Predicting Virtual Markets

Ling-Ling Zhang (lszhang)

Introduction

Predicting markets (such as the stock market) has been and still is a topic of great interest and importance to many people. Unfortunately due to the extremely large number of factors which can affect the market, it is often difficult to accurately assess the value of a new object that is just being released into the world. In this project I will be considering a much smaller scale market in order to better understand how to model the various characteristics of the object in order to accurately assess it's "true value". Although it is certainly possible that none of the particular influences/ qualities of this market are reflected in the real world, perhaps insights gained from modeling this smaller market could lead to the development of more sophisticated models for real world markets.

The Market

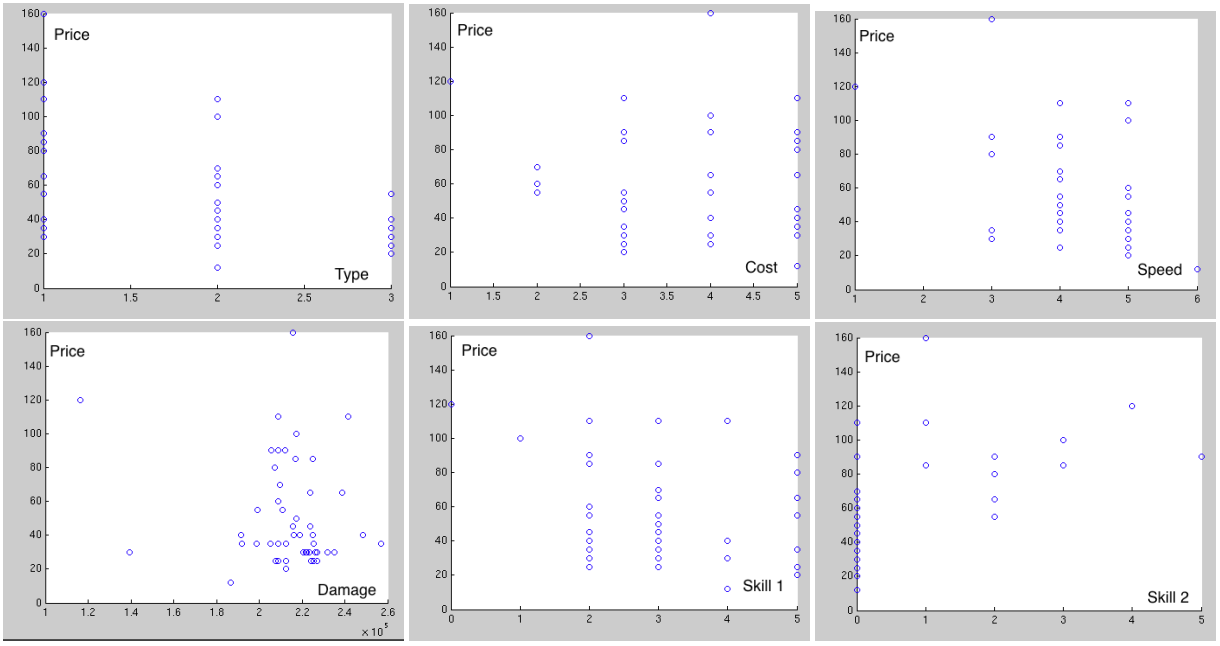
The market upon which we will be applying our model is drawn from a mobile trading card game called Fantasica. Each card has an attack type, deploy cost, attack speed, damage dealt, and up to two skills. Based on these factors the demand of these cards fluctuate and eventually players come to some consensus on the worth of each card.

Data Collection

The data set used in this project is collected from an external trade market (fantasitrade.com) over several different points in time. On this site people are able to put up units for sale at the price they wish to sell them at. We developed a python script to pull the relevant information about the unit and compiled them into discrete features.

The search was restricted to only 6* units. The reason for this decision is that other units are seen more as 'collectibles' rather than cards which are 'useful'. Also due to the high variability data for cards which are old/rare, cards which had less than 50 copies in circulation were also dropped from this set.

The following graphs show the true value of each unit with each of the different attributes we will be using.



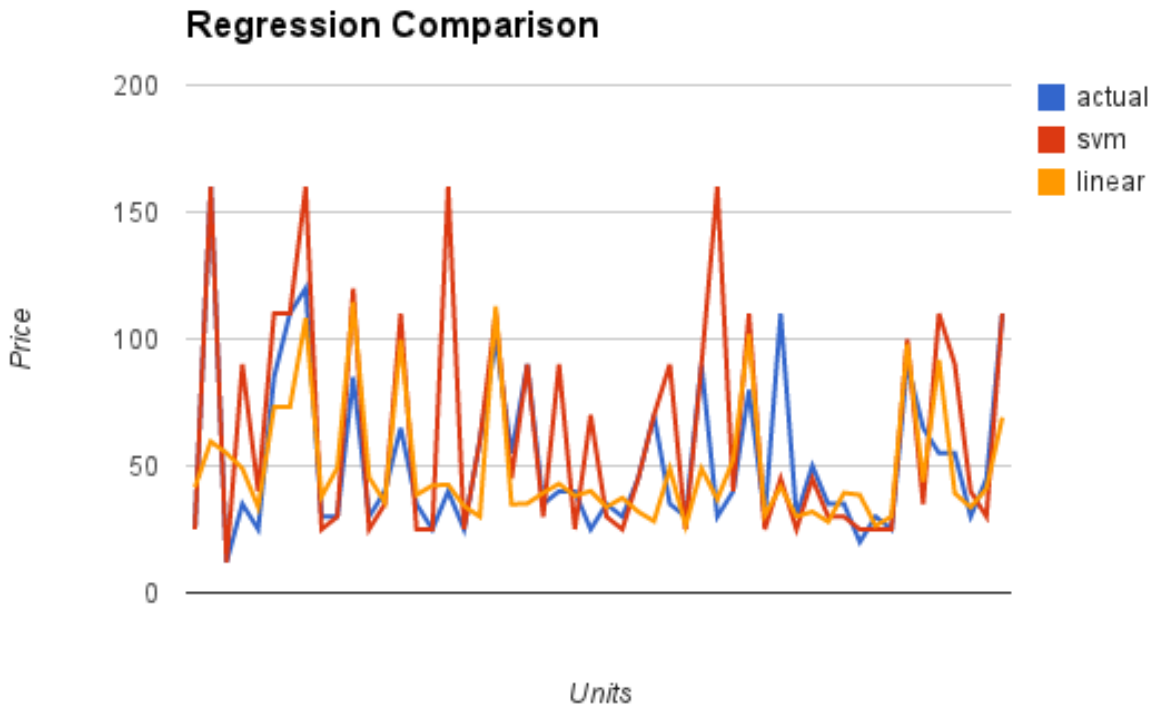
Regression

A first attempt at predicting the prices of units was to use various forms of regression. We used both the linear and logistic regression to fit the problem using the feature set of attack type, deploy cost, attack speed, damage and skills. To test the effectiveness of this model a set of cards (with known prices) was randomly divided into a training group and a testing group. To judge the performance of this model the root mean square of the error was used. The results are in the table below (the range of prices in this model was 12-160).

Table 1: Regression Results

Linear	Logistic	SVM
23.26	23.22	26.2

Although this gave us a decent model, in all three cases the model would occasionally make an extremely bad guess about the price of some unit. Interestingly enough the the svm model differed significantly from the linear and logistic regression models. The bad guesses done by the regressions were due to the fact that the models used some variables as strong predictors of price while almost ignoring other variables. But while linear and logistic seemed to favor one set of variables, svm seemed to favor a different one. Below is a chart comparing the actual prices to those predicted by linear regression and svm.



K-means clustering

In the previous part we tried to use various forms of regression to improve the prediction accuracy of the models. But one thing that seemed somewhat unsatisfactory in the regression model was that while it presented a somewhat decent prediction overall, it was wildly incorrect on several points. These "misguesses" also all appeared to be due to each model not utilizing the information it was given to the fullest in some sense.

Thus in this part we will applied the k-means clustering algorithm to the data with 12 clusters. After creating the clusters the average price of each cluster was computed and used as the price estimate for each unit in the cluster.

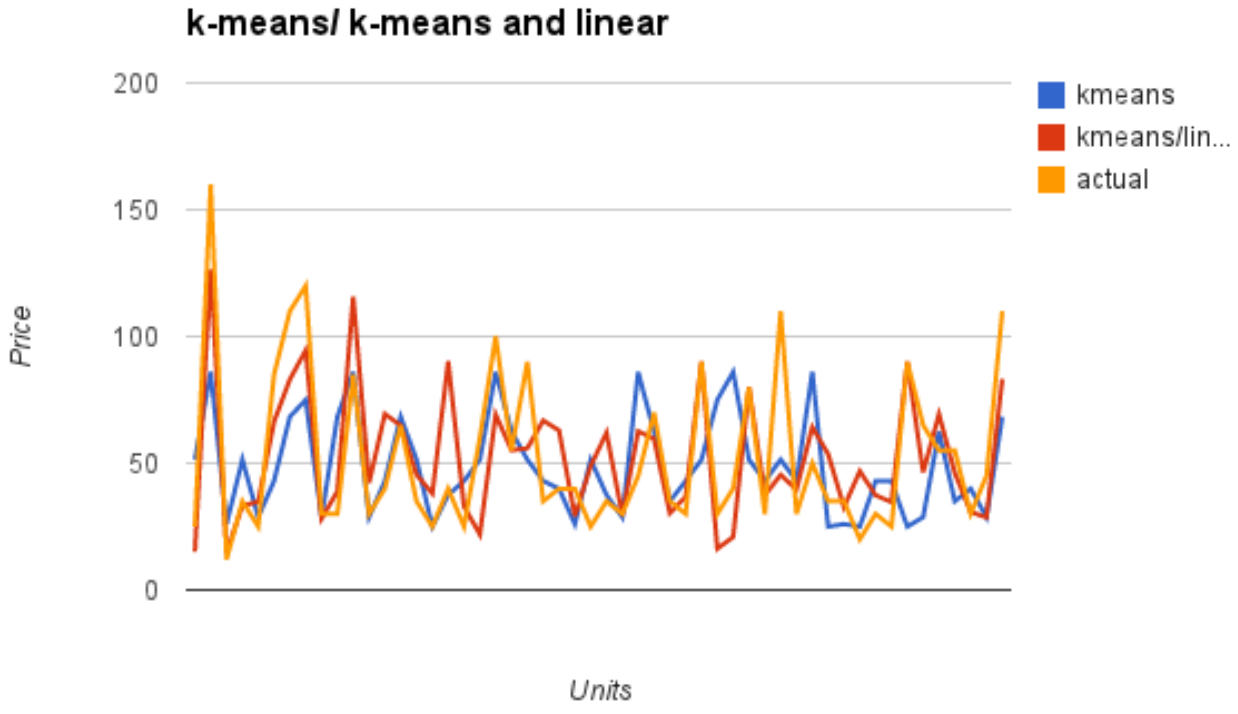
Unfortunately this approach was no more successful than our previous approaches (the error for k-means clustering was 25.15, higher than the error of linear regression). This time the error was due to the fact that units which have vastly different price ranges would end up in the same cluster. Oftentimes very small changes can make the difference between an extremely popular unit and a mediocre (or even unpopular) unit. Thus this model which put such units together was not able to cleanly separate out similar-looking but vastly different units.

Separating Categorical and Quantitative Data

In the previous parts we explored both Linear Regression as well as K-means clustering as two ways to attempt to predict the price of units based on their features. Both of the previous models were somewhat unsatisfactory. In the case of regression some attributes were seemingly ignored in favor of others. In the case of k-means clustering different units which although were very similar in attribute were actually vastly different in price, furthermore due to the nature of k-means clustering, the price classes are very granular and thus are susceptible to large error margins. Another thing that seemed vaguely wrong about the previous approaches was the fact that we were basically blindly throwing data that was both categorical and quantitative in nature at the algorithm in the hopes that it would somehow work out.

In this model we decided to separate out the categorical and quantitative data as follows. First using the categorical data we split the data into 4 clusters. (The categorical data being attack type, and the two skills). Then on each of these 4 clusters we ran linear regression.

The resulting model (although still not perfect) did a much better job of modeling the data than any of the previous approaches. The error for this combination of k-means and linear regression resulted in an error of 16.71.



Discussion

Table 2: Results

Linear	SVM	kmeans	combined
23.26	26.2	25.15	16.71

By combining k-means clustering with linear regression to separately model categorical and quantitative data we managed to significantly improve the prediction accuracy of our model. It should be noted however that this was done on a somewhat select field of items. What we did here is probably about analogous to predicting the price of a new object where the model is constrained to a specific type of object. (For example you could probably apply this model to predict the price of a new pair of headphones that's coming out based on the features that the headphones have).

In order to further improve predictions, there are several things that can be done. First the set of features was select deliberately and not at random. It is certainly possible that there are other features which play a role in the setting price which were not explored in this algorithm. Thus one improvement could be to take all of the features and run a feature selection algorithm over them to find the features which are the best predictors of the final price. Another improvement could be to change the number of clusters that are used in the first step of the combination algorithm. Although if we use too many clusters we may risk overfitting the data.

Works Cited

Fantasica Wiki http://www.fantasicawiki.com

FantasiTrade http://fantasitrade.com

Ng, Andrew, CS229 Lecture Notes