PROJECT REPORT

REGRESSION ANALYSIS ON BLACK FRIDAY DATASET

Harshida, Jegadeesh Chandar

PREDICTIVE MODEL AND DATA MINING
BDA104

Regression analysis on Black Friday dataset

Aim: To learn different types of regression analysis on Black Friday dataset and choose the model that best fits

Dataset: The Black Friday dataset contains 537577 rows and 12 columns, and the column 'Purchase' is the dependent (target) variable.

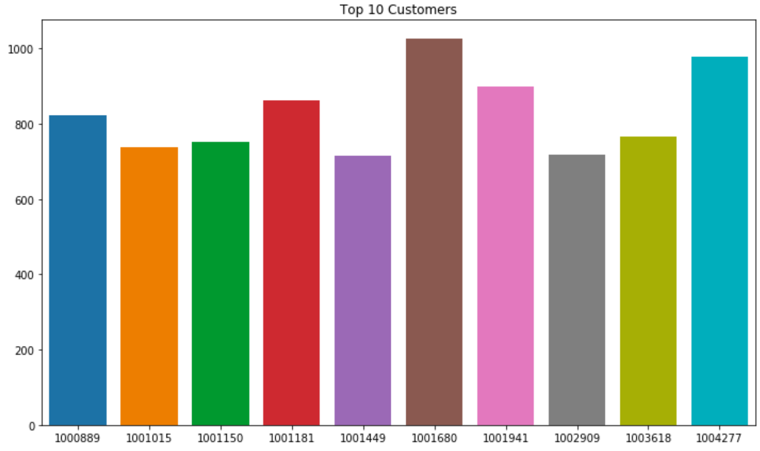
Process Flow:

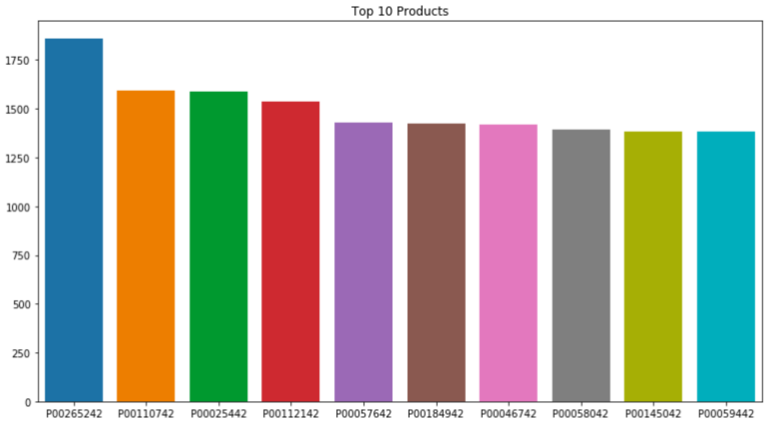
Data Wrangling/cleansing

- **Replacing Nan's**- The data is cleansed by replacing the Nan values. Product_category_2 and Product_category_3 are the columns that contain null values. Since these two columns have levels as values (Product_category_2 - 2 to 18 and Product_category_3 – 3 to 18) they can either be replaced with maximum value of the respective columns or replaced with zeros. Both were tried and the latter gave better results.
- **Categorical to numerical values**-There are totally 4 columns 'Gender', 'Age', 'City_category', 'Stay_in_current_city_years' which have to be converted to numerical. **Label encoder method and get dummies method were used**. Label encoder method was used for the columns which had more than two levels and get dummies method for rest of them

Data Analysis

Determined the top 10 customers and top 10 products based on the number of purchases made

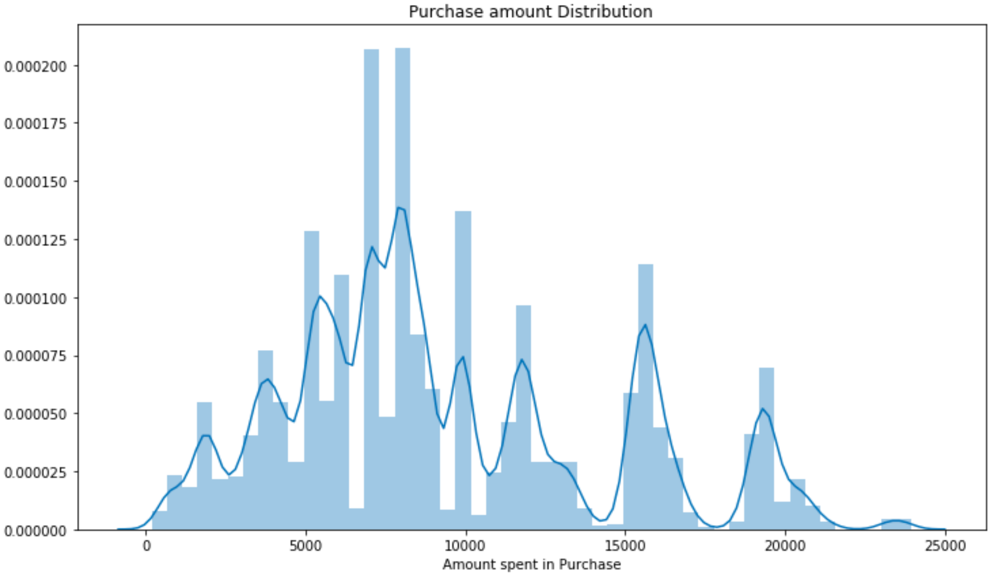




Customer with User ID **1000889** has made the highest number of purchase and product ID **P00265242** is bought very frequently

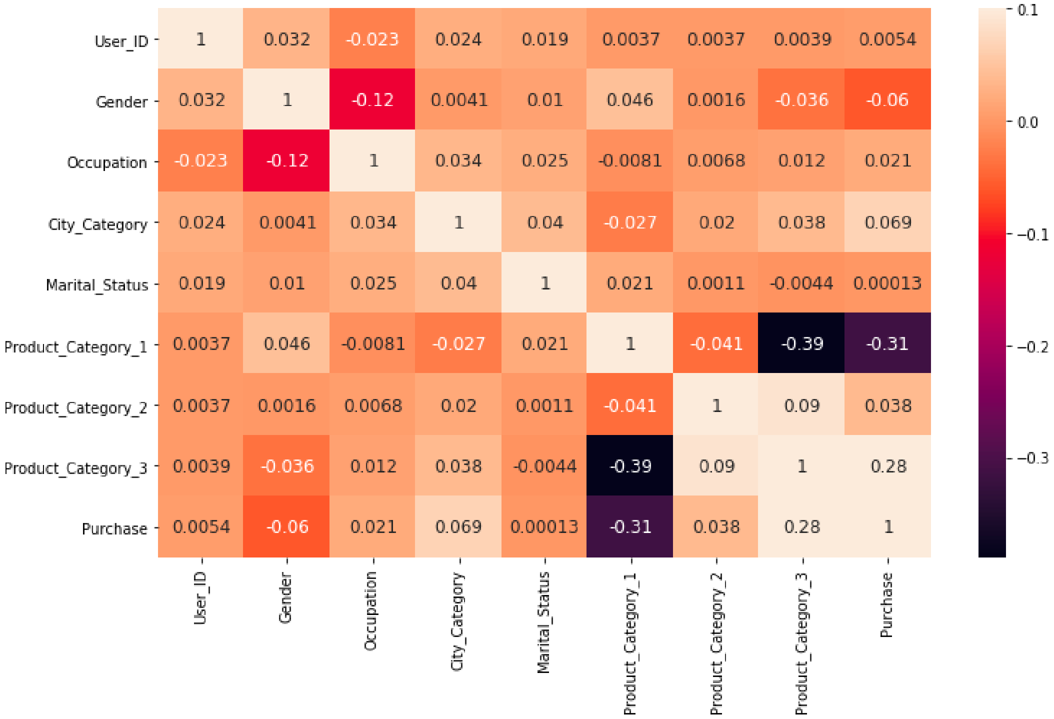
Distribution of the target variable

It is important to visualize the distribution of the target variable to know if there are any outliers and to see the type of distribution



Skewness=0.62427
Kurtosis=0.34312
We can visibly see that the distribution is right skewed, and this can be confirmed by the skewness which is positive. Similarly, there are a smaller number of outliers which is shown by the negative sign in the kurtosis. Usually the kurtosis value of the normal distribution is 3.

- **Correlation plot** This is to determine the effect of each independent variable on the target variable

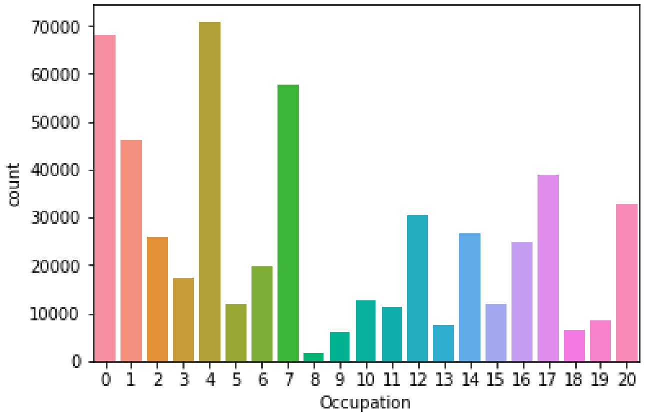


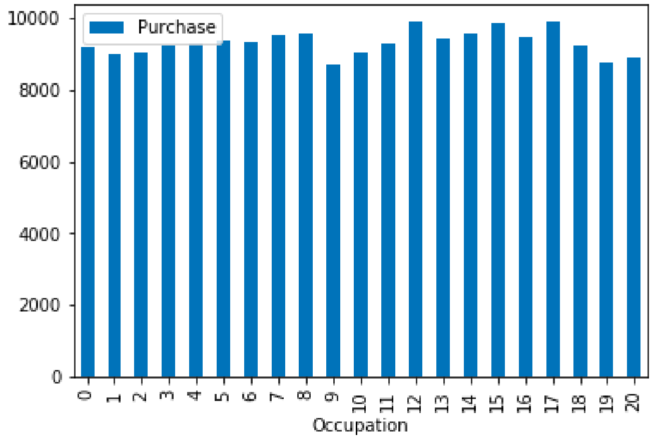
This shows that top 4 variables that have impact on the target variable are

1.Product_category_3 which is 0.28
2.City_category which is 0.069
3.Product_category_2 which is 0.038
4.Occupation which is 0.021

- Product_category_1 has a negative correlation but the strength of correlation is high (-0.31).
- Similarly, there is no multicollinearity between the predictor variables as the coefficients are not close to 1

Analysing variables that have high correlation coefficient





In fig.1 We see that people in occupation 0,4,7 have made highest number of purchase whereas in fig.2 we see that people in occupation 12,15,17 have spent the highest average amount on purchase.

Feature Selection

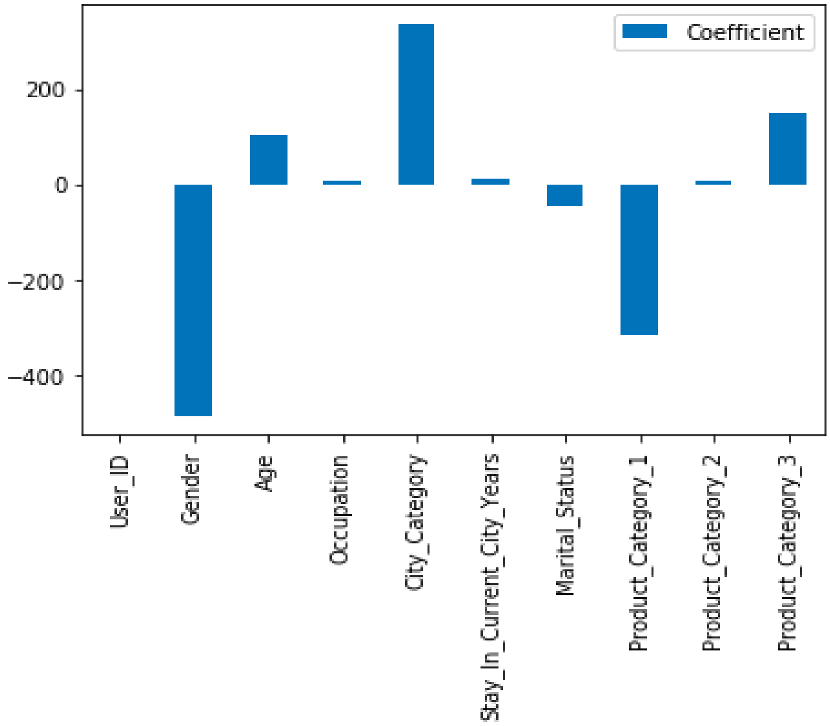
Variance Threshold- This is filter method of feature selection where the dependent variables that do not have much difference in the variance will be eliminated. The threshold value was set to 0.8 which means the variable that do not have change in variance for more that 80% will be eliminated. So, the columns **'Gender', 'City_Category', 'Marital_Status' were eliminated** as they have the values either 0 or 1. When the threshold was reduced to 0.4 'Gender' and 'City_Category' were eliminated.

Recursive Feature Elimination-This is wrapper method of feature selection. This method recursively performs the feature selection procedure with different sets of features until the best fit is obtained. This procedure is repeated until the defined set of features is attained. 5 features were specified, and the following features were selected.

‘Gender', 'Age', 'City_Category', 'Product_Category_1'and
'Product_Category_3' were selected

Since variance threshold eliminated ‘Gender’ and ‘City_Category’ and RFE selected ‘Gender’ and ‘City_Category’ there is a contradiction. Hence, we go for the third method ‘Correlation coefficient’

Correlation coefficient-This is a method in which the train and test data is fit in any of the models and their correlation coefficients are determined. This gives us the measure of how strongly the independent variables are bound to the target variable.



Based on these three feature selection methods it is concluded that **Occupation, Product_category_2, Marital_Status and User_ID will be dropped** before building the models to get better performance.

After dropping features the R2 score was checked with Decision tree regressor. R2 score improved by 13% after feature selection.

Building regression models

- **Linear Regression**
    - Ridge regression
- **Non-linear Regression**
    - Polynomial regression
- **Classifier regressors**
    - Random Forest regressor
    - Decision tree regressor

Ridge Regression

Initially the r2 score was checked by fitting data to a linear regression model and it was around 0.1325.
Getting low r2 score is not always bad as there is bias variance trade off. It just means that the line of best fit model does not account for all the variance in the data.
Ridge regression is a technique that is used for punishing the data that has multicollinearity problems. This also helps in preventing overfitting and reducing model complexity. Ridge regression uses L1 penalty term which is the sum of squares of coefficients.
The dataset was split into training and test data which was then fit into ridge regression model.
Result-The r2 score was almost the same as linear regression model that is 0.1325 and RMSE 4638.488200154108
. This is because ridge regression deals with multicollinearity issues which is not existing in our dataset which is why there is not much difference in r2 score.

Polynomial regression

This is non-linear type of regression which has n degrees of polynomial forms. In this case degree 2 was chosen. The model was initiated and fit on data
Result-The r2 score was around 0.25567 which is twice the score of linear regression model and RMSE 4297.498067318087

Hence polynomial regression fits better that linear regression. As the degree the of polynomial regression increased the r2 score also kept improving. For example- for degree 6 the r2 score was around 0.41535 and RMSE 3808.0439722462056

Decision tree regressor

Decision tree can be used as a classifier and also as a regressor. It is a top down approach where the data is split into smaller subsets depending on the homogeneity of the data.
Result-The decision tree model was built and fit on the data. And the r2 score was around 0.63352 and RMSE 3015.9327159112177

Random Forest regressor

It is just a bunch of decision trees and works in similar fashion to decision tree
Result- The random forest regressor model was built and fit on the data. And the r2 score was around 0.635818 and RMSE 3006.1434045753667

Random forest model best fits the data.

Hyperparameter tuning

This is a process in which the corresponding set of best parameters are given to an estimator before training to get the best possible r2 score.

Steps:

1. Use get_params () attribute for the estimator on which hyperparameter tuning is going to be performed.
2. Create the parameter grid with the respective parameters of the estimator.
3. Initiate the parameter tuning method by giving in the parameter grid created and fit on the data.
4. Once training is done use best_params () and best_score () attributes from the parameter tuning method
5. Apply the obtained set of parameters in re building the estimator and note the change in r2 score.

Ridge regression with Gridsearch CV

The parameter grid was built, and the data was fit on the parameter tuning method.
There was not much change in the r2 score after parameter tuning method

Random forest regressor with Randomized search CV

The parameter grid was built with total of 1008 combinations of parameters and fit on Randomized search CV. The best parameters were taken, and the random forest model was rebuilt.
There was 0.5 of increase in r2 score after parameter tuning
Before tuning-63.5 and after tuning-64.01 in r2 score.

Random Forest regressor with Gridsearch CV

The same was performed with Gridsearch CV with 288 combinations of parameters.
There was 0.6 of increase in r2 score after parameter tuning
Before tuning-63.5 and after tuning-64.1 in r2 score.

The r2 score reached the maximum level before tuning and hence there is not much change after tuning.

There was not much change in RMSE after parameter tuning in Ridge regression, Random forest regressor and Decision tree regressors

Polynomial regression with Gridsearch CV

There is no separate estimator for polynomial regression. The X_train and X_test are transformed using Polynomial features in sklearn processing. And then fit on linear regression. Then a function called Polynomial regression was built by creating a pipeline between polynomial features and linear regression.
Grid search CV was initiated and fit on the data.
Result-The r2 score increased by 3 after tuning. Before tuning-25.53 and after tuning-28.34
And the RMSE reduced from 4297. 498067318087 to 4215.59765287098

Result

Random Forest regressor showed the maximum r2 score which means it best fits the model and the hyperparameter tuning best worked for polynomial regression.

Our dataset contains more of independent variables that have different levels as values (discrete values) instead of continuous numbers as values. Hence the classifier regressors perform well in the regression analysis of the Black Friday dataset.

References

1. **https://towardsdatascience.com/ridge-regression-for-better-usage-2f19b3a202db**
2. https://stackoverflow.com/questions/37110879/which-is-the-simplest-way-to-make-a-polynomial-regression-with-sklearn
3. https://alfurka.github.io/2018-11-18-grid-search/
4. https://towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e
5. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
6. https://stackoverflow.com/questions/30102973/how-to-get-best-estimator-on-gridsearchcv-random-forest-classifier-scikit

7. https://www.programcreek.com/python/example/93973/sklearn.feature_selection.VarianceThreshold