Statistical Machine Learning – Fall 2019

Purchase Capacity Prediction based on User demographics and Product information

Keerthiraj Nagaraj

Electrical and Computer Engineering, University of Florida

ABSTRACT

In this project, I have used Black Friday sales data to predict the amount a given user can spend on a product of a specific category based on his/her demographic information. The target response "Purchase Amount" is a continuous variable, so regression models such as Multiple Linear Regression, Decision Trees, Random forest and XGBoost are trained to predict it based on the input features. K-fold cross validation is used for optimal hyperparameter selection and RMSE is used as the performance metric to test the regression models. The target response was also converted into binary form and classification models are trained to predict purchase level. Finally, promising results are presented in the form of ROC curves.

1. INTRODUCTION AND MOTIVATION

Retail companies collect large amount of data about their consumers including demographic information such as gender, location, age, marital status etc. and their purchase history for products of various categories in the hopes of understanding the customer purchase behavior so that they can offer targeted better deals or set optimal prices for products to maximize profits. These data driven decisions help retailers to provide better service for their consumers and increase their revenue in the process. My motivation to select this data and problem is to familiarize myself on how to work with large scale data for a regression problem which has many categorical predictors. This project will allow me to use the concepts learnt in the class to solve a real-world problem.

2. BACKGROUND AND DATA

The dataset is made available by Analytics Vidhya platform for their Data hack online hackathon and has 550,069 samples ('n') and 12 predictors ('p'). The data contains mostly categorical predictors, so I will be creating additional features during feature engineering. The data contains input features namely User ID, Product ID (3631 unique values), Gender (Male, Female), Age (6 levels), Occupation (20 different levels), City category (3 levels), Stay in current city in years (4 levels), Marital status (Yes, No), Product Category (20 levels), and target feature namely "Purchase amount" which is a continuous variable.

2.1 Exploratory Data Analysis

The first important step of any data science/machine learning project is identifying interesting details about the data, such as variable types, variable levels (if categorical), distributions (if continuous), correlation, variance of target for different input features etc. Exploratory data analysis through data visualization helps us understand data in a much better way and is useful in explaining the results of machine learning models.

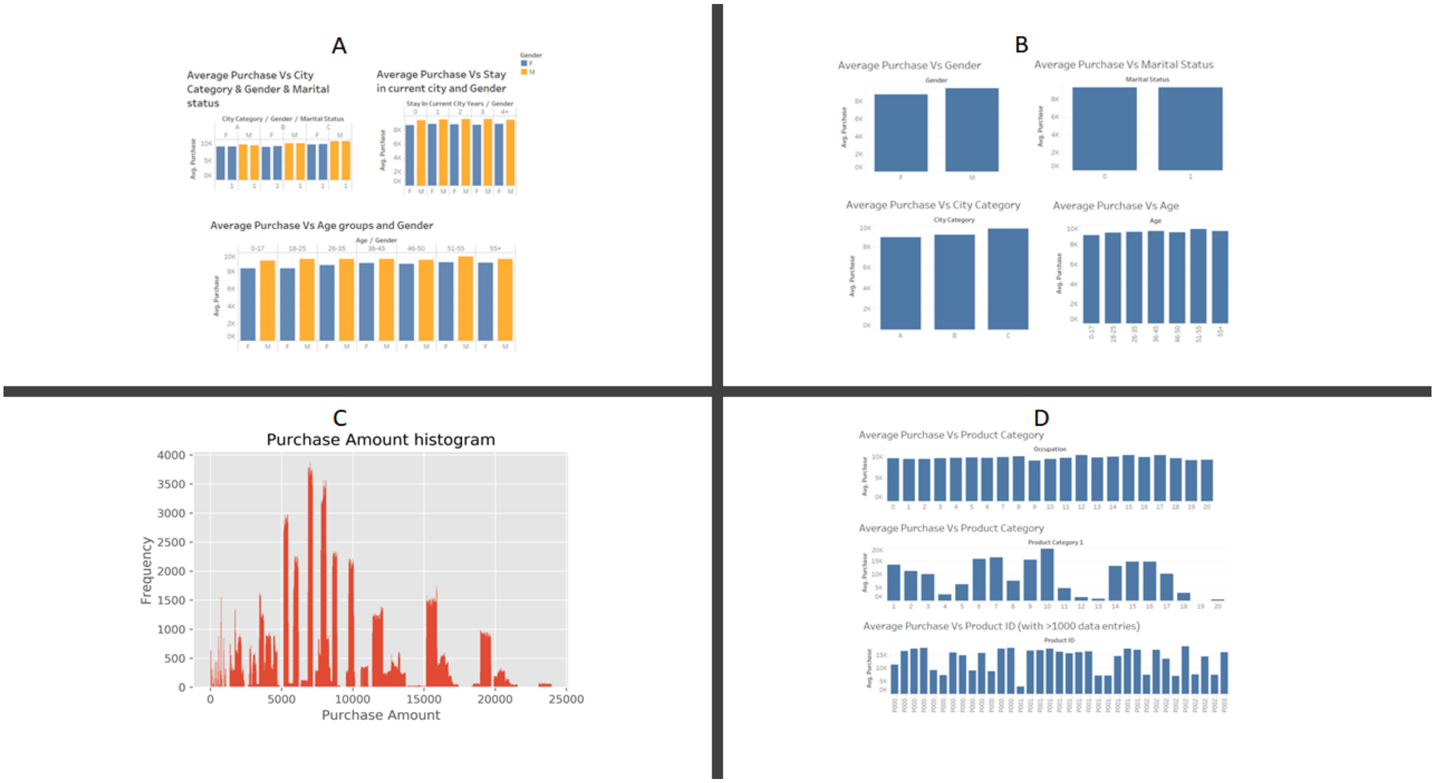

Figure 1 – Exploratory data analysis

Figure 1.c shows the frequency distribution of target variable, Figures 1.b (Vs Gender, Marital status, City category, Age) and 1.d (Vs Occupation, Product Category, Product ID) show the variation of Average Purchase amount for various input features, and Figure 1.a (Vs City + Marital status + Gender, Stay in current city + Gender, Age + Gender) shows the variation of Average Purchase amount with respect to multiple input variables. From

Figure 1.a, 1.b and 1.d, we can observe that the average of target response doesn't vary much expect with respect to Product Category, Product ID, Occupation and City category. It would be interesting to see if machine learning models will be able to capture this information.

2.1 Feature Engineering and Data Preprocessing

The data has mostly categorical variables which makes it difficult to directly apply machine learning models on the raw features, so data preprocessing is needed. Product ID has more than 3000 levels, so in order to eliminate the need to create large number of dummy variables, I counted the number of samples in each Product ID and used it as a numeric feature instead of Product ID. Similarly, I created two additional features named User Count and Product Category Count, which counts the number of samples for unique values of respective variables. I created dummy variables for Stay in current city, converted the Age groups into numeric features, Gender and Marital status into binary features and Occupation as numeric feature. The data finally had 14 input features and 1 target feature ready for model training and testing.

3. METHODOLOGY

The task of predicting Purchase Amount was performed using machine learning models, specifically regression models namely Multiple Linear Regression (baseline), Decision Trees, Random Forests (Ensemble/Bagging approach) and Extreme Gradient Boosting with decision trees (XGBoost – Boosting approach). I also used the previously mentioned tree-based techniques to perform binary classification tasks (predicting the purchase level of a given user for a given product ID – 'low' or 'high') with Logistic Regression as the baseline for performance.

Multiple Linear Regression (MLR): Uses least squares fit to find the weights for each input response that reduces the mean squared error between target and predicted values.

Decision Trees (DT): Decision Tree is a simple yet efficient machine learning technique. Decision Tree creates a series of well-formed questions/conditions about the feature values for each sample. The questions/conditions are formed when the model is trained using labelled training data. Each time it receives an answer for a question from the training sample, it will follow up with other questions/conditions until the tree can predict a target value/assign a label to that sample. The model learns which kind of questions/conditions lead to results that best fit the training data. It has hyper-parameters such as maximum depth of tree, minimum leaf size, split criteria etc.

Random Forests (RF): Random Forest uses an ensemble of decision trees and a subset of input features over multiple iterations to find the best combination of decision trees during model training to reduce overfitting. It has hyperparameters of decision trees along with number of estimators and number of features for each tree. Random Forest is an example for Bagging approach as it uses a bag (collection) of trees to find optimal model.

XGBoost (XGB): Extreme Gradient Boosting with decision trees is an example for boosting technique which uses decision trees as the base estimator. Boosting technique gives more importance for samples which have bad predictions and tries to change model parameters to address them over iterations, which in turn improve the overall model performance. The XGBoost model also has similar hyper parameters as decision trees along with number of estimators, booster type, subsample ratio etc.

Logistic Regression (LOG): Logistic regression uses logistic function (sigmoid function) to model a binary target but can also be extended to multi-class classification problem. It is commonly used as baseline for classification analysis due to its simplicity and ability to perform well in many cases. It has hyperparameters such as penalty type (L1- Lasso, L2-Ridge), fit intercept, solver, and class weight etc

The advantage of using Decision Trees, Random Forests and XGBoost models is that they can be used for both regression and classification, while Multiple Linear Regression and Logistic Regression are used as baselines.

4. PERFORMANCE METRICS

I have used RMSE and R2-score for regression models and, F1-score and ROC curves for classification models as performance metrics for testing the prediction accuracy of trained models on test data. I also used Grid Search Cross Validation (K-folds) for optimal hyper parameter selection for all the machine learning models. Data is split into training and testing sets, and testing data is used for performance assessment of trained models.

Root Mean Squared Error (RMSE): RMSE, as the name suggests is the squared root of mean of all the errors (difference between target variable and predicted value). Lower the value of test RMSE, better is the model.

R2-score (R-squared): R2-score/Coefficient of Determination is the amount/proportion of variance in the target variable that is predictable from the input features. In regression problems, it is a measure of how well the predictions approximate to real data points. Values typically vary between 0 and 1, higher the better.

F1-score: It is the harmonic mean of precision and recall of a classification model. F1-score is a better metric than accuracy, especially in the cases where class sizes are imbalanced, as it is high only when both precision and recall are high, which means both false positives and false negatives are low.

ROC curves: Receiver Operating Characteristics (ROC) curves show True Positive Rate (0-1 range) versus False Positive Rate (0-1 range) for a given classification model using the decision scores/prediction probabilities of testing samples and ground truth of the samples. The area under the ROC curve is a good indicator of the model performance, larger the area (largest possible 1) better is the model performance as it shows that the model can achieve high True Positive Rate for low False Positive Rates.

Grid Search CV: Grid search K-fold cross validation is a way to train models for all combinations of a set of hyperparameters values and choose the model which performs well during K-fold cross validation. In K-fold cross validation, data is initially split into 'k' parts, in each step, 'k-1' parts are used for training the model and the remaining samples are used for testing the model performance. This approach helps us to select optimal hyper parameters that results in models which can generalize well for unseen data as well.

5. RESULTS AND DISCUSSIONS

In this section, I will provide details about data split, model parameters from cross validation, performance metric values for regression and classification tasks and relevant discussion about each figure. The analysis was mainly carried out in Python environment using Scikit-learn, NumPy, Pandas and Matplotlib libraries. The data which has 550,069 samples is randomly split into two parts namely training set with 80% samples (440054) which is used during model training and hyper parameter selection, and testing set with 20% samples (110014). Multiple Linear Regression only takes normalize as an input and it was set TRUE for model training. Grid Search K-fold cross validation resulted in the selection of following hyperparameter values for each of the machine learning model used in the analysis. The value of k in K-fold was selected as 3.

Decision Trees parameters: maximum tree depth = 20, minimum leaf samples = 50, criterion = MSE for regression, Gini for classification, splitter = best at each iteration, all others set to default from scikit-learn library.

Random Forest parameters: maximum tree depth = 20, number of estimators = 10 (anything more resulted in overfitting), criterion = MSE for regression, Gini for classification, OOB-score = TRUE, all others set to default.

XGBoost: maximum depth = 10, number of estimators = 1000, tree method = auto, booster = gbtree, objective = reg:squarederror for regression and binary:logistic for classification, learning rate = 0.05, all others set to default.

Logistic Regression parameters: solver = sag, class weights = balanced (adjusted according to actual class size), penalty = l2 – ridge, fit intercept = TRUE, maximum iteration = 200, all others set to default.

5.1 Regression Model Results

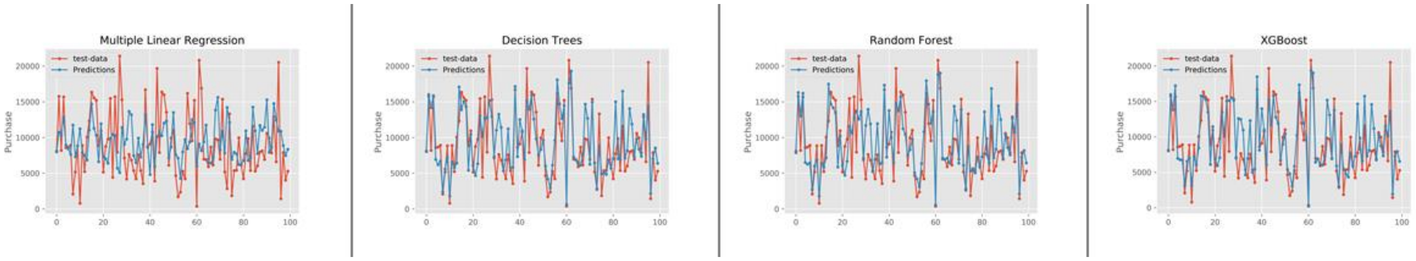

Figure 2. Prediction Vs Target (Purchase amount) for first 100 testing samples.

In Figure 2, we can observe that MLR has worst performance and XGBoost outperforms others (although the plot only shows first 100 testing samples). It is important to know how the models performed for the entire testing dataset.

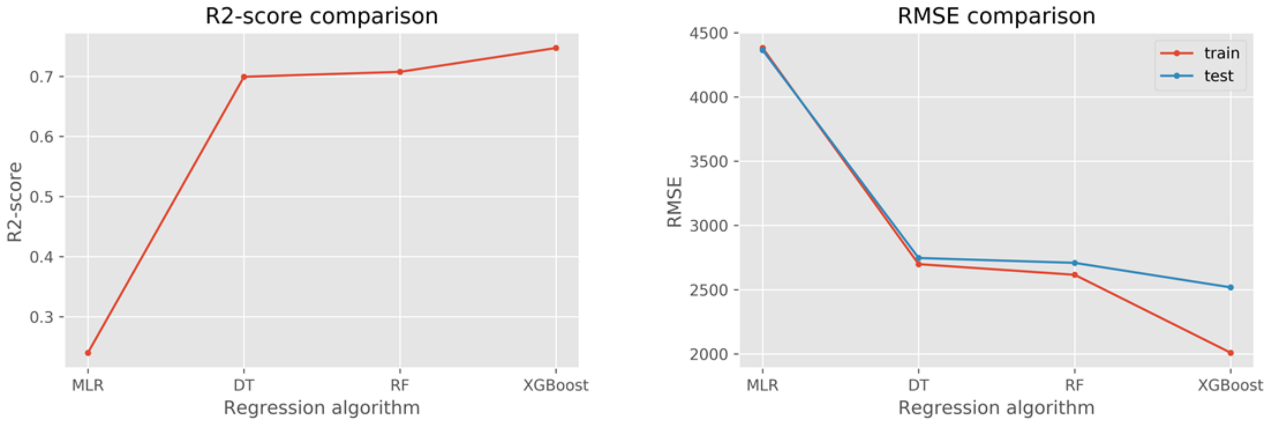

Figure 3. R2-score and RMSE values for various regression models.

In Figure3, we can notice that the performance improves (high R2-score and low RMSE value) in the following order: Multiple Linear regression, Decision tree regressor, Random forest regressor and XGBoost regressor. XGBoost has slightly overfitted as the difference between training and testing high compared to others but still is best model out of 4 as it has much lower RMSE and higher R2-score than the other 3.

5.2 Classification Model Results

To perform classification task, I created an additional variable called as "Purchase level" based on the value of original target response "Purchase amount", which is a categorical variable of two levels – high and low. High purchase level indicates that a given user falls in a group which has high purchase amount for the given product and similarly Low purchase level indicates that the user falls in a group which has low purchase amount for the given product. This categorical variable was converted into a binary variable by encoding Low and High as 0 and 1 respectively. I used 75-percentile value of Purchase amount to group samples into low and high levels, meaning the 75% of all the samples ordered by the purchase amount belong to level low and 25% belong to high level, thereby class 0 has 75% number of samples and class 1 has 25% number of samples in the classification data.

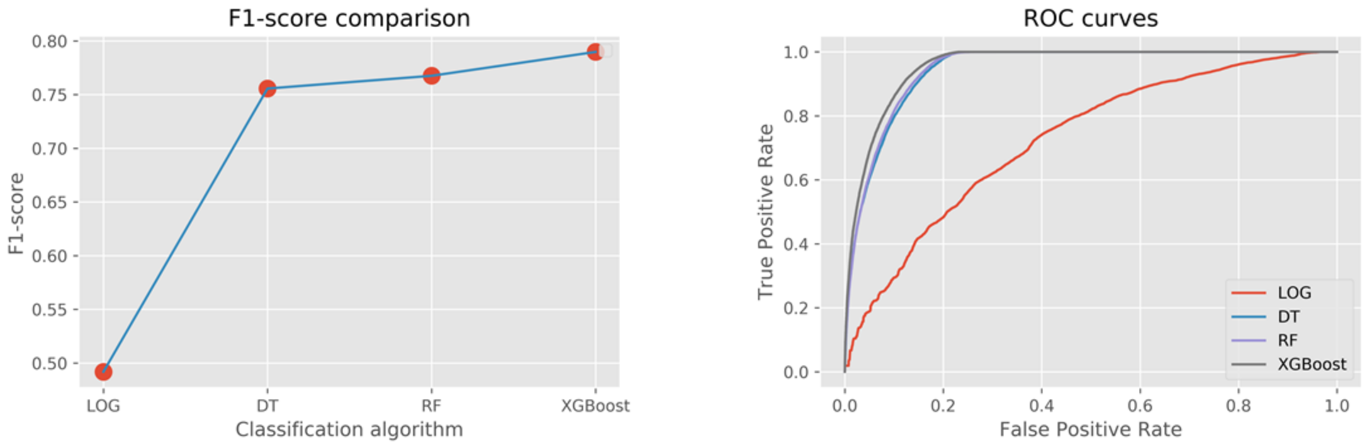

Figure 4. F1-score and Receiver Operating Characteristics (ROC) curves for classification models.

In Figure 4, we can notice that XGBoost has the highest F1-score and largest Area Under Curve (AUC) compared to other classification models, and Logistic Regression has worst performance as expected due to its simplicity.

CONCLUSIONS

In this project, I have used tree based machine learning models such as Decision Trees, Random Forest (ensemble approach) and XGBoost (boosting approach) along with Multiple Linear regression and Logistic Regression as the baseline models to successfully perform Regression and Classification tasks on Black Friday sales data. The objective was to train and test machine learning models to predict the purchase amount of a user given his/her demographic information to a product given its category and popularity. I performed exploratory data analysis to understand the relation of target variable with various combinations of input features, engineered additional relevant features, selected optimal hyper parameters through grid search k-fold cross validation and tested the model performance using metrics like R2-score and RMSE for regression models, and F1-score and ROC curves for classification models. XGBoost (Extreme Gradient Boosting) gave best results in both regression and classification tasks and the experimental results support these statements.

APPENDIX: It contains all the images in high quality and additional figures which show importance of various features as obtained from tree-based machine learning models.