prediction averages are also moderately larger than true averages, which suggests that the exponential price growth also affects some predictions even there.

7.3 Missing features

Gathering the data from a third-party website instead of getting it directly from the makers of the game itself, also leads to some data being unavailable that could be helpful.

For instance, right now there is no available information about how much each item is listed and bought on the transfer market, so the models just assume that the supply and demand for each item is uniform. Related to supply and demand, the models also assume that each item has the same likelihood to be acquired from a pack. Knowing an estimation of how many instances of a new item would enter the game would very likely help improve performance and differentiation between the third and fourth quartile, as the highest rated items also have the smallest chances of coming out of a pack, which drives up their prices even further.

8 Conclusion

The purpose for this thesis was to see if machine learning based regression methods could be used to predict the prices of new items on the FIFA Ultimate Team transfer market to improve the player experience and reduce chances of foul play on the market. Even though the predictions showed that there is potential in the machine learning methods used, due to the exponential rise in prices in the highest rated items, the models tended to overestimate the prices of many items, as the higher prices 'leaked' into the lower rated items.

8.1 Summary

To get a better understanding of the domain, some research into the literature that exists regarding the FIFA game series and Ultimate Team was done. At the start of the thesis itself, the data needed to be gathered. The process for data gathering, fixing, and filtering is described in more detail in chapter 3. As the data gathered is time sensitive, meaning that more data becomes available as the timestamp used increases, the process for training the models and making predictions needed to be planned so that future data doesn't have an effect on the past. This approach is covered in the third chapter, along with how the numerical features were scaled and what performance metric was used.

The initial data had too many features for them all to be used. Which features were discarded, and which new features were created are listed in chapter 4, along with their descriptions and the final list of features selected. With the features chosen and ready to use, chapter 5 goes over which models were used for making predictions, as well as a brief introduction on how they work. In total, eight machine learning models with multiple different approaches were tested.

Chapter 6 goes over the results that are at the end of the fifth chapter. It also describes the biggest issue with the current situation and what features that aren't available right now could mitigate it.

References

[1] *Ultimate team revenue 2018-2021*, Last visited 20.1.2023. [Online]. Available: https://www.statista.com/statistics/217474/electronic-arts-ea-ultimate-team-revenue/.

[2] M. A. Al-Asadi and S. Tasdemır, “Predict the value of football players using fifa video game data and machine learning techniques”, *IEEE Access*, vol. 10, pp. 22 631–22 645, 2022. DOI: 10.1109/ACCESS.2022.3154767.

[3] L. Cotta, “Using fifa soccer video game data for soccer analytics, documento presentado en el 2016 workshop on large scale sports analytics”, *San Francisco*, vol. 14, 2016.

[4] J. Shin and R. Gasparyan, “A novel way to soccer match prediction”, *Stanford University: Department of Computer Science*, 2014.

[5] P. Siuda, “Sports gamers practices as a form of subversiveness–the example of the fifa ultimate team”, *Critical Studies in Media Communication*, vol. 38, no. 1, pp. 75–89, 2021.

[6] P. Siuda and M. R. Johnson, “Microtransaction politics in fifa ultimate team: Game fans, twitch streamers, and electronic arts”, *EA Sports FIFA: Feeling the Game*, p. 87, 2022.

[7] J. S. Lemmens, “Play or pay to win: Loot boxes and gaming disorder in fifa ultimate team”, *Telematics and Informatics Reports*, vol. 8, p. 100 023, 2022.

[8] *Futbin*, Last visited 20.1.2023. [Online]. Available: `https://www.futbin.com/`.

[9] *Json-format*, Last visited 20.1.2023. [Online]. Available: `https://www.json.org/json-en.html`.

[10] *Csv-format*, Last visited 20.1.2023. [Online]. Available: `https://www.ietf.org/rfc/rfc4180.txt`.

[11] S. Makridakis, "Accuracy measures: Theoretical and practical concerns", *International Journal of Forecasting*, vol. 9, pp. 527–529, 4 1993.

[12] A. de Myttenaere, B. Golden, B. L. Grand, and F. Rossi, "Mean absolute percentage error for regression models", *Neurocomputing*, vol. 192, pp. 38–48, 2016.

[13] R. H. Somers, "A new asymmetric measure of association for ordinal variables", *American Sociological Review*, vol. 27, no. 6, pp. 799–811, 1962.

[14] S. Wright, "Correlation and causation", *Journal of Agricultural Research*, vol. 20, pp. 557–585, 1921.

[15] T. Cover and P. E. Hart, "Nearest neighbor pattern classification", *IEEE Transactions on Information Theory*, vol. 13, pp. 21–27, 1 1967.

[16] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009, vol. 2, pp. 44–48.

[17] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems", *Technometrics*, vol. 12, pp. 55–67, 1 1970.

[18] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009, vol. 2, pp. 307–308.

[19] T. K. Ho, "Random decision forests", in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 1, 1995, 278–282 vol.1. DOI: 10.1109/ICDAR.1995.598994.

[20] J. H. Friedman, "Greedy function approximation: A gradient boosting machine", *Annals of statistics*, pp. 1189–1232, 2001.

[21] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009, vol. 2, pp. 389–414.

[22] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain.", *Psychological Review*, vol. 65, pp. 386–408, 6 1958.