Black Friday Sales Analysis

Ankit kumar Singh, Archit Shorey, Rohit Benny Abraham, Yuvraj Goyal

ABSTRACT

We all eagerly wait for Black Friday sales and plan ahead in order to make most out of it. Similar is the objective of a retail outlet on Black Friday. They also aspire to bring the best out of this day. The major objective of a store is to maximize the revenue on this day, by selling off a large proportion of their unsold inventory. The main challenge to achieve this objective is "What optimal prices should the store set to capture demand that maximizes revenue?" The problem we solve would help the business to get the predicted Purchase amount (or Willingness to Pay) for each product for each user. They can use this then to set optimal prices on the product (using Multinomial Model for Price Optimization or others). So, when we find Black Friday Sales Analysis data on Kaggle, it highly motivated our team to work for this interesting real-world problem for ABC Retail Store.

I. INTRODUCTION

The objective of ABC retail stores is to set optimal prices for the products on Black Friday sales day so that it can make maximum revenue. This problem is interesting and equally challenging. The study of past behavior of customers of ABC store and analyze their purchase patterns with help of machine learning to predict the purchase price for the different product can solve this potential problem and help ABC to achieve their objective.

The Black Friday Sales Analysis dataset of ABC Retail Store is available on Kaggle. The goal is to predict the purchase amount of products that customer makes. This is a large dataset of around 550K observations and 12 features. Most of the features are self-explanatory like Customer Age, Marital Status, Gender, City Category etc. and Product features like product id and product categories. But due to privacy concerns, the customer's personal data has been masked. So, understanding the relation of each feature values to target variable was a challenge.

To understand dataset, we did an exploratory analysis. First, we did univariate analysis, that helped us to understand each feature and its distribution. Further, we were interested in the relation of features with the target variable. So, we carried our bi-variate analysis. The exploratory analysis gave us a better understanding of dataset and information about a few important features. The next big challenge was the cardinality of a few important categorical variables. Using those variables was a computational expensive task but ignoring those variables led to poor model performance. This led us to feature engineering and helped us to account those variables importance using new feature introduction.

Further, we started with a naïve model to mark a baseline for our models. To understand variable importance, we used Linear Regression and Decision Tree regression. Moving on, we tried General Linear Models, Random Forest, Support Vector Regression, Gradient Boosting and Extreme Boosting techniques to predict the purchase amount of product. Then we selected the best models and optimized them further to achieve the best and reliable results.

II. RELATED WORK

Beyond the competitive aspect of this project, there was a discussion forum which we referred to get acquainted with what all difficulties previous participants have faced and their approach towards it. One common difficulty was computation on such a huge dataset using regular laptop configurations. Instead, using high-end computational power, we decided to work out this problem smartly using parallel processing of dataset using Apache's H2O ("The Open Source In-Memory, Prediction Engine for Big Data Science") package in R. Indeed, H2O offers an impressive array of machine learning algorithms. The H2O R package provides functions for building GLM, GBM, K-means, Naive Bayes, Principal Components Analysis, Principal Components Regression, Random Forests and Deep Learning (multi-layer neural net models).

III. EXPLORATORY DATA ANALYSIS

Dimensions of dataset:

Rows: 550K
Columns: 12

Due to resource constraints we took a subset of 150K observation for our analysis. We further split the data into Train and Test data with 70:30

Train Dataset: 104981 rows * 16 columns
Test Dataset: 45019 rows * 16 columns

Data Description

The dataset consists of 12 features [Table 1]

TABLE 1
Features and Description

Features	Description
User_ID	Customer ID unique to each customer
Product_ID	Product ID assigned to each unique product
Gender	Customer Gender
Age	Range of age of the customer
Occupation	Occupations of Customer from 1-20
City_Category	Category of City from A to C
Stay_In_Current_City_Years	Duration of Stay in Current City
Marital_Status	Marital Status of Customer
Product_Category_1	Product Category 1 of products from 1-18
Product_Category_2	Product Category 2 of products from 2-18
Product_Category_3	Product Category 3 of products from 3-18
Purchase	Purchase amount (Target Variable)

First, we checked the distribution of the numeric variables to detect outliers and checked missing values. Further, we were interested in understanding of categorical variables so, we did uni-variate analysis and bi-variate analysis. We observed that Male and Unmarried people tend to make more purchases in the overall population (around 75%). However, the purchase amount distribution is same across Gender, Marital Status, Occupation, and City Category (as shown below for Gender). So, we felt these attributes does not explain much about the purchase pattern of the customer.

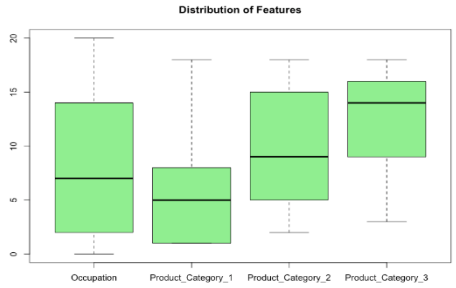

Fig 1. Distribution of Features

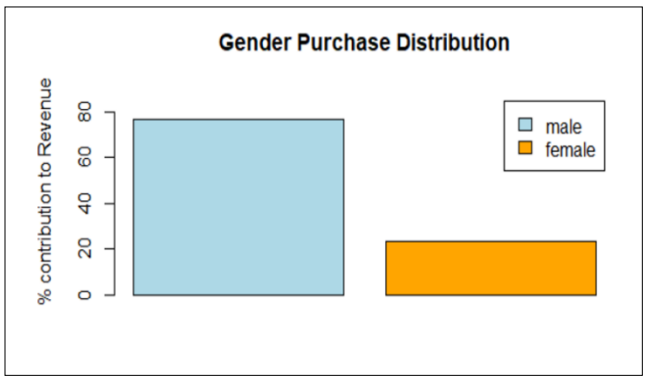

Fig. 2. Gender Purchase Distribution
(Percentage Contribution in overall revenue)

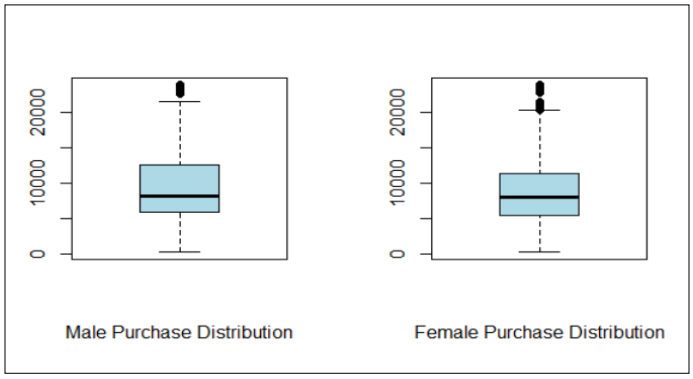

Fig. 3. Gender Purchase Distribution

Product Category was one of the most interesting variables to study. There were 18 unique products categories, but a single product can belong to at most three different categories. For example, Amazon Alexa can belong to Music Category, Entertainment Category and Electronics Category. Also, there were products which belonged to only one category or two categories. The column Product Category 2 and 3 were NA for those categories. So, understanding the right replacement of NA was very crucial for us. Either it could refer to Product Category 1(or the higher category in the hierarchy) or it could be given a 0.

The last variable to analyze was the target variable, Purchase Amount. The distribution on Purchase was fairly normal. The minimum purchase amount for a product was $185 and the maximum amount was $23955.

IV. FEATURE ENGINEERING

Customer and Product Id are two important categorical features in the dataset. But due to the high cardinality of 5k and 3k Levels,

including those two features in the model lead to great computation resources and time. While ignoring the two features the model is performing poorly. So, we included the count of customer id for each unique customer id. This helps us to explain the number of purchases the user has made which is can help explain some properties of the user. We also used Median, Quartile 1 and Quartile 3 values of purchase amount corresponding to each product id. We thought that if a product may be sold to different users for different amounts, the distribution of how high, low or the median amount paid for that product help us predict the purchase amount better. This step helped our models greatly in accounting the variability and increase accuracy on predicted purchase amounts.

V. Models and Methods

1. Naïve Model:

The Naïve model is used to serve as a baseline model. The mean of each product is taken as its predicted amount for the users.
Best RMSE (without feature engineering): 2964

2. Generalized Linear Models (GLM):

GLM is a flexible generalization of ordinary linear regression that allows for response variables that have error distribution models other than a normal distribution. The GLM generalizes linear regression by allowing the linear model to be related to the response variable via a link function and by allowing the magnitude of the variance of each measurement to be a function of its predicted value.
Best RMSE (with feature engineering): 2703

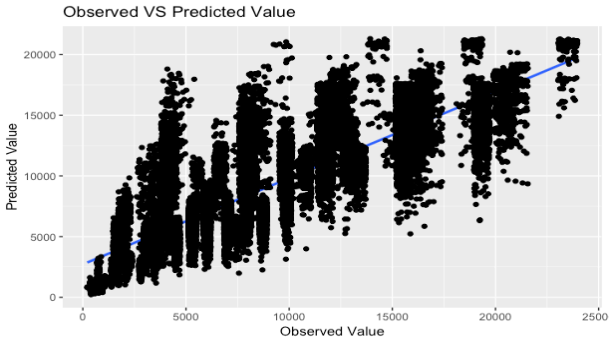

Fig. 4. Residual plot for the GLM model (with predicted values)

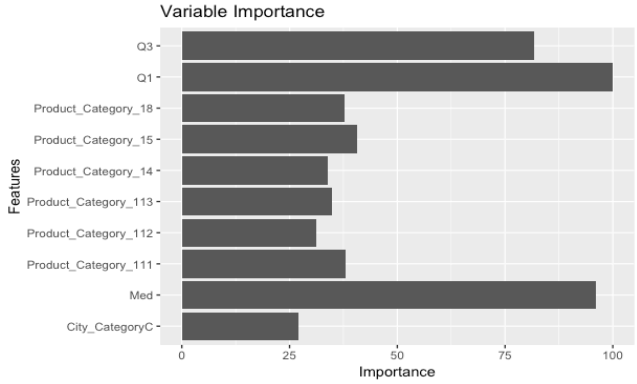

Fig. 5 Variable Importance from the GLM model

3. Random Forest:

Random Forests [2] are an ensemble learning method for classification and regression analysis. The principle of RF is to construct multiple decision trees and return the mode of these classes (classification) or mean prediction (regression) of the individual trees. In contrast to classical decision trees, RF is more robust to overfitting.

Each model was trained using a 3-fold and 5-fold cross-validation process, with the total no. of trees being 50.
Best RMSE (with feature engineering): 2718

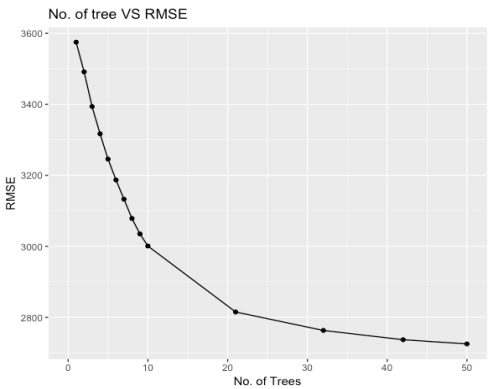

Fig. 6. RMSE plot for random forest model

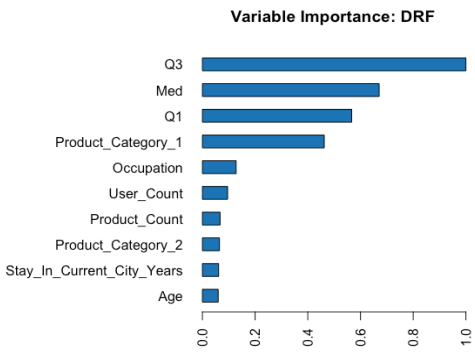

Fig. 7. Variable Importance from the Random Forest model

4. Support Vector Machine:

Support vector machine is a class of supervised learning models for regression and classification analysis. It separates the input feature space in hyperplanes. It can handle linear and nonlinear separation of the feature space using kernels. Support vector machine offers good performance but can be computational complex with high dimensional input space features and nonlinear kernels. The radial kernel was used for the model evaluation with parameters being cost =1 and epsilon = 0.1
Best RMSE (with feature engineering): 2746

5. Gradient Boosting:

Gradient boosting is a family of supervised machine learning techniques for regression and classification problems that are highly customizable. Gradient boosting machine produces a prediction which is an ensemble of weak prediction models (e.g. decision trees). Using boosting techniques and a meta-algorithm for reducing bias, GBM can convert a set of weak predictors to strong ones.

Best RMSE (with feature engineering): 2691

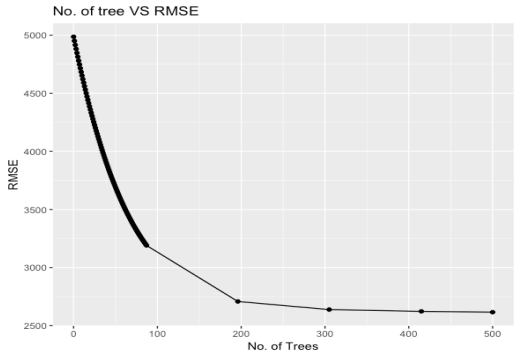

Fig. 8. RMSE plot for the GBM

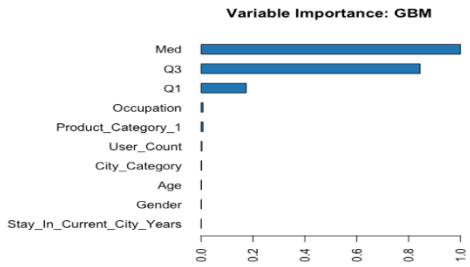

Fig. 9 Variable Importance from the GBM

6. Extreme Gradient Boosting (XG Boost):

XG Boost is a scalable and accurate implementation of gradient boosting machines. The implementation of XG Boost offers several advanced features for model tuning, computing environments and algorithm enhancement. It can perform the three main forms of gradient boosting (Gradient Boosting (GB), Stochastic GB and Regularized GB) and it is robust enough to support fine-tuning and addition of regularization parameters.
Best RMSE (with feature engineering): 2700

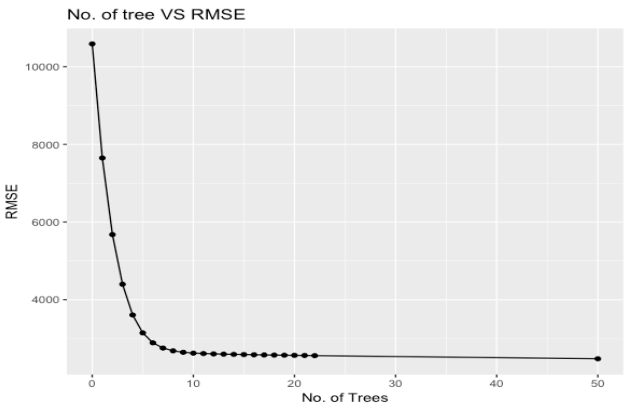

Fig. 10. RMSE plot for the XG Boost model

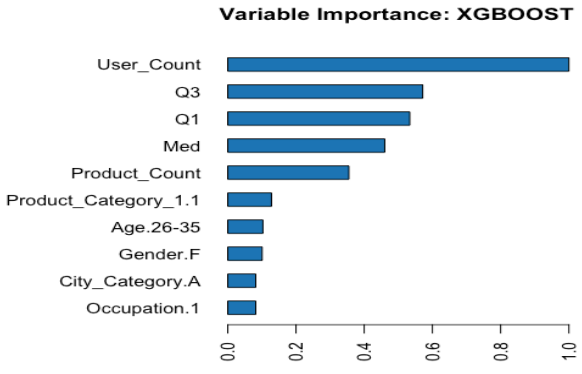

Fig. 11. Variable Importance from the XG Boost model

VI. Experimental Results

Following are the RMSE values of the Models:

TABLE 2

Model's RMSE Values

	Without Feature Engineering		With Feature Engineering	
Model	**Train**	**Test**	**Train**	**Test**
Naïve Model	3002	2964	-	-
GLM	3002	2968	2641	2703
Support Vector	2821	2901	2656	2746
Random Forest	2987	2959	2721	2718
Gradient Boosting	3005	2963	2594	2691
XGBoost	2871	2908	2400	2700

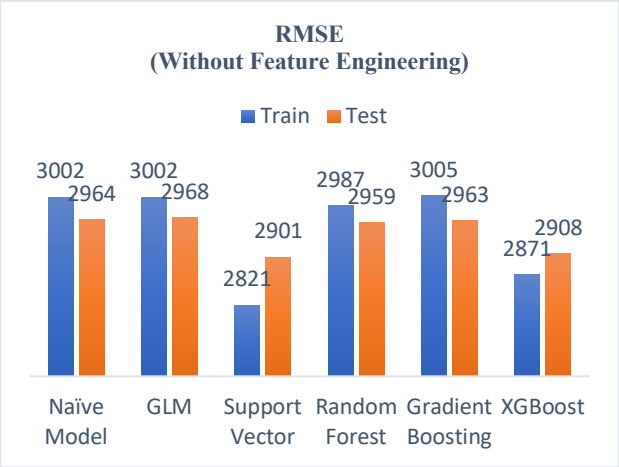

Fig 12. Plot for RMSE's (without feature engineering)

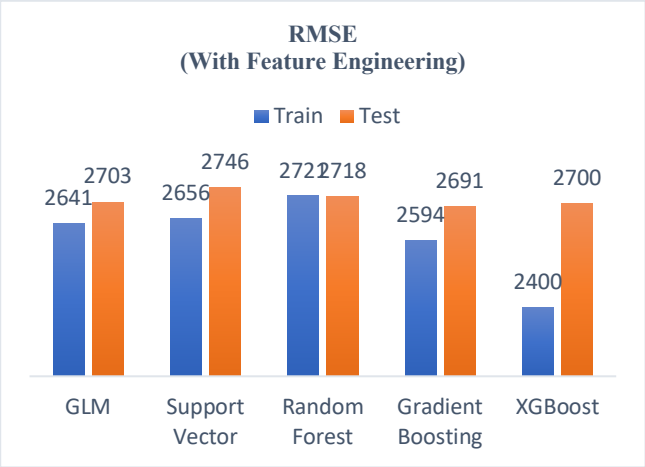

Fig 13. Plot for RMSE's (with feature engineering)

VII. Discussion

To the best of our knowledge, this project provided us with some unique challenging tasks which helped us to get ourselves involve in the world of data science. Prediction capability of models was hugely dependent on the type of features and size of datasets. Feature Engineering was the deciding features of this project and the models which was able to capture the dataset's non-linearity were performing well. Although some models clearly perform better or worse than other models on average, there is significant variability across the RMSE metric. Without feature engineering, even the best models sometimes perform poorly. Looking forward, we think we could bring down the RMSE further by bringing in more properties of User Id in the model by thinking about more feature engineering on it. One example would be to understand that how high or low the user pays for a product compared to its median/mean amount. This would explain his net utility of the product (can be expressed as a ratio) and can prove to be a good predictor.

VIII. Conclusion

Concluding, we learned a great deal about the dataset that we were given for this competition and the subset of the test data which we decided to use for the project. We had a few questions at the start of this project, of which the most important was: "If none of the powerful models are working, what changes should we do to lower down our RMSE". The answer to this question was "Feature Engineering", the most important and distinguishing factor. We learned to leverage the art of feature engineering and how it can make all the difference in the performance of the model. The next hindrance we faced was computation on such a huge dataset using standard

configuration machines. Apache's H2O package was the real savior and helped to increase our efficiency by lowering down the model computation time. In terms of model evaluation, we did use a fair evaluation strategy using cross-validation for regularization hyperparameters and GLM was selected to be the winner. GLM captured the variance between training and test dataset by maintaining good bias-variance trade-off and giving high performance on RMSE metric. Overall, while doing this project, effectively implementing advanced analytics practices, we learnt about the importance of memory optimization, understanding the dataset thoroughly and feature engineering. The message for us was that in the real world the model optimization and evaluation can only be done once all these steps have been thoroughly worked on before.

IX. REFERENCE

[1] https://en.wikipedia.org/wiki/Generalized_linear_model

[2] Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning [Internet]. New York, NY: Springer New York; 2009. Available: http://link.springer.com/10.1007/978-0-387-84858-7

[3]https://www.kdnuggets.com/2017/10/xgboost-top-machine-learning-method-kaggle-explained.html

[4] Rennie JD, Shih L, Teevan J, Karger D. Tackling the poor assumptions of naive bayes text classifiers. International Conference on Machine Learning. 2003. p. 616. Available: http://www.aaai.org/Papers/ ICML/2003/ICML03-081.pdf

[5] Analytics Vidhya Discussion Forum - https://discuss.analyticsvidhya.com/t/black-friday-data-hack-reveal-your-approach/5986/8

[6] Understanding feature engineering in R - https://blogs.msdn.microsoft.com/microsoftrservertigerteam/2017/03/23/feature-engineering-using-r/

[7] Diving into H2o - https://www.r-bloggers.com/diving-into-h2o/

[8] H2o package reference in R - https://www.rdocumentation.org/packages/h2o/versions/3.20.0.8