# Computational DOTA Science: Price prediction of virtual items in online gaming economies.

Bharat Atul Desai
1002098
bharat_atul@mymail.sutd.edu.sg

Foo Lin Geng
1002504
lingeng_foo@mymail.sutd.edu.sg

Lim Yuet Tung
1002416
yuettung_lim@mymail.sutd.edu.sg

CONTENTS



***Abstract*—Over the last decade, the virtual item market in online gaming economies has spiraled in value to an estimated US$15 billion in 2016. As such, this segment of online gaming has become the de facto revenue model for game publishers. These items often operate in an exchange market structure similar to that of company issued equities in stock markets, whereby participants can purchase or sell these commodities to other players. These goods can increase or decrease in value over time depending on players prevailing attitudes, consumption patterns and in-game design decisions made by publishers. In this project, we examine trends in virtual item prices in the DOTA 2 economy over time, and build several price prediction models using statistical and machine learning techniques.**

***Index Terms*—Virtual Economy, Machine Learning, Price Prediction**

## I. INTRODUCTION

Virtual goods are non-physical objects and money purchased for use in online communities or online games[1]. These objects can be bought, sold, or traded, much like physical objects or commodities, by users who own them, through in game transaction transaction mechanisms. Open source information of all virtual item transactions allow third party platforms to summarize market behavior of these virtual items and provide services to customers about virtual item price information. In figure 1, we observe the price movement of a particular virtual item in the DOTA 2[2] economy. Similar to other financial products such as an equity or an option, people can profit off of a trading strategy that involves buying the virtual item asset at a low price and selling it a higher price. Trading


**Fig. 1:** Price chart of Dota 2 Virtual Item

[1]https://en.wikipedia.org/wiki/Virtual_goods
[2]https://en.wikipedia.org/wiki/Dota_2

strategies often involve utilizing technical and fundamental strategies, or both in order to generate alpha (ie. a term used to describe a strategies ability to generate above average profit). Correspondingly, we see that there is room to do the same here; by analyzing certain trends and movements, there is room to systematically identify entry and exit points for virtual item purchasing decisions. With this as a motivating scenario, we begin this paper by analyzing price movements of various virtual items in the Dota 2 economy through data visualization and technical analysis. Combining what we learn from the initial exploratory data visualization, we utilize a data driven approach by scraping both in-game data and external indicators, in order to characterize learning models with the objective of virtual item price prediction.

## II. Relevant Literature

"People pay real money for virtual items for the same reason they pay real money for any consumer commodity, namely status, identity, membership, class, and performance." as put by Vili Lehdonvirta, an Associate Professor at Oxford University. The initial boom in the secondary gaming economy driven by fiat based transactions for virtual items started with the game known as EVE online in 2012[3]. The games secondary market revenue per day reached a value of $35K and so its secondary market revenue per year: $12.775 million. The virtual item economy for games has now reached a valuation of approximately $52 billion, with a CAGR (Compounded annual growth rate) of $\geq 6\%$. More recently, the wave of disruption of cryptocurrency has seen platforms making attempts to further bridge the gap between virtual items and fiat transactions. In this paper, we focus our analysis on the game known as DOTA 2, in which the secondary virtual item economy revolves around cosmetic items (goods that only enhance the visual impact of the game, and does not affect in-game performance of users). DOTA 2 is widely regarded has one of the most revolutionary games ever made, changing the video game landscape with competition prize pools in excess of a million dollars. The number of cosmetic items in DOTA 2 is in excess of several thousands, and is constantly growing. These cosmetic items are often differentiated by characteristics such as rarity and qualities. Rarites refer to the probability of randomly obtaining an item via playing the game, and qualities are a prefix given to names of items, essentially creating a separate version of the same item. More details would be discussed in the sections after regarding the effects of these characteristics. In general, there has been little to no effort by the research community to study and discuss the relevance of applying quantitative finance models to monetizing on virtual items. In this paper, we hope to be able to build predictive models capable of learning specific item price movements and make accurate forecasts that could be used drive investment decisions for virtual items.

[3]http://www.avatarwithin.com/mmorpg/estimation-of-mmorg-secondary-market-size/

## III. Information Retrieval & Data Scraping

In this section, we describe the approaches taken to obtain the relevant data in order to build a learning model to predict prices of virtual items in the Dota 2 economy. Majority of the data was obtained through web scraping and querying open source API's made available by the game developers.

### A. Price Data

Prices of Dota 2 virtual items were obtained by scraping uniform resource locators (URLs) of the official steam[4] community web service[5]. The web service displays accurate transaction details by querying private API's made available by the DOTA 2 game developers. Figure2 shows a screen-shot of an alternative version of the item in Figure 1, as well as the type of price and transaction details that are available for users of the website. As demonstrated in figure 2, information such


**Fig. 2:** Price chart of Inscribed Baby Roshan

as the median sales price over various time periods, as well as the transaction count is made available to patrons. In order to determine how the webpage was querying for the data we examined the webpage's network behavior through chrome's developer tools options and found that the chart function was regularly querying a list declared by the name 'line 1': where price and transaction count information was regularly being appended. A screenshot that displays a sample of the list is displayed below in Figure 3. After scrutinizing the webpage's

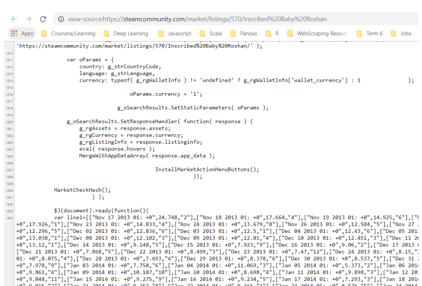

**Fig. 3:** List declaring item prices in HTML page source script of each DOTA 2 virtual item

source HTML code, we found that the list was being stored

[4]Steam is a digitial distribution platform that hosts games from various developers and offers premium servers that allow players to play the game around the world.

[5]https://steamcommunity.com/market

inline. As such, we utilized the Beautiful Soup framework to obtain samples of the HTML code, and regex (ie., Regular expression) to parse and evaluate the data). In addition to price and transaction data, we were also able to obtain information about the item's rarity and quality attributes.

### B. In-Game Data

Utilizing Dota 2's open-source 'OpenData' API, we were able to query for specific in-game data about the item's usage. During the period we surveyed the API, the number of games per day could range from 1-2 million. We sampled in-game data from every 1 in 1000 games by running our code on Google Cloud servers. However, as we were only using the free version, we were blocked by OpenDota systematically after requesting approximately 30 day's worth of data. By changing our IP address, we were eventually able to get approximately 14 months worth of in-game data. For each game, we would record the cosmetics used, check for the location of the game (China, SEA, etc), check for the skills of the match in the game (Normal, High, Very High) which the game appeared, the average ranks of the players using the items, the patch/version of the game, and which team won. Over the games, we would also record how often a hero was picked. We highlight that the skill of the game and the average rank of the players are two different measures of the player's skill[6].

### C. Sentiment Analysis

Additionally, we explored the utilization of sentiment data by scraping dota forums, and examining the frequency of the words related to the hero that the item is associated with. To understand general sentiment pertaining to certain item, we use sentiment regarding to the item-wielding hero as an approximation.

*1) Forum Scraping:* A sample of a forum post[7] that was scraped is show in Figure 4 Sentiment data were first scrapped

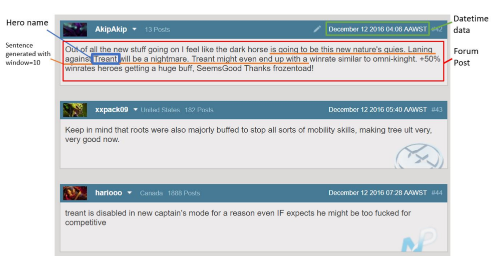


**Fig. 4:** Sample forum post

from dota forum liquiddota.com. Using urllib.request and beautifulsoup package, from an initial url, we scraped for the following information:

1) All forum post texts
2) Their associated datetime information
3) all relevant urls from that page which we can navigate to next (constrained within the www.liquiddota.com domain)

Using new urls found from the page, we navigate to the next url and repeat step 2 recursively Once program visited all the urls found the program search through all the paragraphs to find all hero-related sentences. The program output a csv file with each row containing:

1) Hero ID – the hero which the sentence is related to
2) Sentence – the sentence which contains sentiment data, length specified by the window size
3) Date – date which the sentence is written

A sample of the output from the scrape is shown below in Figure

| | A | B | C |
|---|---|---|---|
| 1 | hero_ID | Sentence | Date |
| 2 | 1 | you are forced to go into it (like vs an anti-mage who has very high mobility). The awful on Death Prophet | 2/4/2013 |
| 3 | 1 | than aghs/refresher otherwise looks fine, though lich is excellent against antimage as long as you take a level of frost armor | 1/30/2013 |
| 4 | 1 | But I guess if you want a carrier that beats magina late game, I would tell you Phanton lancer. | 11/1/2011 |
| 5 | 1 | can't wait to get Dota 2 so I can try antimage out. | 11/1/2011 |
| 6 | 1 | without losing a ton of towers/your team feeding these heroes, antimage can do really really well. oh and do yourself a | 4/11/2016 |
| 7 | 1 | without losing a ton of towers/your team feeding these heroes, antimage can do really really well. Just because they are int | 4/12/2016 |
| 8 | 1 | the go to build. d May 01 2016 10:21 GMT antimage is like probably the only hero that benefited from abyssal | 5/1/2016 |
| 9 | 1 | 2016 12:53 GMT Thoughts on pt urn vanguard midgame oriented antimage with the new vanguard? just came up with the idea | 6/1/2016 |
| 10 | 1 | see more like this. I just learned a ton about antimage that I had no idea about. | 12/26/2017 |
| 11 | 1 | But I guess if you want a carrier that beats magina late game, I would tell you Phanton lancer. | 11/1/2011 |
| 12 | 1 | can't wait to get Dota 2 so I can try antimage out. | 11/1/2011 |
| 13 | 1 | was intended for pub play, but even in pub games antimage is NOT a hero that forces your team to play | 11/2/2011 |
| 14 | 1 | abused in pubs. Also, you probably know this, but picking antimage when playing in a pub game, or among friends, is | 11/2/2011 |
| 15 | 1 | pretty funny. d November 02 2011 04:21 GMT everyone hates antimage because hes decidedly imba | 11/2/2011 |
| 16 | 1 | I got multi stunned and died.. then guess what, their antimage took it and we lost. So since then, I never | 11/2/2011 |
| 17 | 1 | 03 2011 04:08 GMT I've seen someone get lothar's on anti-mage once, two years later I'm still wondering why. And since | 11/3/2011 |
| 18 | 1 | 2011 13:08 Count9 wrote: I've seen someone get lothar's on anti-mage once, two years later I'm still wondering why. And since | 11/3/2011 |
| 19 | 1 | games lately (mostly the joindota vods). so im focussing on anti-mage while keeping this guide in mind, just to get a | 11/4/2011 |
| 20 | 1 | reduction so strong in dota (think spectre, bristleback, old visage, antimage etcetc). its the cumulative reduction with additional flat 'mitigation' to | 11/5/2011 |
| 21 | 1 | November 05 2011 16:27 GMT lol i just saw zsmj magina pub build pms wand phase agi treads BF vlads 3 | 11/5/2011 |
| 22 | 1 | 06 2011 12:42 GMT nice guide, just played against your antimage maybe i should write a guide on venomancer | 11/6/2011 |
| 23 | 1 | tips? i've been running vanguard yasha early-mid game items for antimage nowadays. | 6/27/2014 |
| 24 | 1 | anti-mage. Also, when is it ok to build midas on anti-mage over getting a fast battlefury? | 6/28/2014 |
| 25 | 1 | anti-mage. Also, when is it ok to build midas on anti-mage over getting a fast battlefury? last question: never imo, your | 6/28/2014 |

**Fig. 5:** Output from forum scrape

A total of 52,845 sentences were scrapped from 5,199 urls.

*2) Sentiment Data Preprocessing:* After generating the sentences, they are cleaned by removing:

1) Repeated Sentences
2) Punctuations
3) Symbols
4) Stopwords

Due to a lack of labelled forum sentences data, we utilized the twitter training and testing datasets provided in lab3 as our training and testing data respectively, and then apply the model built based on them to predict sentiments in Figure 6 The training and testing datasets are similarly cleaned through above process.

| | A | B | C |
|---|---|---|---|
| 1 | hero_ID | Sentence | Date |
| 2 | 1 | forced go like vs antimage high mobility awful death prophet | 2/4/2013 |
| 3 | 1 | aghs refresher otherwise looks fine though lich excellent antimage long take level frost armor | 1/30/2013 |
| 4 | 1 | guess want carrier beats magina late game would tell phanton lancer | 11/1/2011 |
| 5 | 1 | cant wait get dota 2 try antimage | 11/1/2011 |
| 6 | 1 | without losing ton towers team feeding heroes antimage really really well oh | 4/11/2016 |
| 7 | 1 | without losing ton towers team feeding heroes antimage really really well int | 4/12/2016 |
| 8 | 1 | go build may 01 2016 1021 gmt antimage like probably hero benefited abyssal | 5/1/2016 |
| 9 | 1 | 2016 1253 gmt thoughts pt urn vanguard midgame oriented antimage new vanguard came idea | 6/1/2016 |
| 10 | 1 | see like learned ton antimage idea | 12/26/2017 |
| 11 | 1 | intended pub play even pub games antimage hero forces team play | 11/2/2011 |
| 12 | 1 | abused pubs also probably know picking antimage playing pub game among friends | 11/2/2011 |
| 13 | 1 | pretty funny november 02 2011 0421 gmt everyone hates antimage hes decidedly imba | 11/2/2011 |
| 14 | 1 | got multi stunned died guess antimage took lost since never | 11/2/2011 |
| 15 | 1 | 03 2011 0408 gmt ive seen someone get lothars antimage two years later im still wondering since | 11/3/2011 |
| 16 | 1 | 2011 1308 count9 wrote ive seen someone get lothars antimage two years later im still wondering since | 11/3/2011 |
| 17 | 1 | games lately mostly joindota vods im focussing antimage keeping guide mind get | 11/4/2011 |
| 18 | 1 | reduction strong dota think spectre bristleback old visage antimage etcetc cumulative reduction additional flat mitigation | 11/5/2011 |
| 19 | 1 | november 05 2011 1627 gmt lol saw zsmj magina pub build pms wand phase agi treads bf vlads 3 | 11/5/2011 |
| 20 | 1 | 06 2011 1242 gmt nice guide played antimage maybe write guide venomancer | 11/6/2011 |
| 21 | 1 | tips ive running vanguard yasha earlymid game items antimage nowadays | 6/27/2014 |
| 22 | 1 | antimage also ok build midas antimage getting fast battlefury | 6/28/2014 |
| 23 | 1 | antimage also ok build midas antimage getting fast battlefury last question never imo | 6/28/2014 |
| 24 | 1 | farm speed much midas helps antimage isnt huge needing levels mostly raw gold | 6/28/2014 |
| 25 | 1 | antimage also ok build midas antimage getting fast battlefury optimize everything minmax | 6/28/2014 |

**Fig. 6:** Cleaned forum data scrape

*3) Model Selection for Sentiment Analysis:* Two different models were attempted for the sentimental analysis task, simple BoW model and Doc2Vec classification model. For BoW model, we obtained accuracy of 0.805 when using Multinomial Naïve Bayes classifier, and accuracy of 0.797 when using Logistic Regression classifier. For Doc2Vec model, we obtained accuracy of 0.808 when using Logistic Regression

[6]https://docs.opendota.com/#
[7]http://www. liquiddota.com

classifier. Since the Doc2Vec model produces the highest accuracy score, the model is used on the cleaned forum scrape to generate a sentimental analysis report, detailing the number of positive and negative sentiment for each hero on each date. This represents sentimental data which will be used to subsequently predict price changes.

| 1 | hero_ID | date | Pos | Neg |
|---|---|---|---|---|
| 2 | Razor | 1/2/2011 | 0 | 1 |
| 3 | Queen of Pain | 1/2/2011 | 0 | 2 |
| 4 | Night Stalker | 1/2/2011 | 0 | 1 |
| 5 | Nature's Prophet | 1/3/2011 | 1 | 0 |
| 6 | Troll Warlord | 1/3/2011 | 1 | 0 |
| 7 | Keeper of the Light | 1/4/2011 | 1 | 0 |
| 8 | Lone Druid | 1/6/2011 | 1 | 0 |
| 9 | Keeper of the Light | 1/6/2011 | 1 | 1 |
| 10 | Winter Wyvern | 1/6/2011 | 1 | 0 |
| 11 | Dark Willow | 1/6/2011 | 1 | 0 |
| 12 | Witch Doctor | 1/7/2011 | 1 | 0 |
| 13 | Night Stalker | 1/7/2011 | 1 | 0 |
| 14 | Keeper of the Light | 1/7/2011 | 1 | 0 |
| 15 | Crystal Maiden | 1/11/2011 | 1 | 0 |
| 16 | Queen of Pain | 1/12/2011 | 0 | 1 |
| 17 | Tiny | 1/23/2011 | 1 | 0 |
| 18 | Death Prophet | 1/25/2011 | 1 | 0 |
| 19 | Keeper of the Light | 1/30/2011 | 1 | 0 |
| 20 | Night Stalker | 2/1/2011 | 1 | 0 |
| 21 | Lone Druid | 2/2/2011 | 1 | 0 |
| 22 | Dark Willow | 2/2/2011 | 1 | 0 |
| 23 | Razor | 2/12/2011 | 1 | 0 |
| 24 | Keeper of the Light | 2/18/2011 | 1 | 0 |
| 25 | Death Prophet | 2/27/2011 | 0 | 1 |

**Fig. 7:** Sentiment Analysis Report

*4) Challenges and Limitations:* Data collected was sparse for the time period considered since the forum we scraped from, liquiddota.com, is a relatively old and unpopular forum. We tried to scrape from other, more popular forums such as dotabuff.com but were thrown HTTP error 429 (Too Many Requests) after scraping for just a few urls, possibly due to security features employed by the host to prevent server overload. When running the Doc2Vec model, since we are training on 1.6 million data points and each contains 300 features, running on local CPU causes it to lag and run out of memory quickly. This was resolved by running the script using Google Colab's accelerated GPU.

After merging all the relevant data with the price information of chosen items, we obtained a final dataset to be utilized for modelling purposes, which is displayed in Fig:_ below.

| List of predictor variables | Description |
|---|---|
| date | Date for which the data is recorded |
| price | Mean of median price estimates taken at different times on a date |
| sell_count | Total number of trades of the item that day |
| item_quality | Quakity prefix of the item |
| item_rarity | Rarity index of the item |
| total_apps_percent | how often the item-wielding hero was picked that day |
| skill_none_percent | Proportion of players in-game, wearing the item that classify as none-low skill players |
| skill_normal_percent | Proportion of players in-game, wearing the item that classify as normal skill players |
| skill_high_percent | Proportion of players in-game, wearing the item that classify as high skill players |
| skill_very_high_percent | Proportion of players in-game, wearing the item that classify as very high skill players |
| win_percent | Porportion players in-game, wearing the item that win a game. |
| rank_tier_percent | Proportion of players in-game, wearing the item that have ranks. |
| avg_rank_tier | average ranks of players using the item |
| patch | patch/ version of the game |
| US West | Proportion of players in-game, wearing the item from US WEST |
| US East | Proportion of players in-game, wearing the item from US EAST |
| Europe West | Proportion of players in-game, wearing the item from EUROPE WEST |
| SE Asia | Proportion of players in-game, wearing the item from SEA |
| Australia | Proportion of players in-game, wearing the item from AUSTRALIA |
| Russia | Proportion of players in-game, wearing the item from RUSSIA |
| Europe East | Proportion of players in-game, wearing the item from EUROPE EAST |
| South America | Proportion of players in-game, wearing the item from S AMERICA |
| South Africa | Proportion of players in-game, wearing the item from S AFRICA |
| China | Proportion of players in-game, wearing the item from CHILE |
| Chile | Proportion of players in-game, wearing the item from US WEST |
| Peru | Proportion of players in-game, wearing the item from PERU |
| South Korea | Proportion of players in-game, wearing the item from S KOREA |
| Positive Sentiment | number of positive sentiment of the item-wielding hero that day |
| Negative Sentiment | number of negative sentiment of the item-wielding hero that day |

**Fig. 8:** Final Dataset with description of featyres

## IV. Exploratory Data Visualization

In this section, we examine the data in order to derive some meaningful relationships through various visualization methodologies. These relationships would later be used to modify and tune our models, to be evaluated by various model testing metrics. In addition to examining the overarching relationships between various data features, we also provide an in-depth examination of the pricing behavior of one of the chosen items. At the end of this section, we also provide a summary of visualizations for the other items utilized in this study.

### *A. Feature Correlation*

We begin our exploratory data analysis by examining correlation between the scraped data that would be used as features for the models. Pearson's correlation is a measure of strength of similarity between two features as a function of one relative to another. The correlation coefficient takes values between -1 and 1, the former denoting a negative correlation behavior, and the latter a positive correlation between the two series of data being examined. A value of 0 on the other hand denotes that the features have no correlation to each other. For n features, a total of $\frac{(n)(n-1)}{2}$ pairwise comparisons are made. Each of these comparisons are represented through a lower triangular matrix as shown in figure 9.

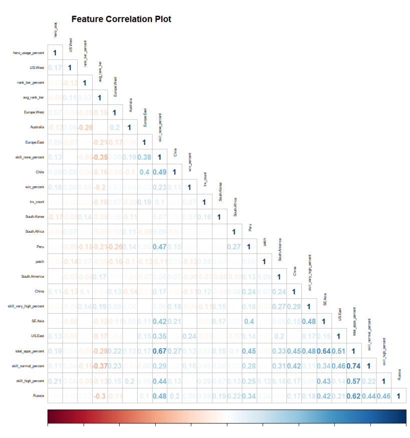


**Fig. 9:** Correlation Plot

The three feature pairs with the highest feature correlation are as follows:

1) Total_apps_percent & skill_normal_percent
2) Total_apps_percent & skill_none_percent
3) Total_apps_percent & SEAsia

As such, it is worth examining if combining these and other features with significant correlation, would result in a better performing model. A further examination of the result of these experimentation is provided in the section: Model performance and evaluation.

### *B. Comparing price movements between item qualities*

Item qualities are various versions of the same item that were released during special events or periods in the calendar

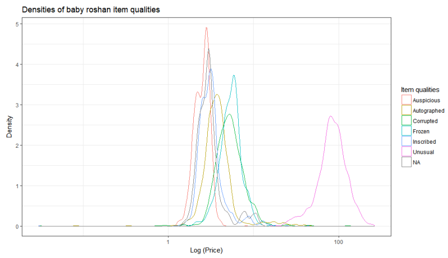


**Fig. 10:** Price densities of various versions of Baby Roshan Qualities

year. In figure 10, we compare some variations of the Baby Roshan item. We notice that there is a significant difference between the ways some of the prices are distributed. The distribution of unusual version is significantly far ahead of most of the other versions, centered at a price closer at \$100, as compared to the rest of the items that vary in price between \$1-\$10. This shows us that each version of the item, or each item quality should be treated as a different item completely that follows a different price movement. Additionally, it is worth noting the distributions of the each of the items prices roughly look like normal distributions. This is also explained by the Central Limit Theorem.

This idea is reinforced in figure 11 where the price movements of the item versions are compared over time. Each item has its own characteristics of peaks and troughs, as well as spikes in valuations.

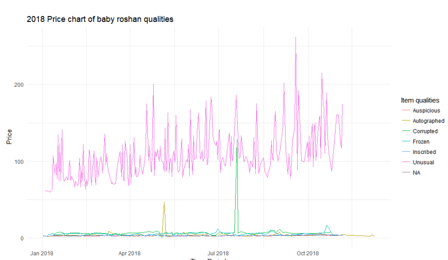


**Fig. 11:** Price history of various Baby Roshan item qualities in the year 2018.

### C. Item-specific visualizations

Next, we go through the sequence of analysis that was performed at an item level. We begin each analysis by first going through the price and transaction volume history of an item since its conception. In this section, we give an indepth analysis of one particular item, and provide a summary for other items at the end.

In the Figure 12, we notice that there is clearly a seasonality trend in the price movement, as well as two clear spikes in relative value of the item. As such, to better examine if the seasonality trend is consistent, we overlap the price movement over the years in figure below. From figure 13 above, it becomes even more clear that the seasonality of the data is consistent over the years trending at a price around the \$0.06 mark. The seasonal movement is better visualized when we examine the same trend with overlapped yearly seasons in

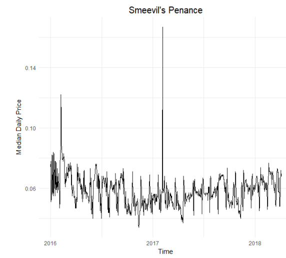


**Fig. 12:** Price history of the item "Smeevil's Penance"

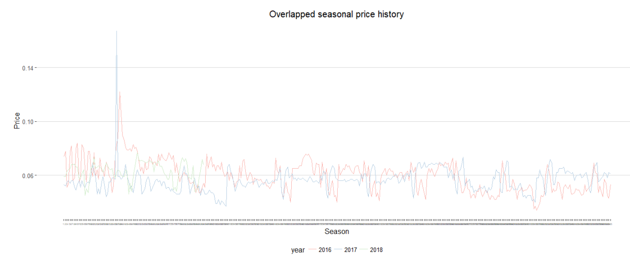


**Fig. 13:** Price movement of Smeevil's Penance overlapped over the years

polar coordinates in fig14 .

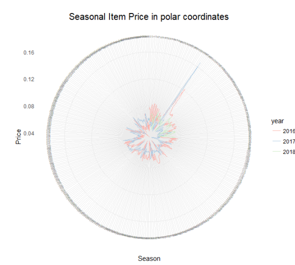


**Fig. 14:** Price movement of Smeevil's Penance overlapped over the years, in polar co-ordinates

The periods between the peaks and troughs are relatively short, and could represent a monthly or even a weekly trend. Interestingly, we notice that the large spikes in price even happened at the same time in the year. This could perhaps be attributed to a the presence of an annual event such as an international gaming competition, which would result in an increase in usage of the item. In the figure below, we examine the frequency of usage of the item, with yearly seasons overlapped again. From Fig 15, in 2016, we notice that there was indeed, a sharp spike in transactions being made in the same period of time as the spike in price movement. This examination thus prompts a discussion about possible events that might cause significant signaling for price movement. Additionally, we further examined the price movement by taking the median of the price movement on a certain day of the year and plotting the price medians over the 365 day season. We can think of the subseries plot in figure 16 as individual boxplots of prices for a certain day of the year.
Next, we examine the seasonality of the item price in further detail. We do this by checking whether there is significance be-

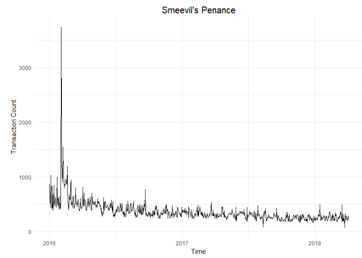


**Fig. 15:** Transaction count history with years overlapped for the item 'Smeevil's Penance'

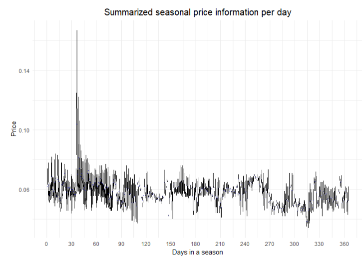


**Fig. 16:** Daily subseries plot of Smeevil's Penance over a 365 day period

tween the item's price at the current time with a previous time-step. The difference in the number of time steps compared is known as lag. Each comparison is regressed onto a linear function, where the closer the regression of the lag is to the linear function, the greater the significance of linear correlation there is between the two time-steps. This is illustrated in figure 17 below. We notice that the most correlation occurs in lag

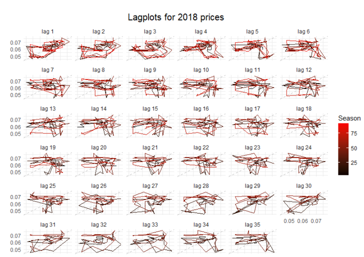


**Fig. 17:** Lag plots of Smeevils Penance

1, the timestep immediately before, and lag 30, the timestep approximately one month before.

In order to further cement these observations about seasonality, we examine the auto-correlation of price movement of the item. Auto-correlation is the correlation between the elements of a series and others from the same series separated from them by a given interval. In figure 18, the x axis denotes the lag for each auto-correlation estimate, where the auto correlation estimate is denoted by the height of each bar. Note that the auto-correlation at a time lag 0 is always 1. The figure also displays a pair of blue, horizontal, dashed lines, representing lag-wise 95% confidence intervals centered at zero. These are used to determine the statistical significance of individual auto-correlation estimates at a particular value of lag, versus the null value at 0. Essentially, The dashed blue lines indicate whether the correlations are significantly different from 0. (Hyndman, 2018)

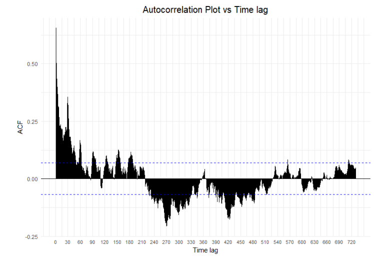


**Fig. 18:** Autocorrelation plot of Smeevils Penance Prices.

Here, we observe spikes in periodicity at low time lags, and another short spike at time lags of approximately every 30 days, as can be seen in figure 18.

In summary, the analysis from the exploratory data from one item has led us to believe that price changes are heavily seasonal and changes in price movement are signalled by the occurrence of events. This leads us to believe that models that are sensitive to seasonal behavior would best be suited for predicting the price of a certain virtual item.

Next, we run exploratory statistical tests to explore the data even further.

## V. Exploratory Statistical Tests

Time series data has 3 main characteristics: Drift, Heteroskedasticity and Periodicity. We ran three corresponding exploratory tests (Vlachos, 2005) on our data to get a better understanding of our time series price data.

### *A. Levene's Test*

To test for heteroskedasticity, we used the Levene's Test. It is used to test for equality of variances between two or more groups. We arbitrarily split our data into 10 segments and ran this test on all our time series data. All of them showed significant heteroskedasticity at a significance value of 0.05.

### *B. Friedman Test*

To test for drift, we used the Friedman Test, which is a form of ANOVA that tests for shifting trends between groups. We decided to use this test because our data has been shown to be heteroskedastic, so the basic assumptions for basic ANOVA do not hold. The Friedman Test is non-parametric and does not require assumptions of normality. It is also stricter and somewhat harder to reject the null hypothesis that the means are the same. We arbitrarily split our data into 10 segments and ran this test on all our time series data. All of them showed significant drift at a significance value of 0.05.

### *C. Auto-Period Test*

Periodicity is a huge part of our project, as we wanted to reliably test for seasonal trends. Our team decided to test for periodicity using a state-of-the-art method called Auto-period test.

This method combines the usage of Discrete Fourier Transform and Autocorrelation. The Discrete Fourier Transform generates candidates for periods that are above a certain threshold of confidence. The candidates are then validated through the Autocorrelation test. Intuitively, the candidate generation phase has a low False negative rates, and the candidate validation phase has a low false positive rates, leading to highly accurate results.

In order for a period to be significant, we defined 95% confidence for the DFT test and 0.5 for the Autocorrelation test. No statistically significant periods were found for any of our data. Thus, even though we observe some periodic trends visually, they are still not statistically strong enough, probably due to the insufficient data collected.

## VI. Price Prediction Models

In this section, we briefly describe the models we applied to the data. The graphs used are from the analysis of the Smeevil's Penance dataset.

### A. ARIMA

We first use a moving average over the data (of 7 days) to smoothen the data.

We then use the STL (Seasonal and Trend Decomposition using Loess) method to separate the time series data into components: Seasonal, Trend and Remainder. This is because ARIMA works best only on unseasonal and stationary data, so we attempt to remove the seasonality component here.[8]

The resulting figure 19 depicts the decomposition into 3 separate components. After removing the seasonal component,

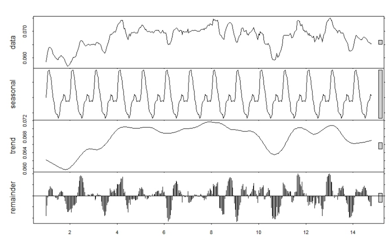


**Fig. 19:** STL Decomposition

we test the data with the Augmented Dickey-Fuller test to check if a unit root is present in our "un-seasonal" time series data. For all our data, stationarity was satisfied after differencing the data once, with a p-value of $\leq 0.05$. We also checked our Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots, which actually showed some signs of periodicity every 7 days for our data. However, they were not significant trends and so we proceeded to generate our ARIMA function.

Once our assumptions of stationarity was satisfied, we could generate our ARIMA model on our time series data. We picked the model with the highest log-likelihood values.

[8]https://otexts.org/fpp2/stl.html

| Item | ARIMA Model |
|---|---|
| Smeevil's Penance | ARIMA(1,1,4) |
| Shifty Minnow | ARIMA(0,1,0) |
| Scavenging Guttleslug | ARIMA(0,1,5) |

**Fig. 20:** Table of ATIMA models

Our projections for the Smeevil's Penance data set are shown in figure 21. We observe that the model has underfit the data. The prior 13 months of data do not seem to be enough for the model to find a good fit.

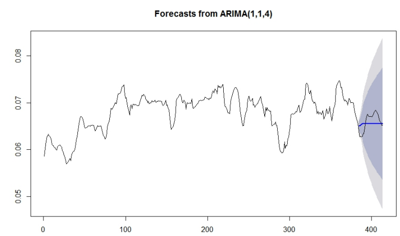


**Fig. 21:** ARIMA Smeevils Penance Projections

### B. ARIMAX

The ARIMAX is simply an ARIMA model, but with added covariates - the in-game predictors. At every step, we include in-game predictors at the same time step as part of the group of predictors.[9]

The projections for the ARIMAX model can be observed below. We no longer observe the underfitting, as before. We

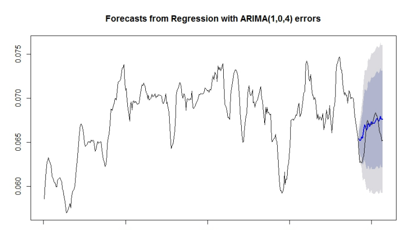


**Fig. 22:** ARIMAX Smeevils Penance projections

will analyze the predictors in the next section.

### C. Ensemble Methods

We also used the following ensemble methods to model the "de-seasoned" time series data: Random Forest, Bagging of Regression Trees, and Bagging of Conditional Inference Trees using the R libraries: *randomForest, rpart and party*.

These tree-based ensemble methods work by generating many decision trees by bootstrapping the data and aggregating them together to form a single model. The decision trees are built through splitting of the data set based on minimizing the

[9]https://robjhyndman.com/hyndsight/arimax/

variance of the resulting segments of a target predictor. We decided to set the number of trees to be 1000.

As the splits are linear, these ensemble methods cannot adjust for increasing/decreasing trends over time. They are unsuited to model our general trend, but can model the seasonality. Thus, we only try to model the "de-trended" time series data using ensemble methods. We add the trend component by running the "trend" from STL decomposition as an ARIMA model, and forecasting that ARIMA model to form a trend. In other words, we are modelling only for the seasonality and using these ensemble methods, not for the trend.[10] We also augmented our data by adding in some seasonal variables. We added in the first two terms of the fourier transform of the weekly and monthly periods as predictors. This allowed for more comprehensive modelling of the non-linear seasonal variations. We also added in monthly/weekly lags as predictors, that is the price one month/week before. With this predictor, we attempt to account for autocorrelation.

We trained two versions of each model - one using only our new augmented predictors, and another one using our in-game predictors as well.

*1) Bagging of Regression Trees:* Regression Trees are trees that splits the data of predictors which can take continuous values. Between Bootstrapped samples, we randomly sampled for the 4 hyperparameters sampling_ratio, minsplit, maxdepth, cp within a large range in order to prevent overfitting.

In figure 23 below, we can see the trend lines predicted by all 100 bootstrapped samples, and the red line represents the aggregated average of all the models. While each individual sample seems blocky and volatile, the aggregated average is rather smooth and well-behaved.

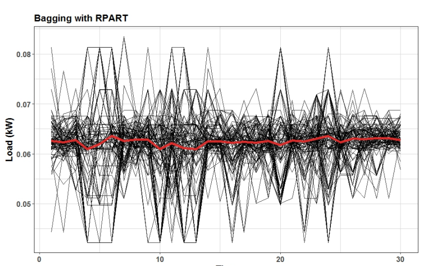


**Fig. 23:** Bagging (Regression Trees)

*2) Bagging of Conditional Trees:* Structure-wise, conditional trees are extremely similar to regression trees and our procedure was also similar. However, Conditional trees uses a significance test procedure in order to select variables instead of selecting the variable that maximizes an information measure, so that the effect of overfitting is minimized.

We again set the 100 bootstrapped samples below.

*3) Random Forest's:* The random forest's method is similar to the regression tree bagging method, except that now only a subset of features would be available to choose from, at each iteration. This method attempts to fix the problem of overfitting on a few main predictors only.

[10]https://petolau.github.io/Ensemble-of-trees-for-forecasting-time-series/

projections.jpeg

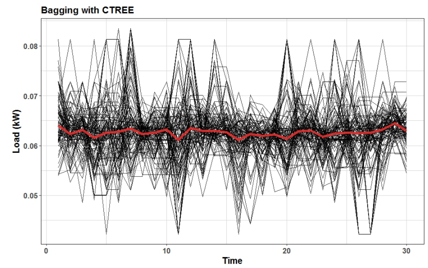


**Fig. 24:** Bagging (Conditional Trees)

We brief tuned the hyperparameters parameters used, and tried up to 5 different values of mtry and nodesize.

Finally, we compare our efficiency between different ensemble methods below.

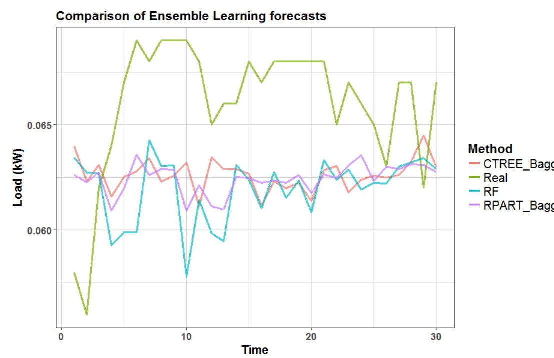


**Fig. 25:** Ensemble Methods Smeevils Penance Projections

## D. Long Short Term Memory (LSTM)

We also attempted a LSTM model. We converted the time series into a supervised learning set by making features at time (t-1) predictor variables for price at time (t) and fed them into a LSTM model.

To fine-tune our LSTM model, we performed feature selection by varying the number of feature selected using SelectKBest function. In Fig26 below, we observe the value of RMSE over different k. We eventually chose to use 13 parameters as it seems to be a good fit. We also tuned for the

**Fig. 26:** LSTM Tuning Parameters

optimum batch size. In Figure 27, we observe the value of RMSE over different batch sizes. We eventually chose to use 31 as our batch size as it seems to be the optimum. Using these hyperparameters, we trained the model for 100 epochs

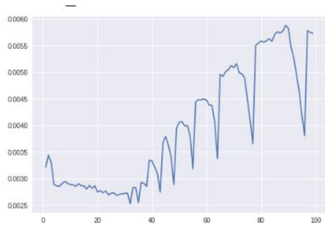

**Fig. 27:** LSTM tuning batch size for Smeevils Penance

on Google Colab and obtained the projections in Figure 28 . We found our model to converge at around 30 epochs, as


**Fig. 28:** LSTM projections, Smeevils Penance

seen from the Figure 29[11].

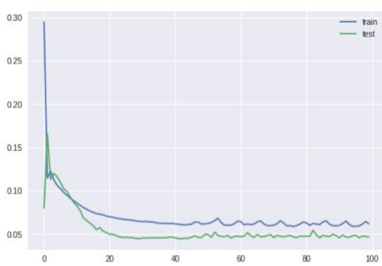


**Fig. 29:** LSTM epoch errors for Smeevils Penance

## VII. Evaluation and Model Performance

In this section, we provide an overview of the methodology used to evaluate the models we utilized. We chose root mean squared error (RMSE) and mean average percentage error (MAPE) as our evaluation criteria. The RMSE is a direct measure of how close our predictions are to actual observations in the test set. The deviations are calculated and squared. The MAPE takes relative scale of data into account and minimizes the relative deviance between actual and forecasted observations.

### A. *Evaluation of test accuracy*

Below, we show our error tables for each method. The test set is 1 month for Smeevil's Penance, and 1 week for Shifty Minnow and Scavenging Guttleslug. Overall, the

[11]Refer to LSTM analysis.ipynb for implementation: https://drive.google.com/drive/u/1/folders/1vhRJrDRwn9NlqHa6najpbyyG1y5OxRd

| | Smeevil's Penance | Shifty Minnow | Scavenging Guttleslug |
|---|---|---|---|
| ARIMA | 2.54 | 2.23 | 2.56 |
| ARIMAX | 1.94 | 2.44 | 1.24 |
| Random Forest | 7.23 | 0.85 | 4.97 |
| Random Forest (with predictors) | 6.72 | 0.78 | 4.74 |
| Regression Tree Bagging | 6.78 | 0.62 | 4.76 |
| Regression Tree Bagging (with predictors) | 6.78 | 0.08 | 5.01 |
| Conditional Tree Bagging | 6.65 | 0.57 | 4.91 |
| Conditional Tree Bagging (with predictors) | 6.34 | 0.08 | 5.27 |
| LSTM | 2.92 | 0.16 | 2.58 |

**Fig. 30:** Table of MAPE values

| | Smeevil's Penance | Shifty Minnow | Scavenging Guttleslug |
|---|---|---|---|
| ARIMA | 1.85e-03 | 7.22e-04 | 2.89e-02 |
| ARIMAX | 1.61e-03 | 7.89e-04 | 1.52e-02 |
| Random Forest | 5.15e-03 | 4.34e-04 | 5.31e-02 |
| Random Forest (with predictors) | 4.88e-03 | 2.66e-04 | 5.10e-02 |
| Regression Tree Bagging | 4.85e-03 | 3.35e-04 | 5.10e-02 |
| Regression Tree Bagging (with predictors) | 4.89e-03 | 6.30e-05 | 5.41e-02 |
| Conditional Tree Bagging | 4.78e-03 | 2.92e-04 | 5.28e-02 |
| Conditional Tree Bagging (with predictors) | 4.59e-03 | 6.30e-05 | 5.57e-02 |
| LSTM | 2.73e-03 | 7.32e-05 | 3.06e-02 |

**Fig. 31:** Table of RMSE values

ARIMAX model performed the best on Smeevil's Penance and Scavenging Guttleslug, and the plain ARIMA model performed the best on Shifty Minnow. This aligns with our original predictions that ARIMAX should work better than plain ARIMA, due to the presence of more information (added covariates). ARIMAX also gave us non-horizontal lines as estimates, which is more indicative of a right fit than ARIMA, which gave us horizontal prediction lines for all three data sets, indicating underfitting.

We believe that Shifty Minnow is an exception, because many of its data points are at $0.03, which is the minimum amount allowable for transactions on Steam. As the price showed little variation through time, our ARIMAX model overfitted on the data and led to a poorer performance compared to the plain ARIMA model, which is really just a (0,1,0) model that predicted a constant price equal to the price observed just before the test set. The "best" models here - the Bagging methods with predictors, simply predicted a constant price of 0.03 throughout, and performed the best without giving us any insight.

On the other hand, we observe that our Ensemble methods really overfitted on the dataset. The non-linearity of the methods caused the models to perform way worse than the plain ARIMA model. The addition of the predictors to the dataset also did not help significantly.

## B. Evaluation of ARIMAX Predictors

```
Series: ts(deseasonal_mean_price[-c(test_start:test_end)])
Regression with ARIMA(1,0,4) errors

Coefficients:
        ar1     ma1    ma2     ma3     ma4  intercept  sell_count  total_apps_percent  skill_none_percent
     0.9340  0.2994  0.067  0.3601  0.6055     0.0636           0              0.0054              0.0017
s.e. 0.0056  0.0565  0.054  0.0400  0.0652        NaN         NaN              0.0055              0.0013
     skill_normal_percent  skill_high_percent  skill_very_high_percent  win_percent  rank_tier_percent
                   0.0012               -8e-04                    4e-04        4e-04              2e-04
s.e.               0.0015                5e-04                    3e-04        2e-04              2e-04
     avg_rank_tier  patch  US.West  US.East  Europe.West  SE.Asia  Australia   Russia  Europe.East
                 0  1e-04    0e+00   -3e-04       -4e-04  -0.0004     -1e-04  -0.0007       -3e-04
s.e.             0    NaN    1e-04    4e-04        8e-04   0.0015      1e-04   0.0014        4e-04
     South.America  South.Africa   China  Chile   Peru  South.Korea
             1e-04        -3e-04  0.0008  0e+00  1e-04       -3e-04
s.e.         3e-04         1e-04  0.0016  2e-04  3e-04        2e-04

sigma^2 estimated as 4.989e-07:  log likelihood=2253.49
AIC=-4446.99   AICc=-4441.72   BIC=-4328.47
```

**Fig. 32:** ARIMAX coefficients Smeevils Penance

We depict our coefficients from the ARIMAX model for Smeevil's Penance in Fig 32 below. Overall, we see that the model only has a very vague idea about the coefficients - the standard error of most coefficients are around the same as the coefficients' values. This is with the exception of the win_percent variable, whose estimated standard error is only half that of the estimated coefficient. The South Africa variable also looks promising, but it might be due to overfitting as the entries for South Africa are quite sparse.

We also observe that the number of transactions in a day is not related to our final result.

```
> fit_no_holdout
Series: ts(deseasonal_mean_price[-c(test_start:test_end)])
Regression with ARIMA(2,1,2) errors

Coefficients:
        ar1      ar2     ma1     ma2  trx_count  total_apps_percent  skill_none_percent  skill_normal_percent
     0.0349  -0.5399  1.8712  0.8984          0             -0.0102              0.0063                0.0110
s.e.    NaN   0.0899  0.1188  0.1112          0              0.0002                 NaN                0.0002
     skill_high_percent  skill_very_high_percent  win_percent  rank_tier_percent  avg_rank_tier  US.West
                 0.0022                   0.0027        1e-04              3e-04              0   -4e-04
s.e.             0.0006                   0.0002          NaN              1e-03              0      NaN
     US.East  Europe.West  SE.Asia  Australia   Russia  Europe.East  South.America  South.Africa    China
      -5e-04      -0.0023  -0.0028      2e-04   -3e-03       -3e-04         -5e-04             0  -0.0031
s.e.   4e-04          NaN      NaN        NaN    2e-04          NaN          3e-04           NaN      NaN
      Chile    Peru  South.Korea
     -7e-04  -1e-04       -2e-04
s.e.  4e-04     NaN          NaN

sigma^2 estimated as 4.377e-09:  log likelihood=521.13
AIC=-988.26   AICc=-943.79   BIC=-930.83
```

**Fig. 33:** ARIMAX coefficients Shifty Minnow

Above, we depict our ARIMAX coefficients for Shifty Minnow. We see that many of our predictors' standard errors are NaN, because there's not enough variability in our data to correctly estimate the standard errors. Since the model overfit the data, these coefficients are not very useful.

```
> fit_no_holdout
Series: ts(deseasonal_mean_price[-c(test_start:test_end)])
Regression with ARIMA(1,0,2) errors

Coefficients:
        ar1     ma1     ma2  intercept  trx_count  total_apps_percent  skill_none_percent  skill_normal_percent
     0.9860  1.9153  0.9336     1.0538          0              0.1753             -0.0833               -0.1[illegible]
s.e. 0.0173  0.0852  0.0791     0.0009        NaN                 NaN              0.0099                0.0208
     skill_high_percent  skill_very_high_percent  win_percent  rank_tier_percent  avg_rank_tier  US.West
                 0.0079                  -0.0295       0.0054            -0.0019              0  -0.0028
s.e.             0.0094                   0.0062          NaN                NaN              0      NaN
     US.East  Europe.West  SE.Asia  Australia   Russia  Europe.East  South.America  South.Africa   China
      0.0005      -0.0279  -0.0230     0.0034  -0.0131       0.0159        -0.0094        -5e-04  0.0841
s.e.  0.0014       0.0069   0.0148        NaN   0.0064       0.0031         0.0030           NaN     NaN
       Chile     Peru  South.Korea
     -0.0013  -0.0057       0.0139
s.e.  0.0016   0.0014       0.0024

sigma^2 estimated as 7.045e-06:  log likelihood=293.91
AIC=-533.82   AICc=-490.62   BIC=-475.96
```

**Fig. 34:** ARIMAX coefficients Scavenging Guttleslug

In Fig 34, we depict our ARIMAX coefficients for Scavenging Guttleslug. Our coefficients are the most significant here, where some standard errors are much less than our coefficient values. We see that having the cosmetic appear in more $very\_high\_skilled, normal\_skilled \& none\_skilledmatches$ lead to a lower price of the good. The trend is opposite for the $high\_skilled$ games. This might mean that $high\_skilled$ matches have some small positive relationship with prices of Scavenging Guttleslug, while the rest are negatively correlated with the price. Interestingly, we refer back to the Smeevil's Penance data set and see that the $high\_skilled$ games show an opposite trend to the other predictors as well.

We see that the total appearances of the item and the win rates are positively related to prices in the same time step.

Games played in Europe West, Europe East, Russia, South America, Peru and South Korea also seem to matter.

## C. Evaluation of Random Forest predictors

In this subsection, we analyze the variable importance that we have obtained from the Random Forest model. Although the Random Forest model as a whole does not predict prices as well as ARIMAX, we still observe that the predictions of prices of Smeevil's Penance and Scavenging Guttleslug improve slightly when we add in the extra predictors, which tell us that the predictors add some value to the price predictions. In Figure 35, we observe that the total appearances of the

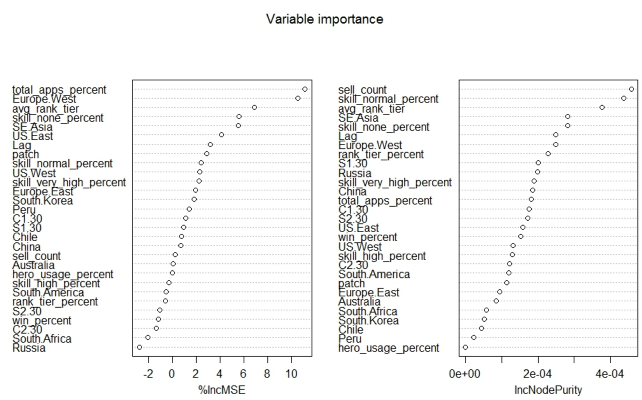


**Fig. 35:** Random Forest Variable Importance for Smeevils Penance

cosmetic item, and the number of games played in Europe West both mattered in lowering the error by about 10%. On the right, we see that the number of transactions is important, and the ranks of the players also matter. The Random Forest model in Fig 36 had decreased price prediction accuracy when we added in the extra variables, so we do not try to analyze it. We also observe that the % increase in MSE and Node Purity is very low, when we take any variable out, so none of these predictors are significant. From the Variable Importance graph for Scavenging Guttleslug above, we observe that the number of games played in US West mattered, and is responsible for decreasing the MSE by 6%. On the other hand, the amount of transactions was a noisy variable, and including it led to a 4% increase in MSE. The node purity was also slightly increased by including the number of appearances of the item, the win rate, and the average rank of the players that featured the cosmetic item.

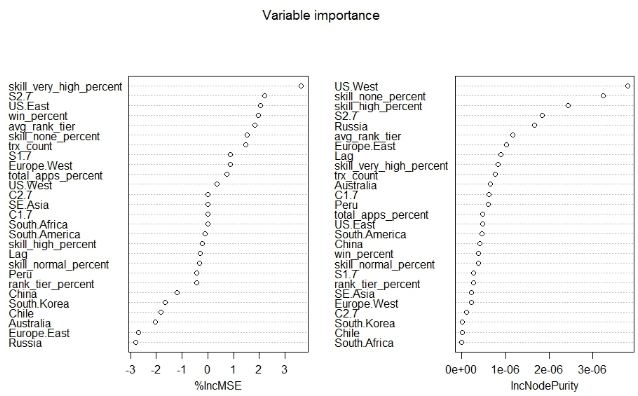


**Fig. 36:** Random Forest Variable Importance for Shifty Minnow

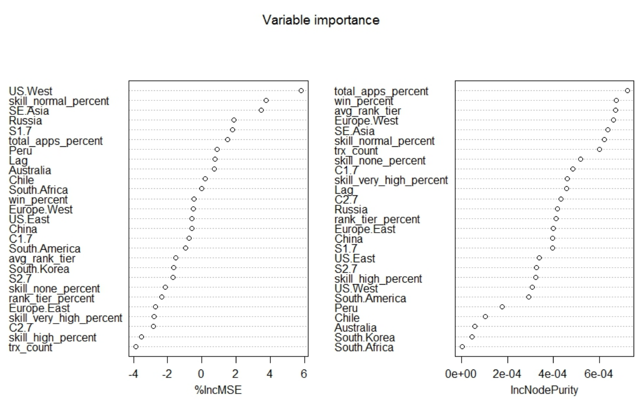


**Fig. 37:** Random Forest Variable Importance for Scavenging Guttleslug

### D. *Overall discussion of predictors*

Overall, from our ARIMAX model, we observe that the rate of appearance of the iterm on the day itself and the win rates of the item increase prices are positively related to price. This matches with our expectations.

We observe some interesting trends with the skills of the matches, with High Skill matches exhibiting opposite trends from the other skill groups. For Scavenging Guttleslug, having higher appearance rates in matches with high skill meant that prices increased, while the trend was inverse for the other skill levels.

On the other hand, we see that the quantity traded of an item was not relevant most of the time, which is a surprise for us. Economic intuition tell us that this is an extremely unexpected result. However, this might be due to the dynamic interaction between demand and supply with the price as a signal, and so the quantity demanded at the time step is not a helpful predictor of the price.

The Random Forest model also show that the total appearances of the item and the win rate of the item mattered. Additionally, the average rank of players with the item was also important - a relationship not picked up by the ARIMAX model.

Overall, we could not find any consistent patterns or trends of location variables being important.

## VIII. Discussion and Conclusion

### A. *Limitations*

In this project, there were a few main limitations.

1) Although we collected as much data as we could, our data is still not comprehensive enough. We were still limited by time and and also faced trouble with the Request bans.
2) We mainly focused on 4 main items throughout the analysis only. While this helped us to focus on interpretability, it is far too small a sample size for it to be representative of general trends.
3) We failed to take into account item correlation. Prices of items are often correlated, especially between bundles of items that belong to the same category (same item slot for the same hero). We might be able to find better predictors from those relationships.

### B. *Conclusion*

In conclusion, we find that traditional ARIMA based methods performed best in predicting future prices of virtual items. Our research finds that there is promising space to utilize statistical machine learning techniques to assist trading decisions in the virtual item economy. Predictors used in the discussion were local to the game analysed, and more research can be done about finding and evaluating predictors that are non-local to all games, that might be helpful in predicting price movements of virtual items in different virtual gaming economies. (5848 words)

## IX. References


1) Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: principles and practice. OTexts.
2) Vlachos, M., Yu, P., & Castelli, V. (2005, April). On periodicity detection and structural periodic similarity. In Proceedings of the 2005 SIAM international conference on data mining (pp. 449-460). Society for Industrial and Applied Mathematics.