




SALE

Predicting Black Friday Consumer Spending



The “Black Friday” Dataset

The Black Friday dataset consists of 12 features and 537,577 rows.

One continuous feature, tells how much money each purchaser spent (in U.S. dollars).

Seven categorical features describe each purchaser sampled. Four discrete features describe each product.

- ❖ Continuous Feature
 - Purchase

- ❖ Categorical Features
 - User ID
 - Age
 - Gender
 - Marital Status
 - Occupation
 - City Category
 - Stay In Current City - Years
 - Product ID
 - Product Category 1
 - Product Category 2
 - Product Category 3

❖ Objective

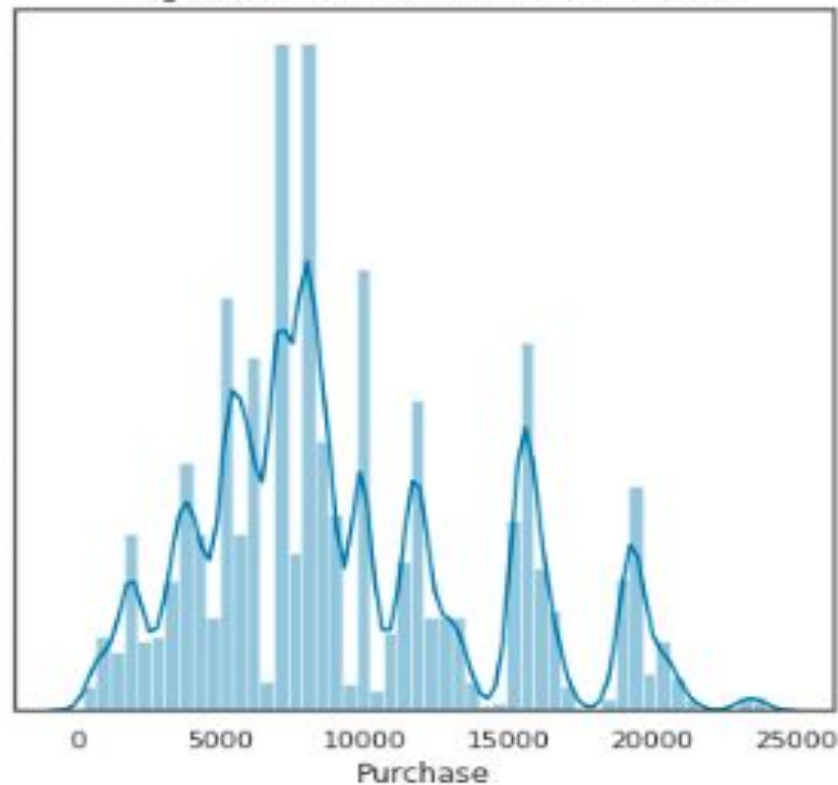
Predict total Black Friday purchases in U.S. dollars

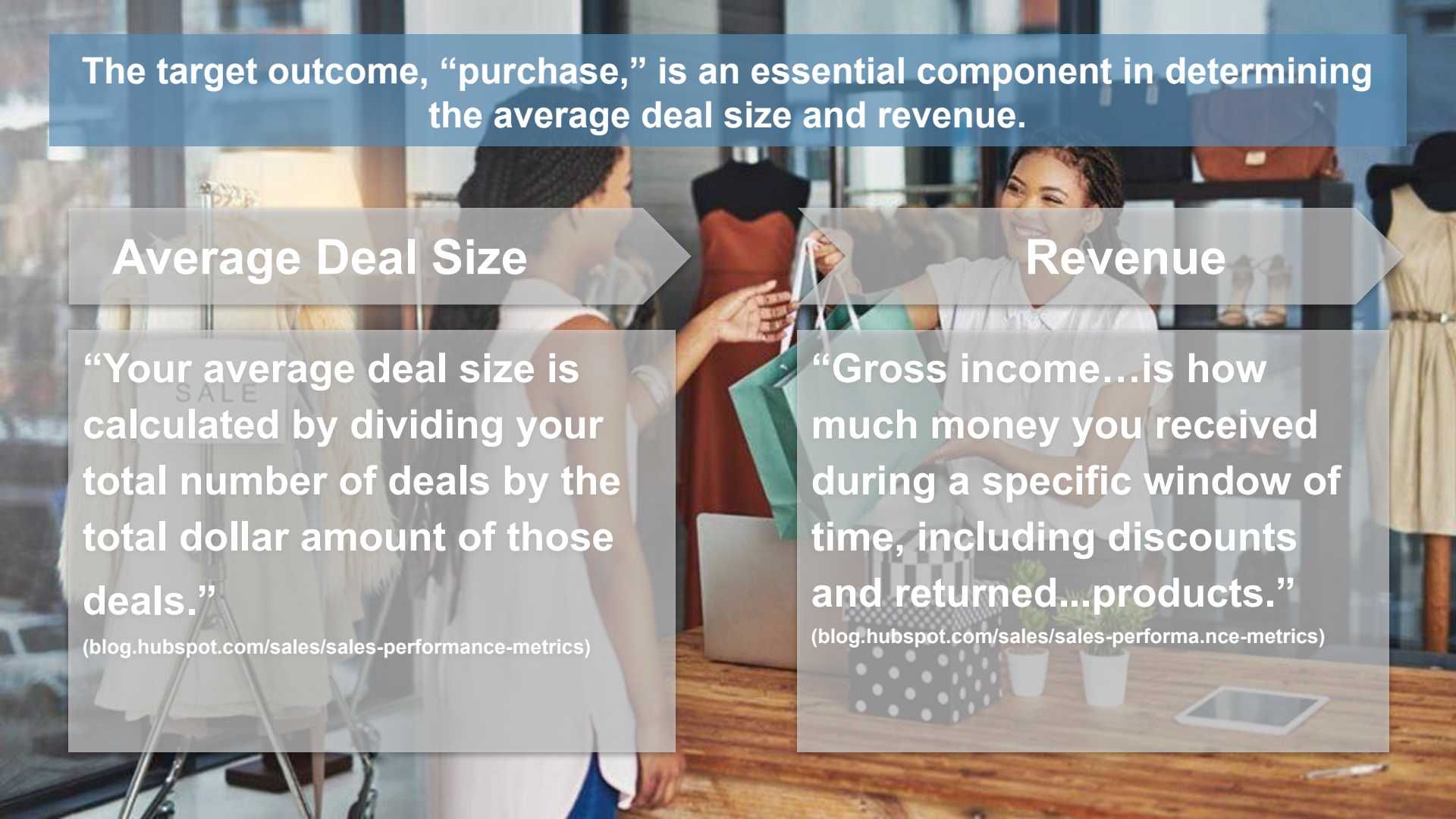
❖ Economic Value

- Average Deal Size
- Revenue component prediction and accuracy values

Outcome Feature: Purchase

Target Outcome: Purchases in U.S. Dollars





The target outcome, “purchase,” is an essential component in determining the average deal size and revenue.

Average Deal Size

“Your average deal size is calculated by dividing your total number of deals by the total dollar amount of those deals.”

(blog.hubspot.com/sales/sales-performance-metrics)

Revenue

“Gross income...is how much money you received during a specific window of time, including discounts and returned...products.”

(blog.hubspot.com/sales/sales-performance-metrics)



SALE

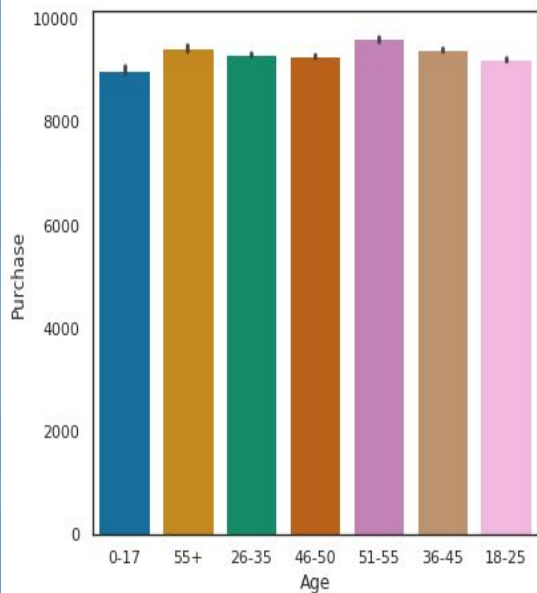
Age
Gender
Occupation
Product Category

Key Features

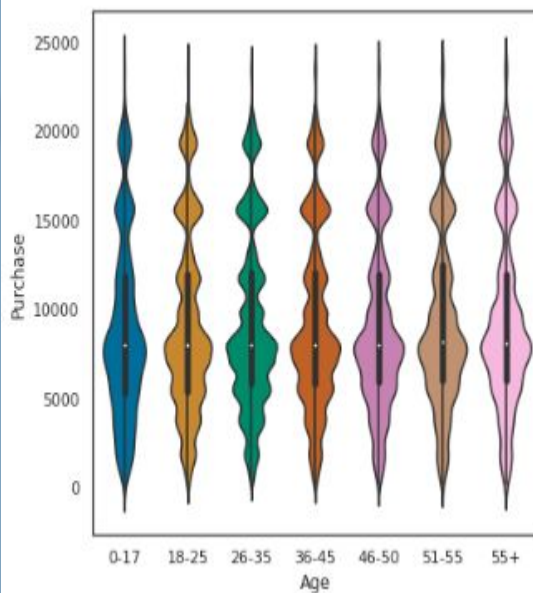


Age

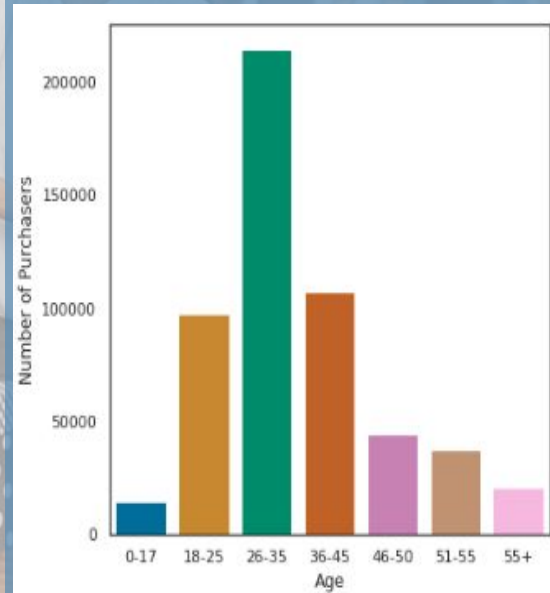
Age/Total Spending



Age/Spending Dist.



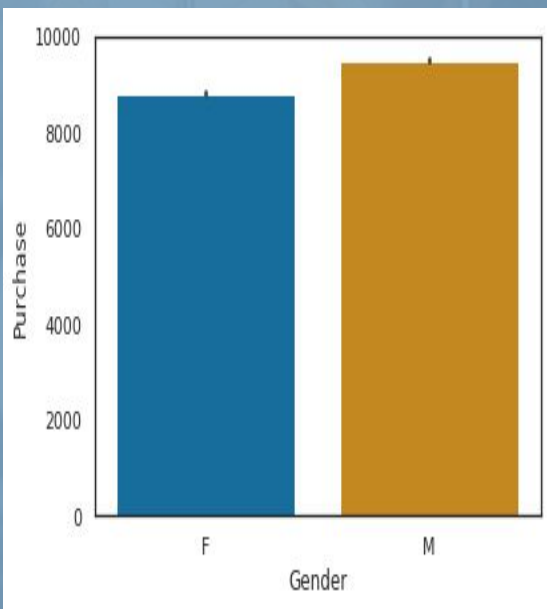
Age/No. Sampled



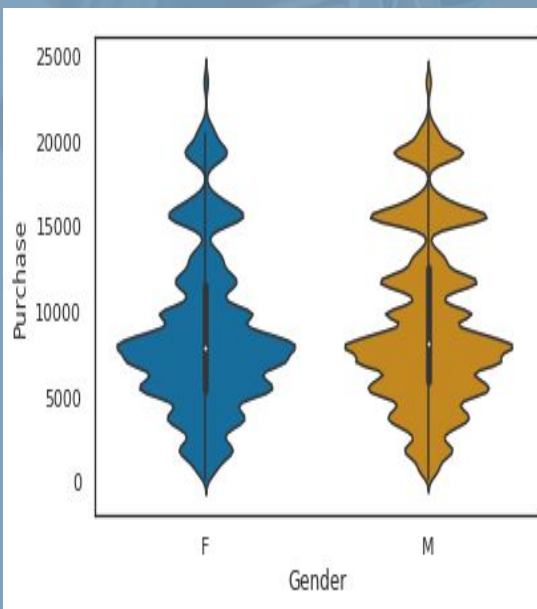


Gender

Gender/Total Spending



Gender/Spending Dist



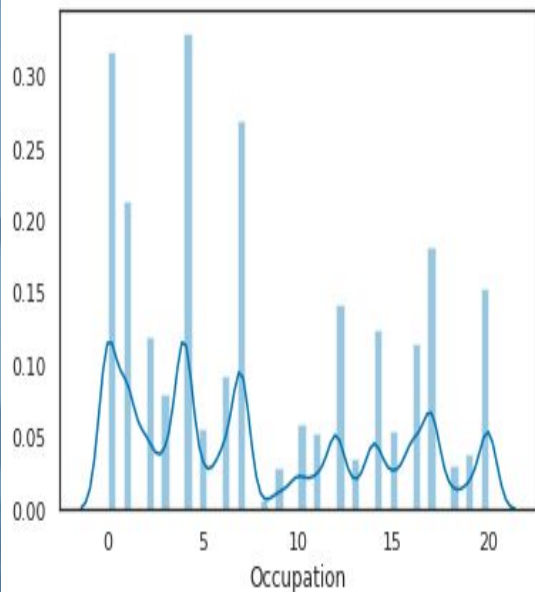
Observation

Men's Black Friday spending exceeded that of women by approximately \$1,000, with men outspending women in the \$15,000. to \$20,000. purchase range.

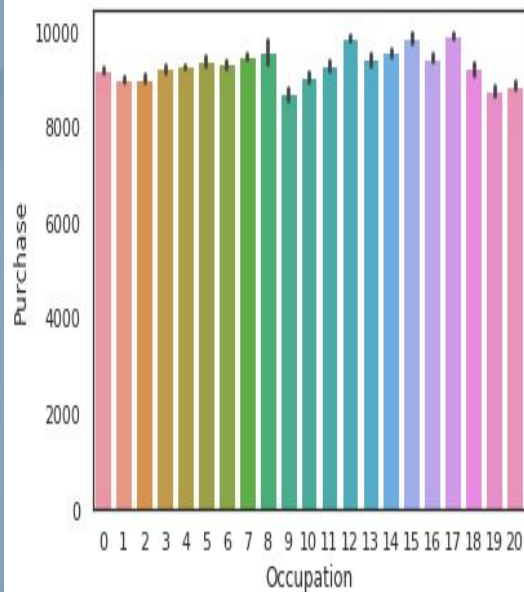


Occupation

Occupations



Occupations/Purchase

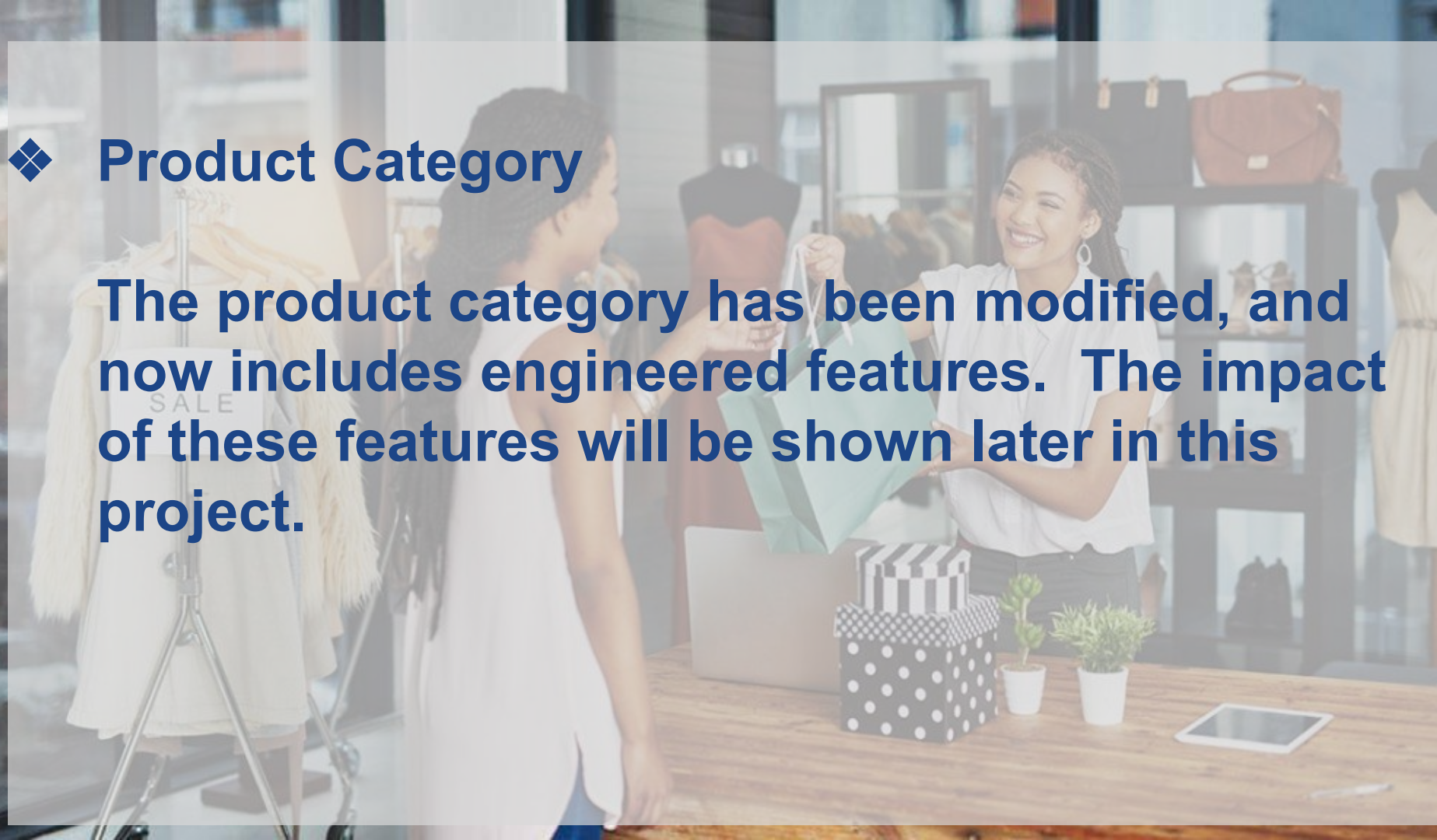


Observation

Though a significant number of people sampled fell into three occupations, spending seemed to be fairly evenly distributed across occupations. This may indicate that certain occupations with fewer practitioners may be vastly higher paying. Such groups may be a consideration target markets.

❖ Product Category

The product category has been modified, and now includes engineered features. The impact of these features will be shown later in this project.



Initial Regression Models Under Evaluation

K Nearest Neighbors

Ridge Regression

Random Forest

Total Data Set Use:
100%

Training Set: 80%
Test Set: 20%

Random Forest Regression

Max Depth	Estimators	Accuracy	Training Score	Test Score
100	1000	0.30	0.33137	0.31105

Ridge Regression

Alpha	Tol	Accuracy	Training Score	Test Score
1.0	0.001	0.27	0.26818	0.26780

K Nearest Neighbors Regression

Neighbors	Accuracy	Training Score	Test Score
100	0.24	0.26861	0.26003

Random Forest Regression

The preceding model evaluation included two engineered features, prod_1 and prod_2. The table below contains results after creating and adding 16 engineered features, prod_3 through prod_18.

(These features will be discussed in the slides that follow.)

Max Depth	Estimators	Accuracy	Training Score	Test Score
45	500	0.61	0.80688	0.61895
50	500	0.61	0.80708	0.61797
50	600	0.61	0.80889	0.61581
50	650	0.61	0.80892	0.61573

Original Product Category Features

**Product_Category_1, Product_Category_2,
and Product_Category_3**

The original product categories served as bins that showed the type classification (no. 1 through 18) of the first, second, and third, item purchased. Product_Category_2 and Product_Category_3 were populated with NaN, if the consumer only purchased one product.

Feature Limitations

The original features did not provide information on how many products of types 1 through 18 were bought by the total number of purchasers sampled. Consequently, there was no way to know, which products were the best sellers.

Engineered Features

prod_1

through

prod_18

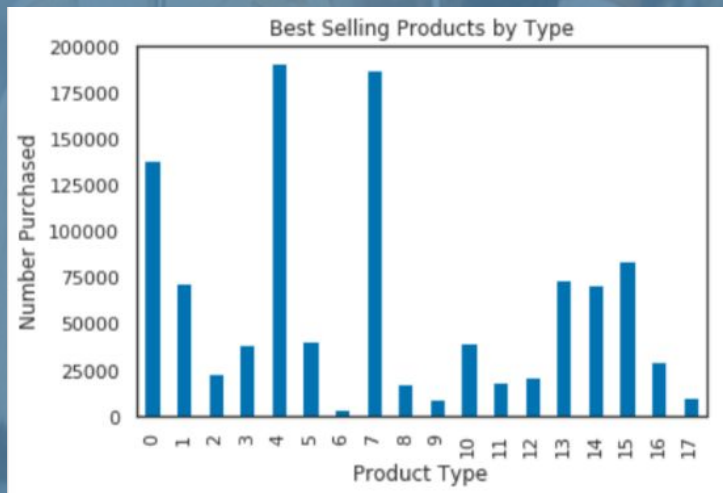
Feature Creation

```
df['prod_1'] = np.where(  
    (df['Product_Category_1'] == 1) |  
    (df['Product_Category_2'] == 1) |  
    (df['Product_Category_3'] == 1),  
    1, 0)
```

```
df['prod_1'] = np.where(  
    (df['Product_Category_1'] == 2) |  
    (df['Product_Category_2'] == 2) |  
    (df['Product_Category_3'] == 2),  
    1, 0)
```


Best Selling Product Types

Note: prod_1 through prod_18 are represented below as numbers 0 - 17, with number 17 representing prod_18.



Best sellers: prod_1, prod_5, and prod_8
Second best sellers: prod_2, prod_14, prod_15, and prod_16.

Feature Importance (Top 10)

Random Forest		Gradient Boosting	
prod_1	100.000	prod_1	100.000
Occupation	29.9262	prod_6	26.2529
prod_6	28.2286	prod_10	24.4100
prod_10	24.9854	prod_5	22.3356
prod_5	21.8158	prod_2	14.2918
prod_8	16.9898	prod_8	10.7028
prod_2	14.4675	prod_11	9.3354
prod_13	9.5785	prod_13	8.9488
prod_16	7.9415	prod_3	7.7442
prod_7	7.7751	prod_16	6.7844

Final Regression Models Under Evaluation

Random Forest

VS.

Gradient Boosting

Can a gradient boosting regressor reduce overfitting, while maintaining or improving accuracy?

Data Set Use:
10%

Training Set: 50%
Test Set: 50%

Random Forest Regression

Total Data Set Use: 10%, X, y Test Set Size: 50%

Data Set Includes: all engineered features

Max Depth	Estimators	Accuracy	Training Score	Test Score
50	650	0.59633	0.90093	0.59739
50	3000	0.60644	0.90114	0.59772
100	3000	0.60645	0.90336	0.58777

The model continues to grossly overfit; and, accuracy decreased when given a smaller portion of the original data set.

Gradient Boosting Regression

Total Data Set Use: 10%, X, y Test Set Size: 50%

Data Set Includes: all engineered features, learning rate: 10%

Max Depth	Estimators	Accuracy	Training Score	Test Score
2	2000	0.61140	0.621924	0.61093
3	3000	0.62640	0.67638	0.64580
3	2000	0.63576	0.67208	0.63894
3	1000	0.63505	0.65714	0.63443
5	1000	0.62128	0.74183	0.63189
50	1000	0.36060	0.93775	0.37318

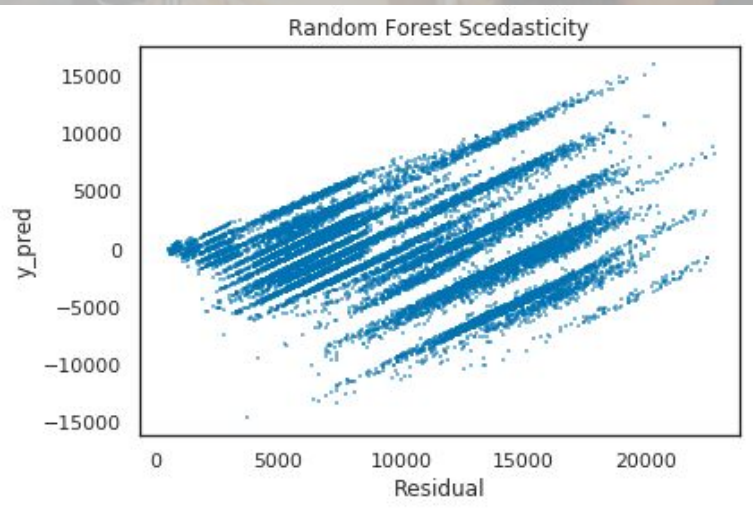
Gradient Boosting Regression

Total Data Set Use: 10%, X, y Test Set Size: 50%

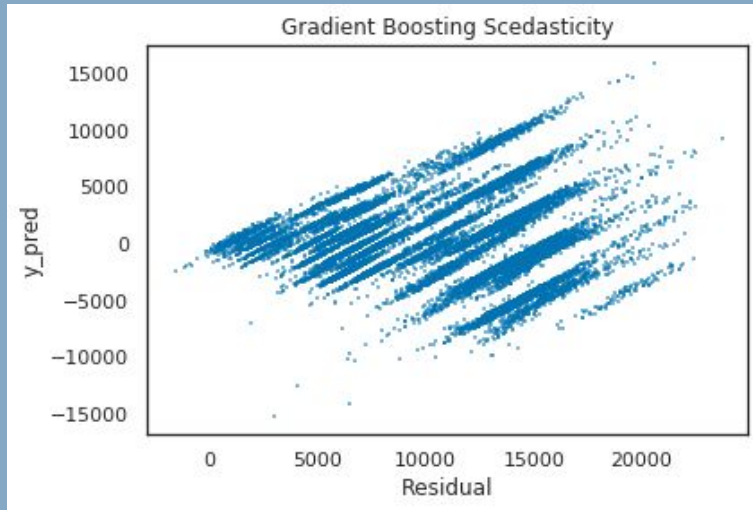
Data Set Includes: all engineered features, learning rate: variable

Max Depth	Estimators	Learning Rate	Accuracy	Training Score	Test Score
3	2000	0.15	0.63108	0.68239	0.63913
3	2500	0.15	0.63030	0.68686	0.63805
3	2000	0.20	0.62875	0.68951	0.63766
3	2000	0.05	0.63528	0.65540	0.63235

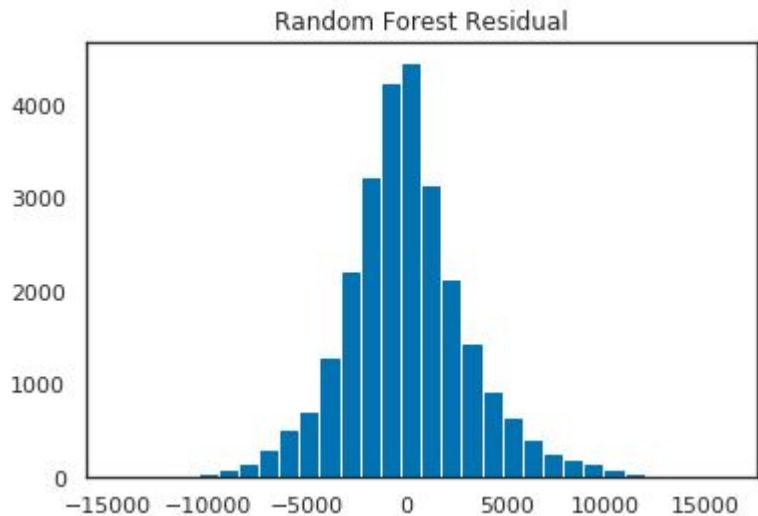
Random Forest Scedasticity



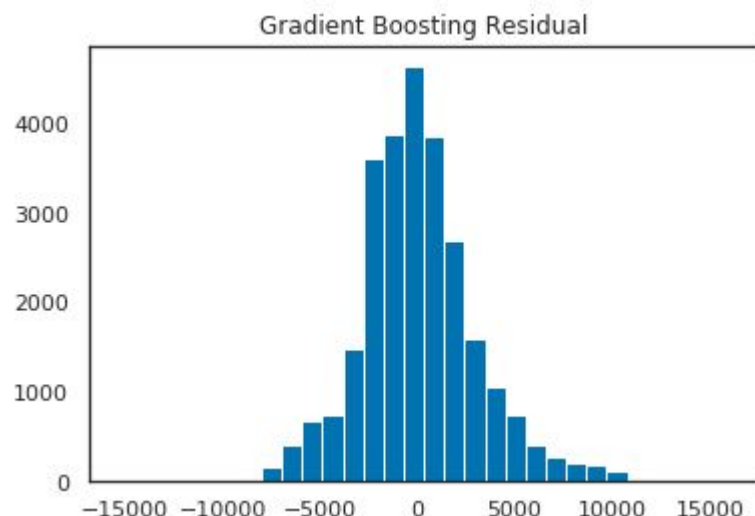
Gradient Boosting Scedasticity



Random Forest Residual



Gradient Boosting Residual





Things to work on: Make `df2['Occupation']` dummies, then run the model, and try to remedy the skedacity issue.

In the gradient boosting model, use only features with an importance of ≤ 1.0000

Maybe eliminate some of the lower scoring features in the random forest model, maybe not. Also reduce the max depth. This may keep the model from using insignificant features. It may also reduce overfitting due to noise.



Project Author:
Katrina Lanae Johnson

LinkedIn:
<https://www.linkedin.com/in/katrina-johnson-87891b22/>

Your feedback is welcome,
and appreciated!