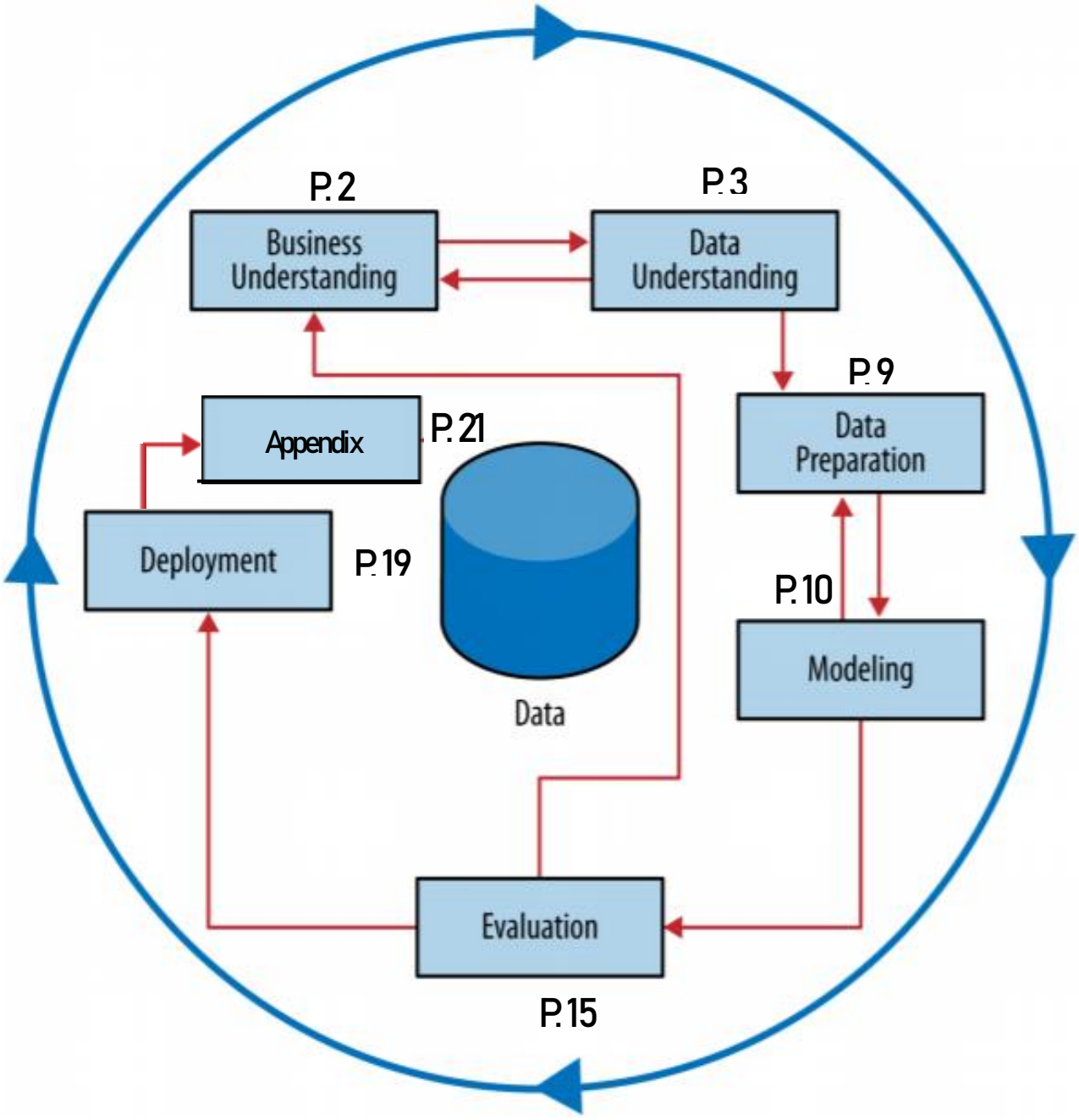
Data Mining for Business Analytics: Sales Forecasting

Shirley Gao, Nomuka Luehr, Adhiya Badruddin, Darius Allen, John Lee

Professor George Valkanas

Table of Contents



Business Understanding

Background

With the dominance of internet retailers, especially during large events like Black Friday, it is more important than ever for brick and mortar stores, like the one represented by our data, to invest in better tools to generate accurate sales forecasts and minimize the occurrence of shortages or overstocking. As the middlemen of many industries, large retailers do not have high profit margins, with 3.2 percent as the industry average. Therefore, success lies in maximizing the volume of sales while minimizing the potential costs of over stocking. Companies such as Amazon, who have perfected their sales forecasting, are a direct threat to physical retailers, who must attempt to remain competitive and profitable. In fact, surveys suggest that almost 25% of Amazon's retail revenue comes from customers that first tried to buy the product at their local stores.[1] To manage their inventory, stores rely on sales forecasting based on previous sales to anticipate demand and assist them in stocking their physical locations. With accurate forecasts, managers in all aspects of the supply chain can make well-informed business decisions for staffing, logistics, sales and inventory. Ultimately, optimal sales forecasting will allow for peak operational alignment in terms of logistics and workforce.

Part 2: Current Solutions to Sales Forecasting

While some companies have invested heavily into sales forecasting technology, many physical retailers still rely on few measures besides past sales data. Stores usually only consider the size of customer base, amount of sales in past years, and estimations of the growth rate of the market. Most stores therefore focus on one feature of a consumer, their spending habits, however, changes within

this sector between years can be due to many reasons besides actual increase in volume, including inflation. Forecasting the growth or decline of a single variable without the help of variables that capture the surrounding environment besides price and volume may not be complex enough to factor in all potentially informative attributes.

Part 3: Our Solution and Target Variable Based on Business Understanding

Our project uses data from one Black Friday from a retail store. We hope, through our model, to predict the optimal amount of inventory stock for three product categories for next Black Friday based on the characteristics of their shoppers rather than using sales frequency and timing as inputs; the purpose of our models in general is to predict sales for a product category for next year based on the characteristics of the retailer's pool of customers. Therefore, our models will answer the two following questions: 1) Will this customer purchase a product from Product Category X? 2) How much product will a customer purchase from Product Category X?

Beyond sales forecasting, we will implement cost analysis to visualize the success of our model in predicting optimal stock levels, which we measure through the corresponding sum product of the cost of over and understocking and the error rate of our model. This will ultimately allow us to evaluate how well our model meets our business requirements.

Data Understanding: Data Search

Before modeling, we must first understand the information required from our data given our business goals and then pick a data set that allows us to meet them. Our team's first imitative is to accurately predict which product category a given customer instance will purchase. Afterwards, we want to use the regression results to estimate an instance's purchase amount, in terms of number of products. Since in both cases we have a set target variable, our models will be supervised. Additionally, since we want to predict the amount of products sold on top of a product category, our models will include classification and regression.

Data Requirements

Given that we will implement a supervised classification or regression model to predict sales forecast values for different customer instances based on their features, our data set needs to include the following:

- Historic data on relevant features, such as identifiable demographical or behavioral characteristics
- Historic data on both our target variables, product types purchase and the amount of those a customer with certain attributes purchase
- Since we plan on using regression models, our data needs to be able to be presented in a numerical from

Data Source

Our Data:

https://www.kaggle.com/mehdidag/blackfriday?fbclid=IwAR1nyEOI_fa0N2slSwZ7jUQcVC3pjfM9AJ7YIKma06mRJxAv9w5gBSa_LfE

After finding limited results from searching for commercial data sources from private companies, we decided to use the following data of sales of three different product categories, represented through purchase amount, at a physical retail store on Black Friday – a first glimpse is shown below.

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Product_Category_1	Product_Category_2	Product_Category_3	Purchase
0	1000001	P00069042	F	0-17	10	A	2	3	NaN	NaN	8370
1	1000001	P00248942	F	0-17	10	A	2	1	6.0	14.0	15200
2	1000001	P00087842	F	0-17	10	A	2	12	NaN	NaN	1422
3	1000001	P00085442	F	0-17	10	A	2	12	14.0	NaN	1057
4	1000002	P00285442	M	55+	16	C	4+	8	NaN	NaN	7969

Data Exploration

After finding our data source, we analyzed potential target variables and each included feature for relevancy, type, size, and completeness of data. These quality assurance measures will allow us to evaluate and improve the quality of our data and decipher if we there are data cleaning and filling measures we can implement. Additionally, we looked at the distribution of different values for each feature and its relationship with purchase amount to see if there are already evident trends, such as purchase power of certain categorical values. While trends and potential purchase power dynamics will be captured by our model, feature exploration will allow us to better understand how to optimally format our data for future steps.

Labels: Product Category 1 | 2 | 3

Type: Integer

Observations: The three columns for each product type will serve as the categories for classification and amount of products purchased within each category for our regression models. The data format is suitable for models that take categorical, as we will dummyze them, and numerical values. However, Product Category 2 and 3 contain significant amounts of missing values, denoted by the "NaN", which will have to be filled in order to put into our models.

```
Product_Category_3    69.441029
Product_Category_2    31.062713
dtype: float64 %
```

Exihibit 2: Percentage of missing values in Product Category 1 & 2

Feature Exploration

User_ID & Product_ID

User ID Type: Integer

Product ID Type: Categorical Object

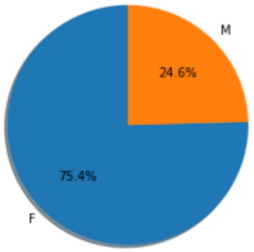
Observation: The "User ID" is unique to a customer account in the store but not to each purchase instance within the data. As the "USER ID" is not formatted in a numerically utile format and its

inclusion may be confused with product categories, we will omit this feature and treat every instance as a different customer. Similarly, the data includes “Product IDs” on top of products purchases organized by three categories. Since “Product IDs”, formatted in the form “P000XXXX” cannot be easily represented as a number, we have chosen to omit this feature.

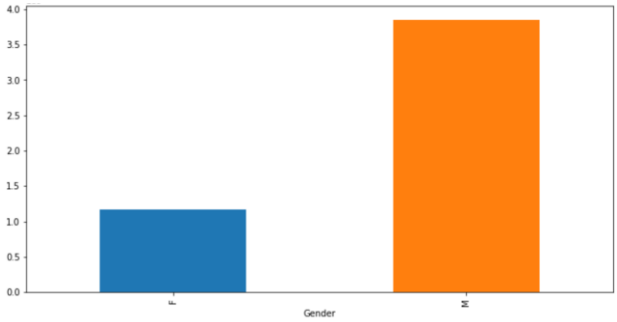
There were too many products.

Gender

Gender [“M”, “F”]



Gender (X) & Purchase (Y)

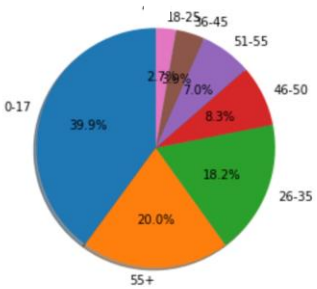


Type: Categorical Object

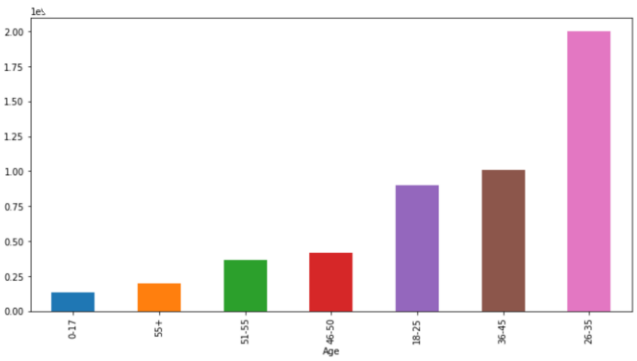
Observations: The feature “Gender” includes two unique categorical values “M” and “F”, which will need to be dummyzed to comply with regression models. An interesting aspect of this feature is that while nearly three quarters of the shoppers are female, the males contribute (~3.6 versus ~1.2) three times more towards overall purchase and therefore have a higher purchase power. This is likely to be demonstrated by the model

Age

Age
[0-17, 18-25, 26-35, 36-45, 46-50, 51-55, 55+]



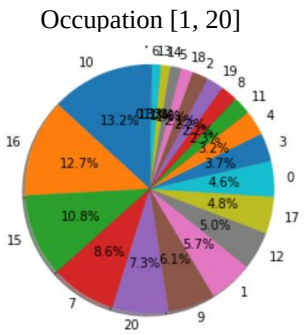
Age (X) & Purchase (Y)

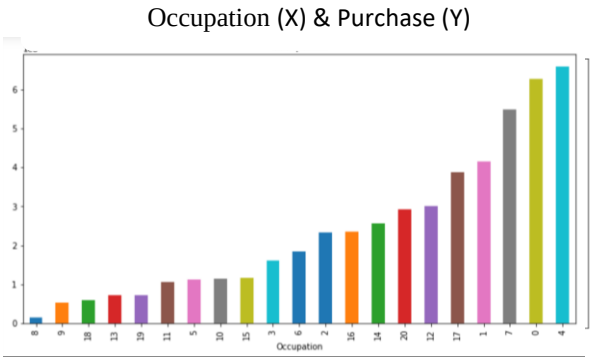


Type: Categorical Object

Observations:
Since the range of values for age is high, the data set uses ranges of varying lengths (from as low as 4 (for 51-55) and unbounded (for 55+) to represent this feature. While it makes it easy to visualize different consumer segments, this type of representation cannot be used for regression models and will need to be transformed into an appropriate format. Additionally, similar to the imbalance in purchase power between males and females and the inverse relationship between greater relative to feature size and purchase, the larger the age category size the smaller the purchase power.

Occupation

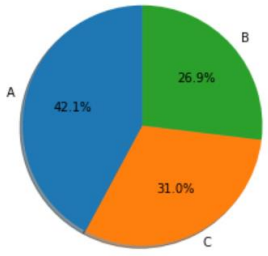


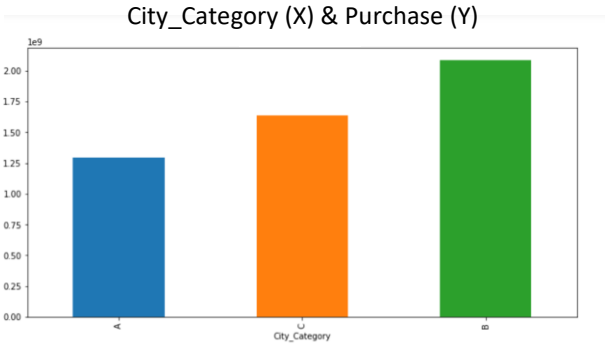


Type: Integer

Observations: The data set uses a discrete set of integers from 1 to 20 inclusive to represent different occupations, which will work well with models that use categorical and numerical data types. There does appear to be a correlation between certain occupations and purchase amount, which will be included in the model's prediction. The data does not include detail regarding which occupation corresponds with which value.

City_Category





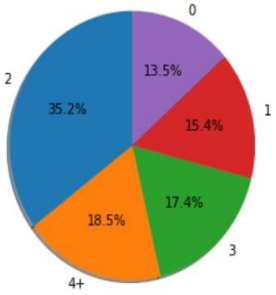
Type: Categorical Object

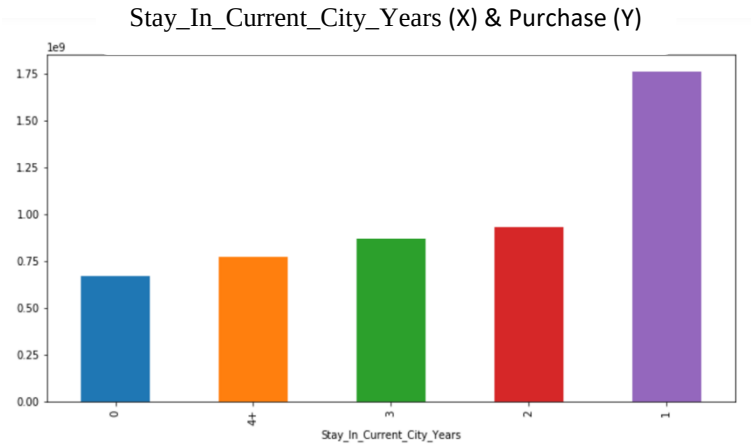
Observations: The categorical values ‘A’, ‘B’, and ‘C’ must be represented numerically for our models.

The inverse relationship between category size and purchase power is present for the city categories.

Stay_in_Current_City

Stay_In_Current_City_Years
[1,2,3,4+]



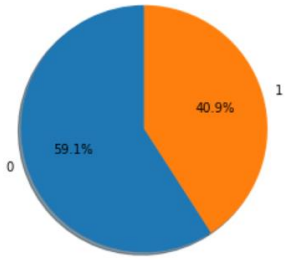


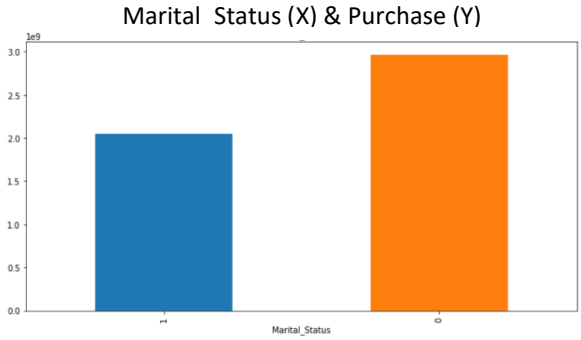
Type: Categorical Object

Observations: While the rest of the data can be represented by an integer, the addition sign next to the four at “4+” needs to be handled with in order to be placed into a regression model.

Marital_Status

Marital_Status
(0=“Single” : 1=“Married”)





Type: Integer

Observations: The data uses binary integer values of “1” and “0” that correspond to “married” and “single”. The category size of those that are single and married are proportional to their purchase power (the color is switched in the second graph).

Data preparation

Complete Data

Variables Changed: Product Category 1 & Product Category 2

Our first data preparation step was to ensure the completeness of our data set. To clean our data of missing or misconstrued data points and replace them with the appropriate pseudo data, we decided to use the constant “0”. During our first glance, we noticed that there were large amounts of “NaN” values in both Product Category two and three. Product category one did not contain any NaN values, so its data was untouched. We could have taken a few different routes to solve this issue, including predicting values besides a constant, but logically we assumed that “NaN” would be synonymous with “0” in that it represented that the customer instance did not buy a product from the category.

Dummysizing

Variables Changed: Gender & Marital Status

As our core model will be a regression variant, which requires all numerical fields, it is imperative that we are able to accurately represent our categorical values through numbers. Starting with Gender, we created two new unique columns in our data to replace the original: “Gender_F” and “Gender_M.” In these columns, the data point was either 0 or 1, corresponding with male or female. Similarly, to represent a customer’s Marital Status numerically, we opted to change from the binary “Yes” and “No” to 0 and 1.

Dummysizing Non-binary Variables

Variables: Age & Stay in Current City

While binary number representation works for categorical values with two logical options, for variables with many categories or require minor changes to become numerical we created non-binary dummy variables. The feature “Age” originally had 7 different age range groups: 0-17, 18-25, 26-35, 36-45, 46-50. 51-55, and 55+. While having a similar effect, we created 7 integer variables, ranging from 1 to 7, to represent each range. Additionally, to achieve a numerical representation for our “Stay in Current City” field, we replaced data points of “4+” into “4”.

Data Preparation for Accurate Testing

Considering that our data set is limited in time span, as it only encompasses one year, we implemented two measures to ensure the quality of our evaluation measures. We opted for a randomized test and training split before running our models and incorporated cross validation into our training methods. The train and test split will allow us to assess our model based on its performance on new instances, which will maintain the integrity of our results, in our deployment phase. The iterated methodical train and test split used in cross validation will ensure that our models are able to update and learn repeatedly from predicting for new instances during the training phase.

For our test and train data split, we decided to use ninety percent for training and save ten percent for testing. As our model has many instances but lack in time variation, we felt a greater training ratio would be beneficial because the cross validation included in the models will allow the models to learn more often from new instances. Furthermore, we chose 3-fold cross validation to optimize computation time, as we will be repeating our models multiple times to predict for more than various product categories.

Modeling

We will be using both classification through a decision tree classifier and regression through ridge regression to predict product category purchase amounts. As Product Category 1 does not have any missing or “0” values, as in every customer purchased a product from this category, we will be using only ridge regression for our predictions. However, since we must first decide if a user will

purchase any product from Product Category 2 or 3 and then calculate purchase amounts afterwards, we will integrate both the classification and regression model for the remaining two.

Baseline Model

Before implementing our models, we want to create baseline predictions of product category amount and resulting error rates that can be used for comparison and performance evaluation. As predicting based on averages, similar to expected value, resembles current simple sales forecasting models discussed in our business understanding phase, we have opted to use mean absolute error as our baseline prediction tool.

For each product category, we first found the mean absolute error, which is the summation of the difference between every instance value and the mean of the entire set of values. Afterwards, we predicted product category purchase amount for each instance using the mean as our value. To calculate the “percentage off” of our base predictions, we found the percentage difference between the predicted mean based values and actual values. The baseline prediction measures for our three product categories are shown below.

```
Baseline

Product_Category_1
Mean absolute error: 2.8898
Our predictions are 0.5645 % off from the actual values

Product_Category_2
Mean absolute error: 5.5991
Our predictions are -0.5719 % off from the actual values

Product_Category_3
Mean absolute error: 5.3813
Our predictions are 0.3549 % off from the actual values
```

Feature Selection

A crucial step that will be included in every one of our models will be picking optimal features. As we will be using a regression model, data reduction techniques will be especially useful; being

selective about features will help our regression results, the coefficients, to be stable every time we run it. Especially since we are predicting for three different product categories, decreasing the number of features will allow it to generalize better.

Forward Feature Selection

We chose to use forward feature selection as our method for selecting optimal features based on the measurement of negative mean absolute error. As the forward feature method chooses “good features” based on resulting model measures, we actually have to implement a regression model within the process.

To start, the forward feature method will run ridge regression individually for all features and the feature with the model that yields the lowest negative mean absolute error was labeled as a known “good feature” and will be included in our model. The method will be iterated again but will run a model that includes the feature we just identified as “good” and an additional one, where when combined with the first feature will result in the lowers negative mean absolute error out of the other feature combinations and one that is lower than the previous iteration. These steps were repeated until the inclusion of an additional feature caused the measure to increase; the resulting array of “good features” are therefore the features that should be included in our model to achieve the optimal measure of negative mean absolute error.

Ridge Regression Model

Target Variable: The amount of products the customer will buy in the given product category

The ridge regression estimates product category amount while deploying complexity control and cross validation within the feature selection phase.

Feature Selection & Complexity Control

To prevent overfitting of data, we will use alpha as our complexity control parameter. Furthermore, we will normalize our data since sales forecasting incorporates randomness. In the

same phase, we will select our optimal features. Therefore, for every feature, total of 15, we will use forward feature to see if we want to add it to our set of “good features” and run complexity control to find the corresponding best alpha value out of the possible options (Exhibit 5). After all iterations, we will choose the alpha and combination of features that yield the lowest negative mean absolute error.

```
alpha_ridge = [1e-15, 1e-10, 1e-8, 1e-4, 1e-3,1e-2, 1, 5, 10, 20]
for i in alpha_ridge:
    model = Ridge(alpha=i, normalize=True) # we select ridge regression
```

Exhibit 5: Complexity control of alpha

Cross Validation

After choosing the best alpha parameter, within the same iteration we will train our ridge regression model using three cross fold evaluation. This means that in each our training phases we will first train the model on two thirds of our data, test the model based on the remaining third, and repeat these steps until all partitions have been included in training and testing.

Ridge Regression Product Category 1

To start the model, we first used forward selection and surprisingly found only one optimal feature, “Age”.

```
We select ['Age_18-25', 'Age_26-35', 'Age_55+', 'Age_0-17', 'Age_36-
45', 'Age_46-50']
Mean absolute error (train data): 2.8745
```

After completing complexity control and cross validation, we tested our model on the remaining ten percent test data from the random split. Our results are as follows:

```
Product_Category_1
Model: Ridge Regression
Mean absolute error:  2.8629
Our predictions are 0.5678 % off from the actual values
```

Classification Product Category 2 and 3

Target Variable: Will the customer purchase an item from this category

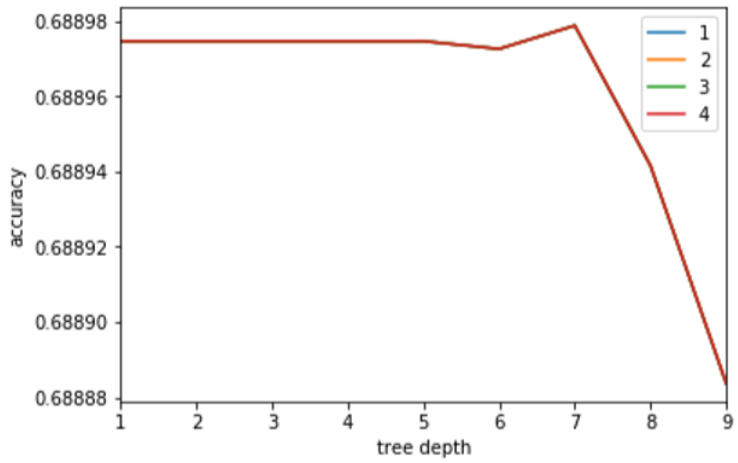
For our Product Category 2 and 3 we will use classification model, a decision tree classifier, to predict if a customer will buy a product from our category. We will then use the “0” results, signifying “No”, as our regression value for no purchases. As decision tree classifier already uses feature selection, we will not need feature selection as in ridge regression. Additionally, we will use AUC/ROC to evaluate and choose the correct threshold values, shown in next phase.

Complexity Control & Cross Validation

As a complexity parameter, we iterated through various different max tree depth values and found the optimal one that yielded the highest accuracy when used with three fold cross validation.

Product Category 1 Max Depth

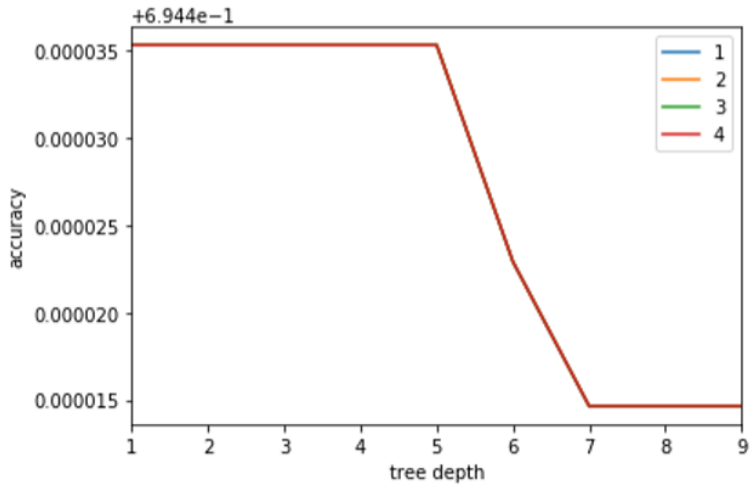
```
Text(0, 0.5, 'accuracy')
```



From the graph, we choose max_depth = 7 We do not set min_samples_leaf because it does not make a significant difference.

Product Category 2 Max Depth

```
Text(0, 0.5, 'accuracy')
```



Product 2 and 3 Ridge Regression

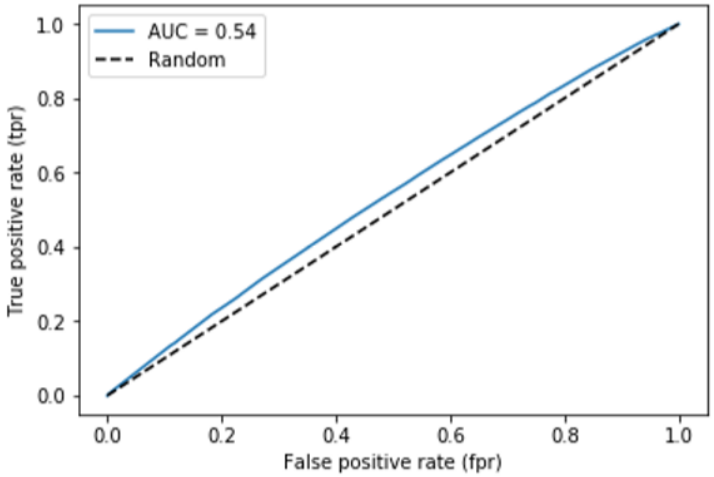
After running classification and answering our first question, we will use the “0” values found before as values for our regression analysis and then run ridge regression to find the rest of the values. We will use the same for loop that includes forward feature selection, cross validation, and complexity control using ridge regression to predict the values for Product Category 2 and 3. The results of the classification and ridge regression problem for Product Category 2 and 3 will be discussed in the deployment phase.

Evaluation

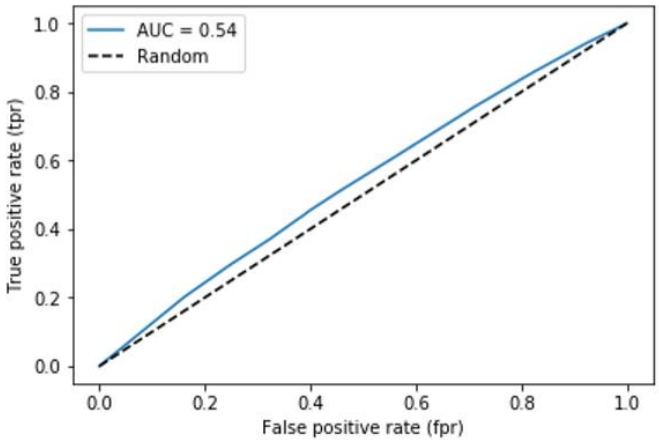
For our ridge regression models we used the negative mean absolute error for evaluation purposes. However, for the classification part of our modeling we used AUC/ROC to choose our threshold values, which represents the probability above which we will declare that the customer will buy any product in the category.

First, for both Product Category 2 and 3, we found the AUC score based on the decision tree classifiers trained in last section using the true positive rate, false positive rate, and various different thresholds.

AUC/ROC Product Category 2

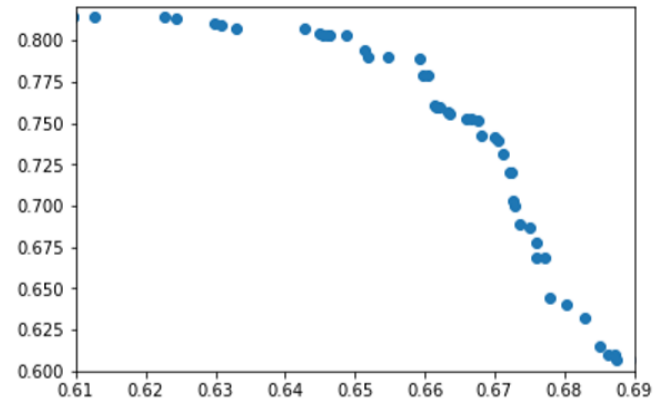


AUC/ROC Product Category 3



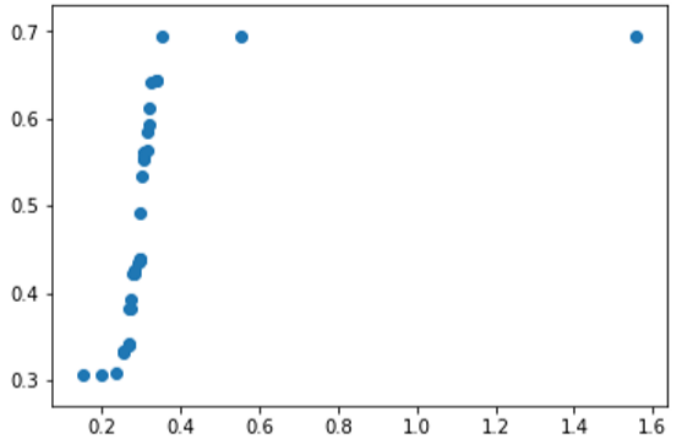
Afterwards, we graphed the overall accuracy of our model based on different threshold values and picked the threshold value that resulted in the highest accuracy.

Optimal Threshold for Product Category 2

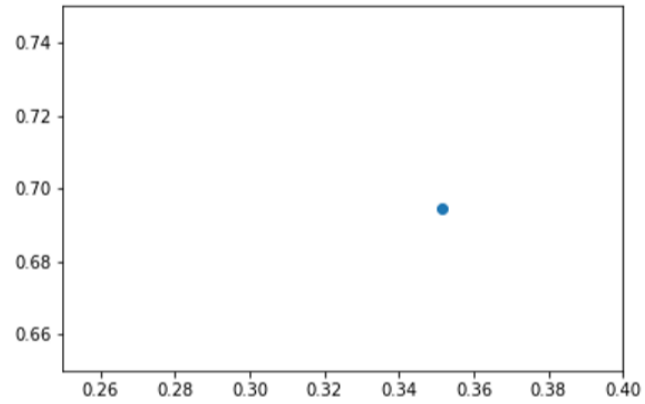


From the graph, we pick 0.674 as our threshold value (the point where accuracy starts to drop sharply.

Optimal Threshold for Product Category 3



Zoomed In View



Set threshold = 0.332778

Deployment

Finally, we will use the remaining ten percent of new testing data to evaluate the performance of our models in terms of our business problem. Beyond looking at our resulting mean absolute error, for each product category, we will first estimate expected profit, based on the cost benefit data below, and then compare then with the estimated profits found using our baseline model.

Baseline

```
Product_Category_1
Mean absolute error: 2.8898
Our predictions are 0.5645 % off from the actual values

Product_Category_2
Mean absolute error: 5.5991
Our predictions are -0.5719 % off from the actual values

Product_Category_3
Mean absolute error: 5.3813
Our predictions are 0.3549 % off from the actual values
```

Product Category 1 Results

Features

```
We select ['Age_18-25', 'Age_26-35', 'Age_55+', 'Age_0-17', 'Age_36-
45', 'Age_46-50']
Mean absolute error (train data): 2.8745
```

Results

```
Product_Category_1
Model: Ridge Regression
Mean absolute error:  2.8629
Our predictions are 0.5678 % off from the actual values
```

Product Category 2 Results

Features

```
We select ['City_Category_C', 'Age_18-25', 'Age_0-17', 'Age_26-35',
'Age_36-45', 'Marital_Status_N', 'Age_51-55', 'Gender_F']
The best mean absolute error we get is 5.5932
```

Results

```
Product_Category_2 - after classification
Model: Ridge Regression
Mean absolute error:  6.3828
Our predictions are -2.9521 % off from the actual values
```

Product Category 3 Results

Features

```
We select ['City_Category_C', 'Gender_F', 'City_Category_A', 'Age_55
+', 'Age_46-50', 'Age_51-55', 'Age_36-45', 'Age_18-25']
The best mean absolute error we get is 5.3638
```

Results

```
Product_Category_3 - after classification
Model: Ridge Regression
Mean absolute error:  4.7526
Our predictions are 71.4929 % off from the actual values
```

Cost Data

Although our data only includes information about purchase decisions and not the accompanying costs, we created our own pseudo cost data to better visualize the impact of our model on our business problem. Price per product was found by finding the weights of each product category as a portion of overall number of products sold then multiplying it by overall sales. Then we found the product price per category by dividing the corresponding amount by number of products in each category. Surprisingly, the product price was the same for each category at around $518.

Price Per product= $518

Individual benefit = $518

Overstock cost = $518*0.98

Understock cost = Price

Cost of Model per One Unit of Error: %Overstock*Overstock Cost + %Understock*Understock Cost

Cost Estimates

Product Category 1

Baseline Cost: $14.96 per unit

Predicted Cost: $14.82 per unit

Product Category 2

Baseline Cost: $29.00 per unit

Predicted Cost: $33.06 per unit

Product Category 3

Baseline Cost: $27.87 per unit

Predicted Cost: $24.61 per unit

Conclusion

Using the cost data we created based on the industry average of two percent profit margins for retailers during Black Friday, we visualized the potential of our model in reducing costs related to incorrect sales forecasting. Since sales forecasting can have large potential business and ethical risks, as in correct predictions can significantly impact customer purchase behavior, we felt that analyzing the potential cost ramifications would help mitigate this problem. Using our decision tree classifier and regression model, we were able to reduce the overall cost of in correct sales forecasting for Product Category 1 and 3. However, our model actually increased the potential cost of wrong sales forecasting. We believe the discrepancy in Product Category 2 is due to the potential errors in classification seeping into the regression model. While for the majority of products we were able to positively improve on the traditional sales forecasting model, our gains were not as significant as we expected.

Appendix

Contributions

Shirley Gao – Modeling and data preparation

Nomuka Luehr – Modeling, data understanding, and evaluation

Adhiya Badruddin – Data preparation, modeling

Darius Allen – Business understanding and evaluation

John Lee – Deployment and business understanding