Analysis on Players Data

Garima Garg Case Study

Abstract

This is a data set of 100K players (Randomly selected). It contains data based on prior 4 weeks of player behavior. It also includes variables showing lifetime play and status - such as how recently they played, how long they have been playing for, their VIP status etc.The goal is to first predict whether a person will pay or not, and next is to cluster players and recommend campaigns. My methods includes logistic Regression with lasso penalty for prediction and k-means for clustering.

1 Introduction

Given the data set of 100K players, my intend for this study is to use various classification methods and seek for an accurate method for predicting whether a person will pay or not. Penalized Logistic regression and Random Forests have been used for prediction whereas k-means clustering is one of the best approach to deal with this type of data set. Penalty can be either ridge or Lasso, but choosing lasso penalty has an advantage that it does variable selection too.

2 Problem 1: Predicting who will pay

Briefly describe your approach to this problem and the steps you took

- Data Analysis: Since the data was sparse, first approach was to find out zero variance/near zero variance predictors, these predictors do not contribute to the analysis, so I removed them from the data. I was able to remove 24 predictors from my original data set that reduced my feature size to 194. Data was normalized so that each of the features were in range (0-1). Next, I looked for correlation among features to understand the multi collinearity in the data. Also, the data set has imbalanced classes, around 95% of players don't pay and only 5% of players pay.

- Feature Selection: After this, next approach could be to to identify all possible correlations among features, and to do feature selection based on correlation. If two of the features are highly correlated, then one is good enough for the analysis, or I could have used PCA to reduce the dimension, but the problem with PCA is identifiably(new dimensions are hard to interpret). Feature selection techniques was not used as penalized logistic regression with Lasso does feature selection. So, I skipped this step.

- Penalized logistic regression:
  Penalized Logistic regression is one of the best approach to deal with this type of data set. Penalty can be either ridge or Lasso, but choosing lasso penalty has an advantage that it does variable selection.

2.1 Basics:

– How well does your model work?
  Usually in the case of large class imbalance, a model can predict the value of the majority class for all predictions and achieve a high classification accuracy, also known as Accuracy Paradox, so to get accurate results, one should look at precision, recall and area under the ROC curve. I used 10 fold cross validation and divided the data into training, validation and test set. Table 1 shows the confusion matrix. K-fold cross validation gives you 10 estimates of prediction accuracy that can be combined into an overall measure. So I trained my model on training set, validated the tuning parameter lambda on validation set, and finally used the best lambda got from validation stage on test data to do prediction.

	Actual Pay	Actual Not Pay
Pred Not Pay	8632 (95.27%)	0 (0%)
Pred Pay	32 (0.35%)	396 (4.3%)

Table 1: Confusion Metric for Penalized Logistic Regression when cutoff value=0.5

– How do you know for sure thats how well it works?
  I used 10 fold cross validation to avoid any problem of over fitting. Also, before trying it on the entire data set, I used under sampling of majority class and tried it on the smaller data set. I took equal samples from both the groups(Pay vs Not Pay) and then used it on training and test set to check accuracy.

– What stats will you use to prove its accuracy?
  Accuracy is 99.6468%, false positive rate is 7.4%, true positive rate is 100%. Since this is the case of large class imbalance, one should also look at ROC curve, The AUC value was close to 0.80 which indicates the power of the classifier.

– What insights did you obtain from this data? For example, what features are important? Why?
  The important features based on the results of penalized logistic regression are recency, vip-age, payment-cnt-1d, purchase-success-pct-1d, spin-total-cnt-1d, spin-mixed-cnt-1d, spin-mixed-cnt-1w, spin-paid-cnt-1w.
  This makes sense because whether a person pays or not will be dependent on recency i.e. how long since he last played? So, if the player hasn't played for a month, then this factor will influence the pay. Also, pay will be related to payment-cnt-1d, purchase-success-pct-1d, spin-paid-cnt-1w.

2.2 Next steps

– What other data (If any) would have been useful?

To help predict payment, I think we can capture a variety of more features:

* **Login frequency**: Understanding a person's login frequency is an indicator of their addiction to the game. If a person logs in multiple times, chances are that they may pass free levels and pay for higher one's.
* **No. of friends who pay for the game**: Social connections play a very important role in determining one's decisions. If multiple friends have paid version of the game, it may influence other to pay for the game too.
* **Age**: Age is an important data point that can help determine whether the user will pay or not. Mobile savvy teens can pay, parents may buy games for their children, etc.
* **No. of invitations received**: Current data set contains information about invitations sent, perhaps invitations received can fetch useful information too.
* **Zipcode/Location**: It is possible that people in certain zipcode/location have higher concentration of paid users, and that may effect others in the same location too - due to social reasons, rich neighborhood etc.

– What are some other things you would have done if you had more time?

I would have used Supervised PCA to identify the key features. Supervised PCA is a dimensionaly reduction technique but supervised. With PCA, we get new features which are hard to interpret, but with supervise PCA, we can identify those features that are related to the outcome. After identifying those potential features, I would have used the prediction algorithms like partial least squares, or nearest shrunken centroid. Partial Least Squares produces scores which maximize the correlation among responses and predictors. I tried out random forests as a primarily approach on smaller sample from this data set, but it would be interesting to try it out on the data of the order of 100k.

3 Problem 2: Segmentation/Clustering

Clustering: Please create some clusters or segments for these 100K players. The aim here is simply to understand if our database contains different groupings of players

– What technique(s) did you use? Since the feature space was large, so I used PCA to first reduce the dimension, and then used k-means clustering to identify the clusters. K-means clustering is widely used because is an extremely simple and efficient algorithm, but the problem lies in the selection of K. After I found the clusters, I used decision tree to understand how features were related within a cluster.

– How many clusters do you end up with? Why is this number of clusters better than any other number?
To select K, I computed the sum of squared error (SSE) for some values of k (for example 2, 4, 6, 8, etc.). The SSE is defined as the sum of the squared distance between each member of the cluster and its centroid. If you plot k against the SSE, one will see that the error decreases as k gets larger; this is because when

the number of clusters increases, they should be smaller, so distortion is also smaller. The idea of the elbow method is to choose the k at which the SSE decreases abruptly. From the visual plot, K=3 looked good.

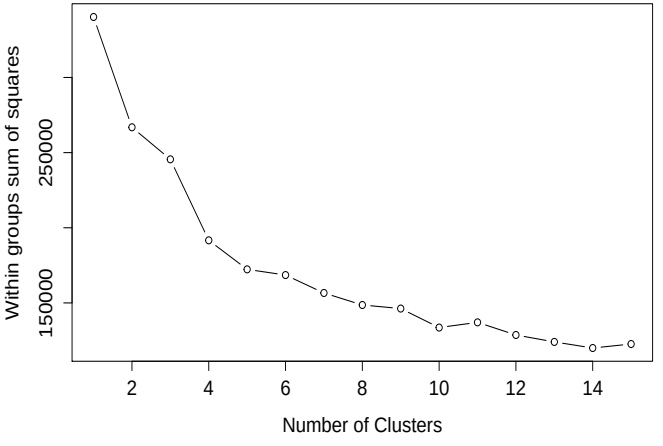

Figure 1: Identifying clusters for K-means clustering

– How would you find out if these clusters are robust over time? That is, say in 1 month from now if we were to repeat this exercise, would we find the same clusters?
For robust clustering, the members within a cluster should be very different from members of the other clusters and at the same time very similar to members within (i.e. belonging to the same group). For this, metrics such as silhouette measure, gap statistic, or CH index can be used.The average silhouette measure is how members within their own clusters are closely grouped and at the same time how loosely these members belong to neighboring clusters. A silhouette measure close to 1 implies that the members are, on average, in their correct clusters, while a silhouette close to -1 implies that the members, on average, have been assigned to the wrong clusters. Gap statistic performs a sequential hypothesis test of clustering your data for K=1,2,3,... vs a null hypothesis of random noise, which is equivalent to one cluster. Its particular strength is that it gives you a reliable indication of whether K=1, i.e. whether there are no clusters. Table 2 shows the silhouette measure summary. As you can see, the measure is close to 0.80, that is close to 1. :

cluster1-44056	cluster2-14044	cluster3-41900
0.7704116	0.7973360	0.7492009

Table 2: Cluster sizes and average silhouette widths (silhouette measure summary)

The CH index is another metric, which is essentially the ratio between the "dissimilarity" and "tightness" of an average cluster. Dissimilarity measures how a

cluster, in its own right, is different from other separate clusters, while tightness measures how members within a cluster are similar to each other.

– Please describe each of your clusters. What is the best way to visualize them? Fig 1 shows the k-means clustering(k=3) on 3 principal components. There is good separation between the three clusters when components are PC1 and PC2. Since the separation is not good when components are PC3 and PC2, so I re-run the k-means on two clusters. Fig 2 shows the two clusters on three PCs.

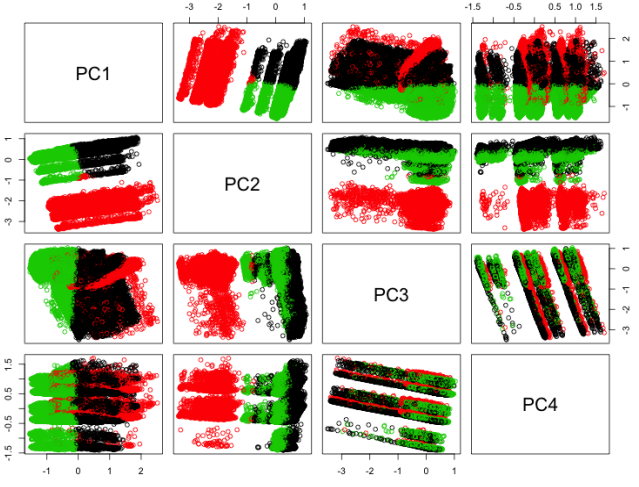

Figure 2: Three clusters on PCs using K-means clustering

– What is key to understand about players in each different cluster?
Each of the clusters gives us a broader view about the players active/inactive status. The players who don't play a lot, are in one cluster, and the players who play a lot are in another cluster. Given the time and my computing resources, I would have explored more with the association rules mining, that can directly make recommendations. Also sparse k-means clustering since the data is sparse.

3.1 Recommendations

Based on your learning from the above questions, what are some recommendations you would make to your product manager about

– Creating campaigns: what should the aim of those campaigns be?

Given the focus is on recency, the aim of the campaigns would be to minimize dormant period of the players - by non-annoying social notifications (such as "John and Mary are online and missing you, want to join them?", offering referral incentives (such as "If you play and invite 3 other friends, earn credit worth $150"), or motivating to return to the game in other ways (such as "A lot has happened to Farmville since you last left it, want to check out new changes?").

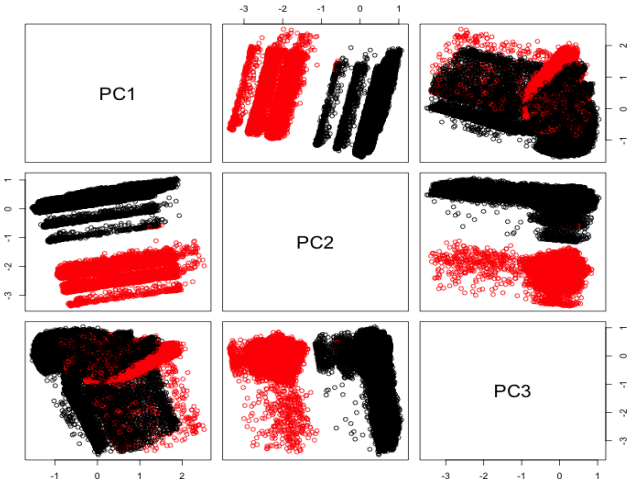

Figure 3: Three clusters on PCs using K-means clustering

There can be a lot of possibilities, to launch novel campaigns to help users get back to the game.

– How many variants would you suggest for the campaigns?

  Talking about social notifications, there can be multiple variants in terms of motivation, but we can start with 2-3 variants. The variants can depend on the message type - emotional (such as, "John and Mary are missing you"), update (such as, "John and Mary are online, you may want to join"), other. Human-computer interaction literature has diverse experiments in this space on message's effectiveness.

– How would you test the results of those campaigns?

  The results can be tested with A/B testing, by comparing metric like ratio of players going from dormant state to active state in each of the above mentioned variants of the campaign. This would be similar to comparing the metrics in the experiment vs control group.