Machine Learning Engineer Nanodegree
Capstone Project
Joe Udacity
December 31st, 2050

Definition

* black Monday was the sell-off the day before the stock market crash of 1929 ,it Touted as the biggest shopping event of the year, it can get a little overwhelming when trying to find the best bargains.,While a relatively new concept in South Africa, it's been a big deal in the United States for years when retailers took advantage of the day after the traditional Thanksgiving Day holiday. It is often characterized by amazing deals and sometimes crazy behaviour by bargain hunters.

Project Overview

* A retail company “ABC Private Limited” wants to understand the customer purchase behaviour (specifically, purchase amount) against various products of different categories. They have shared purchase summary of various customers for a selected high volume products from last month.they decided to build a model to predict the purchase amount of customer against various products which will help them to create personalized offer for customers against different products.so Finally we want to predict the Purchase by getting some features of the product .and surly This will help for more effective future sales/marketing technique to maximize their profits

---

Problem Statement

* The problem focus on black Friday event ,it revolves around ABC company that wants to understand the behaviour of it's customer in this event to enhance their sales plan .So through the project we set model that can predict the amounts that the customers will buy from the shop products and how much from each category

* Many big companies use such systems to make their sales plans and promotional strategies such as amazon or Google where it posted recently a competition on kaggle requesting the Contesters

to make a model that predicts customers revenue. According to Google this also helps in determining the marketing budgets.

* we would solve this problem through cleaning our data by replace the missing data ,and normalize the data . then applying the regression model like linear regression ,decision tree and random forest . then improve the result of prediction by the grid search then evaluate it by learning curve .

Metrics

* First I applied cross validation using k-fold .

*then I used RMSE (Root Mean Square Error) to evaluate my model because RMSE is a suitable general-purpose error metric. Compared to the Mean Absolute Error, RMSE punishes large errors.Another benefit of using RMSE over something like MSE is that the error is in the same unit as the prediction. So i can quickly determine how far off my prediction is.

It represents the sample standard deviation of the differences between predicted values and observed values (called residuals). Mathematically, it is calculated using this formula:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

Analysis

*I have Data set of 550 000 observations about the black Friday in a retail store ,i get it through “kaggle” website https://www.kaggle.com/mehdidag/black-friday/version/1?login=true

But I used only 10000 sample because I use a local workspace not a GPU so the model was very slow .

*the data set contain several features that help us predict the price of the product like the gender of the customer as women may buy more than men so

*the data set contains 12 column with different kinds of variables either numerical

or categorical .

*features :

1-User_ID

2-project -ID

3-Gender : female-male

4- Age : with three cat " 0-17 / 26-35/ 36-45 /46-50 / 46-50 / 51-55 /

+55

5-Occupation :

6-City_Category

7-Stay_In_Current_City_Years : contains four values " 0 / 1 /2 /3 / +4

8-Marital_Status :contains two values: "1" mean married / "0"mean

single

9-Product_Category_1

10--Product_Category_2

11-Product_Category_3

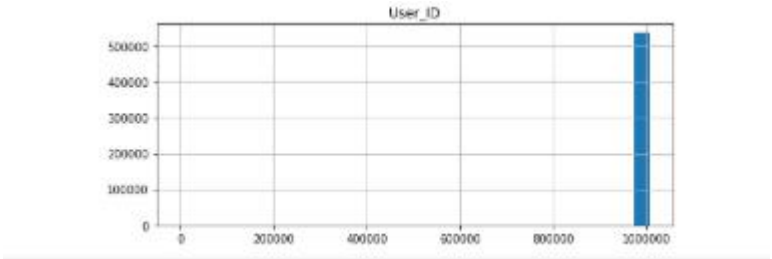
12-Purchase

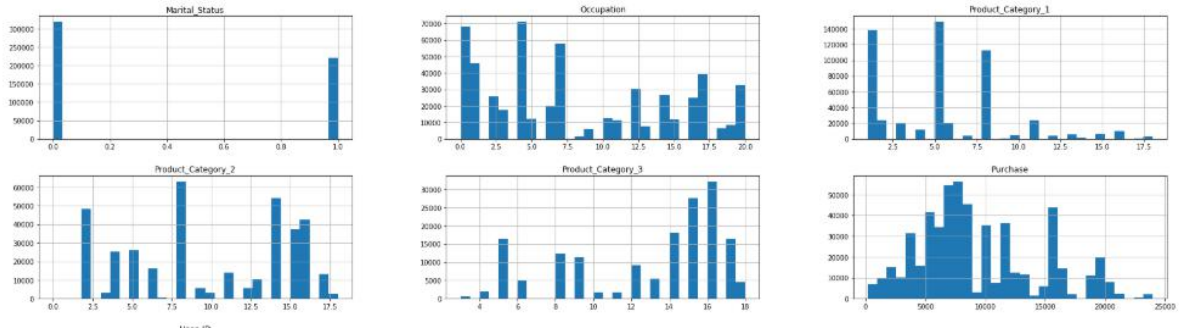
Exploratory Visualization

1-First we describe the data set to show the mean,std,max,min...

	Occupation	Marital_Status	Product_Category_1	Product_Category_2	Product_Category_3	Purchase
count	537577.00000	537577.000000	537577.000000	370591.000000	164278.000000	537577.000000
mean	8.08271	0.408797	5.295546	9.842144	12.669840	9333.859853
std	6.52412	0.491612	3.750701	5.087259	4.124341	4981.022133
min	0.00000	0.000000	1.000000	2.000000	3.000000	185.000000
25%	2.00000	0.000000	1.000000	5.000000	9.000000	5866.000000
50%	7.00000	0.000000	5.000000	9.000000	14.000000	8062.000000
75%	14.00000	1.000000	8.000000	15.000000	16.000000	12073.000000
max	20.00000	1.000000	18.000000	18.000000	18.000000	23961.000000

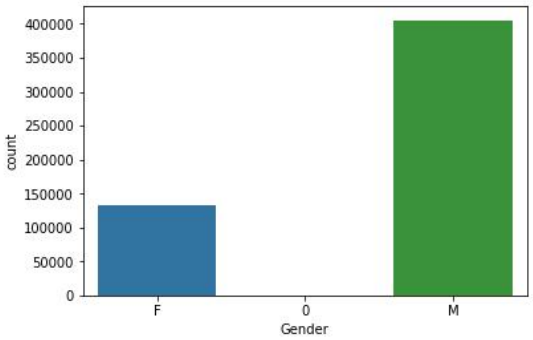
To check if there is any skewing data we graph the distribution of the features



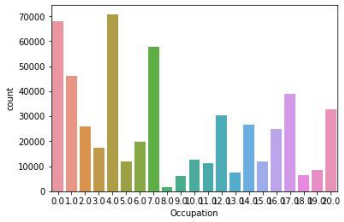


The graph show that the distribution of all columns is normal

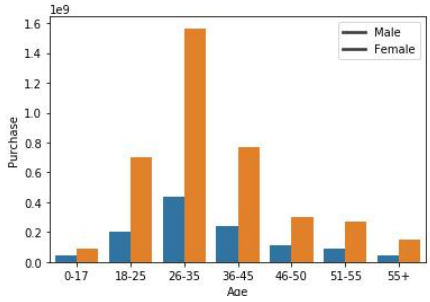
```
In [11]: sns.countplot(fearures['Gender'])
Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x1dc145f5358>
```



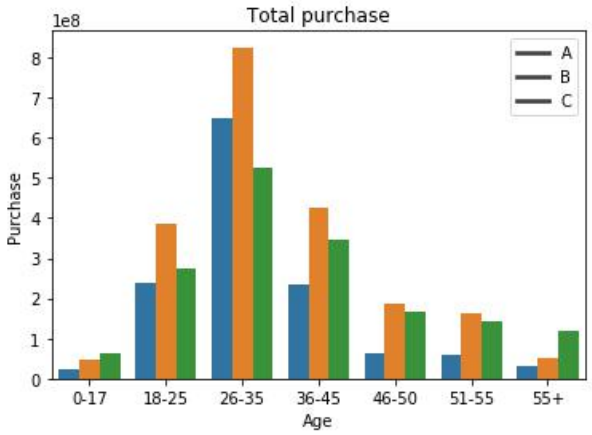

```
In [13]: sns.countplot(fearures['Occupation'])
Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x1dc14640550>
```



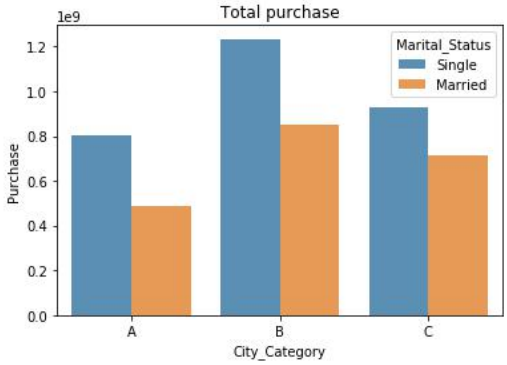
3-to know the relation between the features and target"purchase"



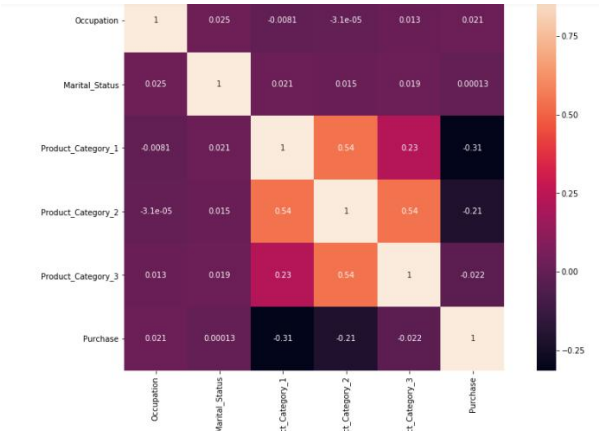
The graph for the age show that the people at range female at 26-35 give the highest sales



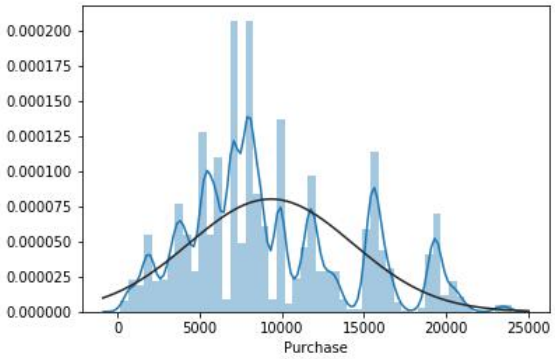
It show that the female at 26-35 from class a is the most sales



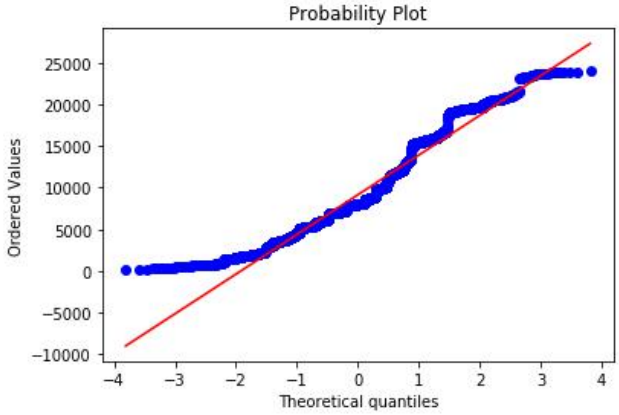
The single people from class B give the highest sales



Through the correlation graph : It is appear from showing the distribution for all features according to the purchase that is high affect for these features: product categories ,occupation,martial statues



This graph show that the purchase is very inconsistent as between two highest frequency there is a low bar .and after the frequency drop down it start to go up again! So we need to

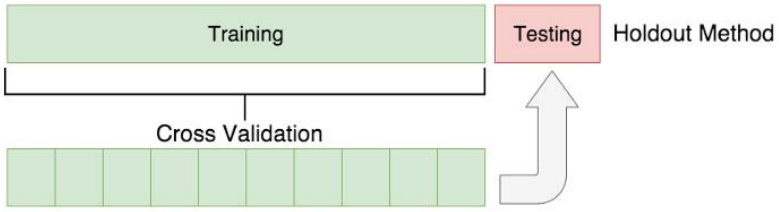


Then I used the plot of probability to make sure if the purchase need any transformation and decide it is type : and surly I appears that it needs

Algorithms and Techniques

*first I imported cross validation and KFold, cross_val_scorefrom Sklearn by theses parameters:(n_folds=10, shuffle=true, random_state=10 ,as these parameters give me better

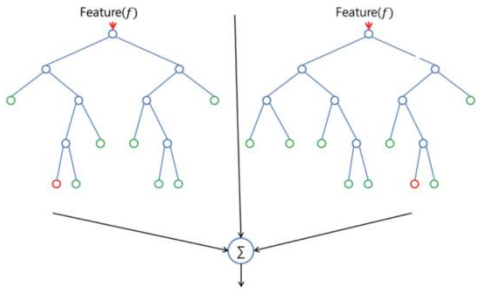
In K Fold cross validation, the data is divided into k subsets. Now the holdout method is repeated k times, such that each time, one of the k subsets is used as the test set/ validation set and the other k-1 subsets are put together to form a training set. The *error estimation is averaged over all k trials to get total effectiveness of our model*. As can be seen, every data point gets to be in a validation set exactly once, and gets to be in a training set *k-1*times. This significantly reduces bias as we are using most of the data for fitting, and also significantly reduces variance as most of the data is also being used in validation set. Interchanging the training and test sets also adds to the effectiveness of this method. As a general rule and empirical evidence, K = 5 or 10 is generally preferred, but nothing's fixed and it can take any



value.score , Data Permitting:

*after each model I apply a grid search :*to find the optimal* hyper parameters *of a model which results in the most 'accurate' predictions.*

*the first model I applied is the random forest :it is a supervised learning algorithm. Like you can already see from it's name, it creates a forest and makes it somehow random. The „forest" it builds, is an ensemble of Decision Trees, most of the time trained with the "bagging" method. The general idea of the bagging method is that a combination of learning models increases the overall result.One big advantage of random forest is, that it can be used for both classification and regression problems, which form the majority of current machine learning systems. I will talk about random forest in classification, since classification is sometimes considered the building block of machine learning. Below you can see how a random forest would look like



with two trees: It adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model. therefore, in Random Forest, only a random subset of the features is taken into consideration by the algorithm for splitting a node. You can even make trees more random, by additionally using random thresholds for each feature rather than searching for the best possible thresholds (like a normal decision tree does).
Another great quality of the random forest algorithm is that it is very easy to measure the relative importance of each feature on the prediction. Sklearn provides a great tool for this, that measures a features importance by looking at how much the tree nodes, which use that feature, reduce impurity across all trees in the forest. It computes this score automatically for each feature after training and scales the results, so that the sum of all importance is equal to 1.

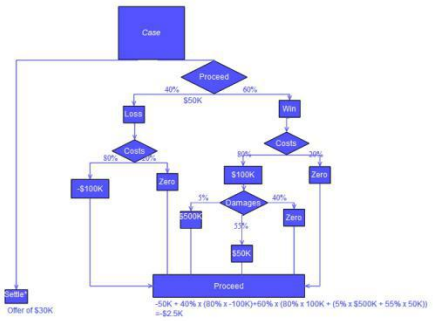
So first I imported Random Forest Regressor from sklearn then apply it and fit it to our features and target ,by parameters: cv=10, *max_depth=None*, *min_samples_split=2*, *min_samples_leaf=1* then give a result with 1023 , To enhance the result of the prediction I used the grid search with parameters 'n_estimators':[60,80,100,120,140],'max_depth':[6,7,8] and we get the best parameters and the best score for this model

*But to get the best score I tried two other models ,The second is decision tree it is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a

class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules.I n decision analysis, a decision tree and the closely related influence diagram are used as a visual and analytical decision support tool, where the expected values (or expected utility) of competing alternatives are calculated.

When fit it to our data with it is default parameters : *min_samples_split=2*, *min_samples_leaf=1*, *max_depth=None* it get the RMSE with 1580 and by applying the grid search with parameters of 'min_samples_split' :range(10,500,20),'max_depth': range(1,20,2) it gives a

Best score: 29.07



*Then finally by linear regression model:This method that is mostly used for forecasting and finding out cause and effect relationship between variables. Regression techniques mostly differ based on the number of independent variables and the type of relationship between the independent and dependent variables. Based on the given data points, we try to plot a line that models the points the best.

When fit it to our data I get *RMSE : 1937.35* , *mean square ERROR: 44.14604062732 92* And after the grid search it gives *Best score: 44.21*

so this is the best model until now. It is a method of modelling a target value based on independent predictors.

Benchmark

The benchmark used linear regression with RMSE =4300
And ridge regression model and decision tree with 2600 rmse ,but through my project I used random forest model ,linear regression and decision tree. And get good rmse with only 10000 sample not all the data set like the benchmark

---

Methodology

*Then I start data exploration process with : is null () . sum() , I found many missing data in the product category2 and 3 so we filled these samples by importing Imputer library with strategy = 'mean' ,
And by applying the graph for the purchase column by sns . distplot we notic e some unbalance so we imported boxcox and get Lambda: 0.488688
Then we needed to encode the categorical data in columns age ,gender,city category ,using get_dummies,so our model can get better accuracy
The difficult phase through cleaning is encoding it as the code have some errors until I solved it

*after cleaning our data I split it to target and features to start separating them to test and train samples , using k-fold I set the test sample=40% and set the random state =43 .

*through setting the models I faced many problems ,first they were very slow and it took much time to show any prediction , so I get only 10000 sample of the 555000 sample .to make the project faster cause I use local device not a

GPU and I was afraid to drop my accuracy .and decide which parameters were suitable was difficult too until sitting the grid search .
Finally I wanted to make features selection to improve our result and not to load our model with no benefit .and choosing the model was difficult also ,I tried svm but have low accuracy.

finally I tried to set plotting to determine if the model is robust with bootstrapping. But it so hard for my prcessor to finish this process it took much time so I deleted it .

Implementation

First I set a random forest model with the default parameters and
Then tried other models like the benchmark ,decision tree with better result with RMSE : 1584.2 and linear regression with RMSE : 1937.3

Refinement

to improve the result I set the grid search for random forest with parameters of 'n_estimators':[60,120,140], 'max_depth':[6,7,8]}.
and i finally get the best parameters:

bootstrap=True, criterion='mse', max_depth=7,
max_features='auto', max_leaf_nodes=None,
min_impurity_decrease=0.0, min_impurity_split=None,
min_samples_leaf=1, min_samples_split=2,
min_weight_fraction_leaf=0.0, n_estimators=140, n_jobs=1,
oob_score=False, random_state=None, verbose=0, warm_start=False)

And Best score: 28.84
And to improve the result for the decision tree we set the grid search with pa rameters of {'min_samples_split' : range(10,500,20),'max_depth': range(1,20, 2)}

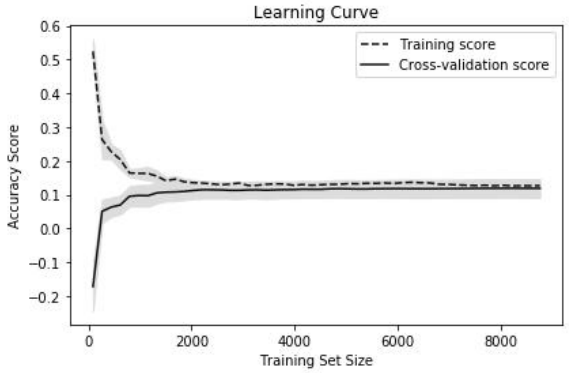
Results

finally I noticed that the linear regression with
RMSE : 1937.3521087
mean square ERROR: 44.146040
and after applying grid search we get Best score: 44.21

Model Evaluation and Validation

The advantage of linear regression and it is great when the relationship to between covariates and response variable is known to be linear (duh). This is good as it shifts focus from statistical modeling and to data analysis and pre processing. It is great for learning to play with data without worrying about the intricate details of the model.is extrapolation beyond a specific data set.so Linear Regression is great for learning about the data analysis process. However, it isn't recommended for most practical applications because it oversimplifies real world problems.Space complexity is very low it just needs to save the weights at the end of training. hence it's a high latency algorithm . Its very simple to understand Good interpretability Feature importance is generated at the time model building. With the help of hyper parameter lambda, you can handle features selection hence we can achieve dimensionality reduction

Then I applied the learning curve with the best model "linear regression with it is best parameters after the grid search , it shows how error changes as the training set size increases. The diagram below should help you visualize the process described so far. On the training set column you can see that we constantly increase the size of the training sets.



As the graph show the difference between the prediction result for the training and testing data is small as the accuracy increases with increasing the training samples then until it be stable ,and the accuracy for the test

samples decrease until it be stable with small differance between the train set

---

Justification

The bench mark set the decision tree with 2680 rmse for 550000 sample ,and I set linear regression with 1937 with only 10000 sample ,and surly increasing the data set would increase the result

---

Reflection

We set five steps through our project , first exploration the data then cleaning it and apply visualization to the features to show the affect on the purchase to make a good prediction . then divide the data to test and train with the k fold cross validation ,and finally set several models until give the best accuracy with the grid search ,...and so we can predict the purchase of any product according to the given features and surly this would increase the shop sales .

*Through all the five steps we applied , the most *difficult aspects*
Was applying the grid search and tunning the parameters until getting the best score

*And the best interesting aspect was the pre processing .

Improvement

Surely if we used all the data set with all 550000 and by searching for other models with more effective technique we would get better score

To use a suitable processor to apply plotting to determine if the model is robust with bootstrapping and other hard visuals

---

Resources

https://seaborn.pydata.org/generated/seaborn.distplot.html

https://www.datacamp.com/community/tutorials/ml-black-friday-datasethttps://www.kaggle.com/dalalmanish/black-friday-python

https://www.freelancer.com/community/articles/algorithms-machine-learning-engineers-need-to-know

https://medium.com/diogo-menezes-borges/project-3-analytics-vidhya-hackaton-black-friday-f6c6bf3da86f
Code used for this graph is from https://www.kaggle.com/serigne/

https://machinelearningmastery.com/handle-missing-data-python/

https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html

https://chrisalbon.com/machine_learning/model_evaluation/plot_the_learning_curve/

https://www.kaggle.com/sungsujaing/

https://stats.stackexchange.com/questions/153131/gridsearchcv-regression-vs-linear-regression-vs-stats-model-ols

https://scikit-learn.org/stable/auto_examples/model_selection/plot_learning_curve.html#sphx-glr-auto-examples-model-selection-plot-learning-curve-py

https://medium.com/diogo-menezes-borges/project-3-analytics-vidhya-hackaton-black-friday-f6c6bf3da86f

https://github.com/ahm320/Black_Firday/blob/master/Black_Friday.ipynb