Predicting Sales of Summer Clothes in e-Commerce platform - Wish

Presentation By :
Effy Paulose

CONTENT

ABSTRACT

- ✓ Over the past few years, online shopping has gradually become the mainstream shopping method. More and more local retailers chose to start their businesses on e-commerce platforms. However, few can survive due to the competitive pressure from the big companies and the entry barriers.

- ✓ This project identifies product listing strategies, primarily visual and textual presentation, that can help retailers to raise their product sales.

- ✓ To achieve that, we build Machine Learning Algorithms such as Linear Regression, polynomial Regression, SVR, Decision Forest Regression, random Forest Regression, and used VotingRegressor to boost the results.

INTRODUCTION

✓ **ORGANIZATIONAL BACKGROUND**

Wish is an American online e-commerce platform for transactions between sellers and buyers founded in 2010. The platform personalizes the shopping experience visually for each customer, rather than relying only on a search bar format. It allows sellers to list their products on Wish and sell directly to consumers.

✓ **CHALLENGES FACED BY THE ORGANIZATION**

Due to the non-tactile nature of online products, to attract customers and promote their sales, e-commerce vendors rely more on an alternative group of presented visual and textual information such as product images and titles.

✓ **PROBLEM STATEMENT**

Task is to predict the number of units sold for the sales of Summer Cloth in Ecommerce Wish.

✓ **RESOURCE REQUIREMENT**

Sales of summer clothes in E-commerce Wish - Dataset contains product listings as well as product ratings and sales performance collected from Kaggle. With this, the correlations and patterns regarding the success of a product and its various components can be studied.

AIM OF THE POJECT

01

To analyze the dataset based on units sold of summer clothes

02

To see the patterns on sales of summer clothes

03

To know what are the factors that will affect the sales

04

Predict the number of units sold of the products.

OBJECTIVE

Converting data into an appropriate form using various preprocessing techniques

To understand the relationship, visualize and identify the pattern between selected attributes that affect the unit sold of summer clothes

To predict the number of units sold

To determine the appropriate Machine Learning algorithm for sales forecasting.

Selecting various metrics to compare the performance of the applied Machine Learning algorithms.

TEAM BACKGROUND AND SKILLS

This project identifies product listing strategies, primarily visual and textual presentation, that can help retailers to raise their product sales. To achieve that, we build Machine Learning Algorithms.

Statistics: Analysis of variance and hypothesis testing

Probability: Helps in predicting future consequences

Data Modeling: Analyze the unstructured data models, identifying the underlying data structures, finding out the patterns, and filling the gaps between the places where data is nonexistent

Programming Fundamentals: Strong basic fundamental skills such as computer architecture, algorithms, data structures, complexity, etc.

ML Libraries & Algorithms: Using the algorithms and libraries that are developed by other developers and organizations

Software Design: Develop algorithms and systems that can easily integrate and communicate with the other existing technologies

ML Programming Languages: Python: Python is equipped with a wide range of useful libraries which help in processing data efficiently and in scientific computing.

DATA OVERVIEW

- ✓ In this project, there are labeled sales data from summer clothes from different merchants that provide information such as item type, item price, shipping option, merchant id, etc.
- ✓ These data were extracted from Kaggle and will be used to train and improve the model for Machine Learning.
- ✓ In the dataset being analyzed, there are 1573 instances and 42 attributes. The dataset has been properly divided into training and testing data to build the model.

METHODOLOGY

- Problem Statement: Predict the number of units sold
- Dataset Collection
- Data Preprocessing
- Data Analysis
- Feature Selection
- Build and Testing a training model for prediction
- Comparison of various learning models based on the results
- Prediction of the sales and Testing

DATASET COLLECTION

Collected the dataset from Kaggle for sales of Summer Cloth in Ecommerce Wish

	title	title_orig	price	retail_price	currency_buyer	units_sold	uses_ad_boosts	rating	rating_count
0	2020 Summer Vintage Flamingo Print Pajamas Se...	2020 Summer Vintage Flamingo Print Pajamas Se...	16.00	14	EUR	100	0	3.76	54
1	SSHOUSE Summer Casual Sleeveless Soirée Party ...	Women's Casual Summer Sleeveless Sexy Mini Dress	8.00	22	EUR	20000	1	3.45	6135
2	2020 Nouvelle Arrivée Femmes Printemps et Été ...	2020 New Arrival Women Spring and Summer Beach...	8.00	43	EUR	100	0	3.57	14
3	Hot Summer Cool T-shirt pour les femmes Mode T...	Hot Summer Cool T Shirt for Women Fashion Tops...	8.00	8	EUR	5000	1	4.03	579

DATA PRE-PROCESSING

- ✓ This is a key step to making models that can predict/classify depending on the dataset and the question aim to be answered.
- ✓ Requires to be aware of the background of the data and the question.
- ✓ These are a few steps that are used at the Data Preprocessing stage.

1. Removing the null values
2. Transform categorical variables
3. Removing the features that have 1 unique value
4. Engineer new feature
5. Remove unnecessary features

✓Removing the null values

- All the 5 rating count features with null values are replaced with 0 since the value could be null (for that rating) because no customer rated it.

- 'has_urgency_banner' feature tells us whether or not the product has an urgency banner. Therefore, this feature becomes a categorical variable:
    - 1 denoting it has an urgency text
    - 0 denoting it does not have an urgency text (Replace null with 0)

✓Transform categorical variables

Curate and reduce the different values that are present in features so there is some uniformity in the dataset and the sparsity is reduced.

1. **Product Color**

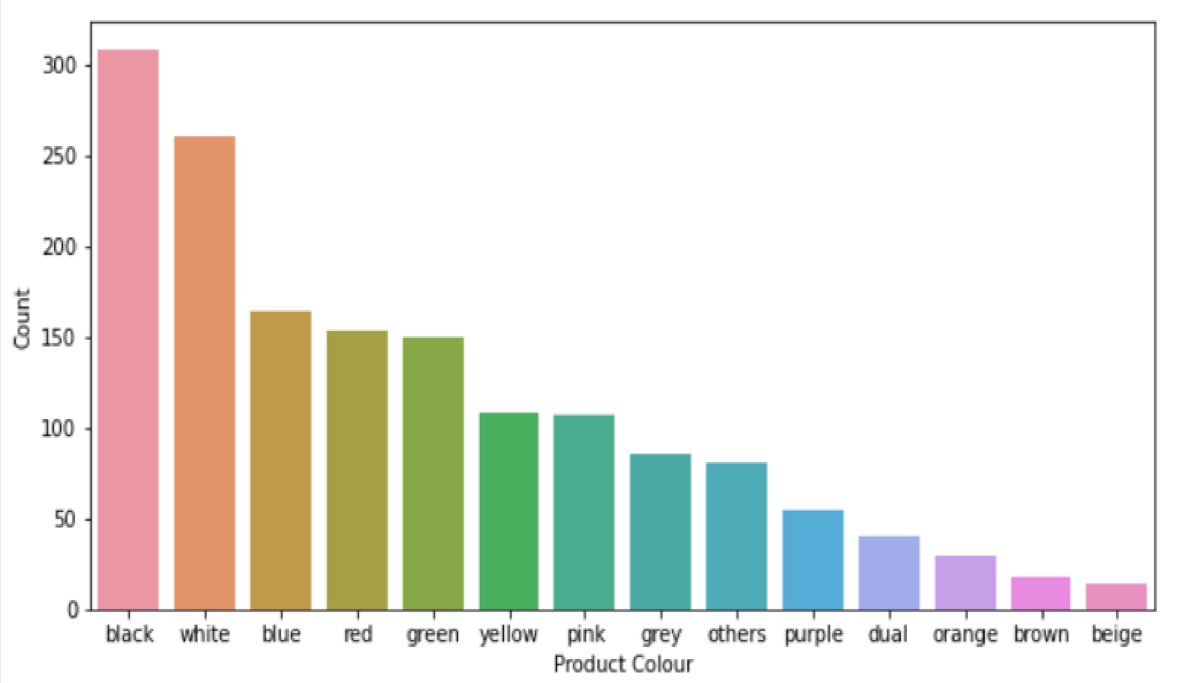
- 'product_color', 'product_variation_size_id' and 'origin_country' will be used in this category to reduce the number of categories in each feature

- segregate different colors into basic colors - 'black', 'white', 'blue', 'red', 'green', 'yellow', 'pink', 'grey', 'purple', 'orange', 'brown', 'beige' are the basic colors opted

- replaced np.nan with 'others'

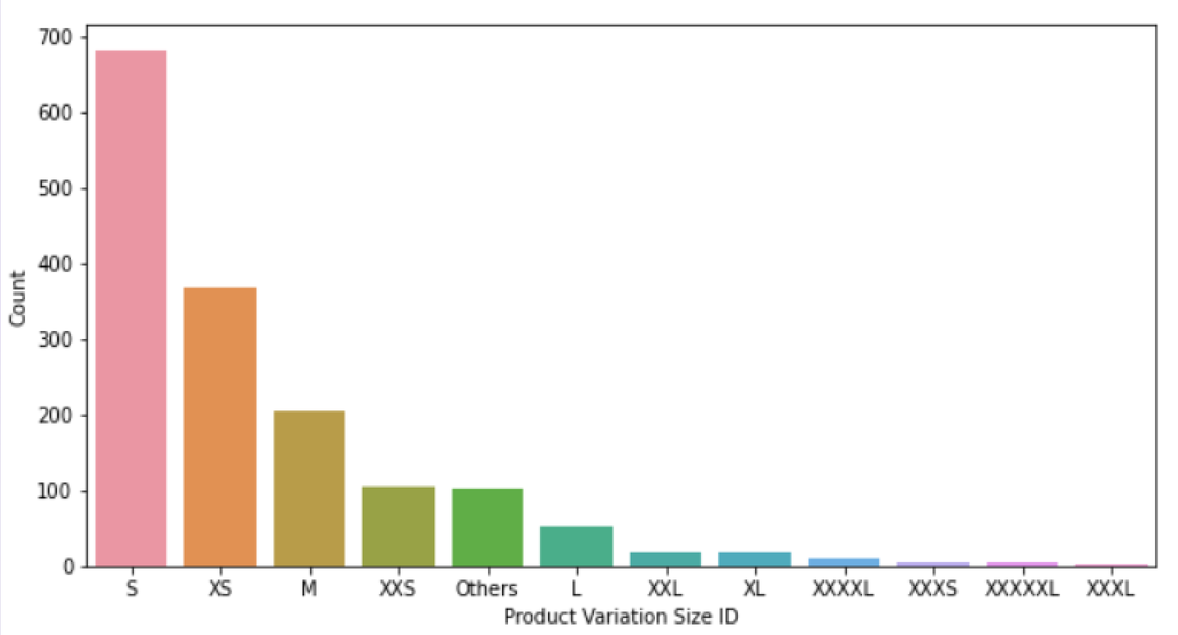
- categories opt adding a category 'dual' for products that have two colors

- adding a category 'others' for products that are multi-colored or have a print on them



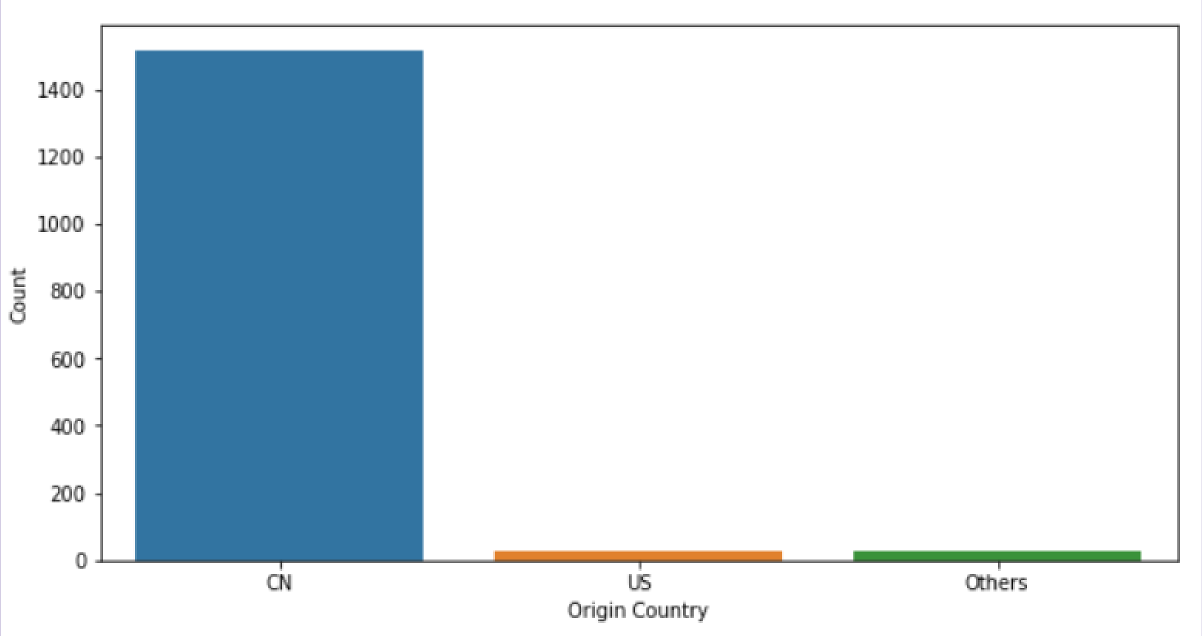
2. **Product Variation Size ID**

- The categories opted are XXXS, XXS, XS, S, M, L, XL, XXL, XXXL, XXXXL, XXXXXL, Others
- All null values will be under the category 'Others'



3. **Origin country**

- The categories opted are 'CN', 'US' and 'Others'(VE, SG, AT and GB).
- All the null values are categorized under 'Others'



✓ Removing the features that have 1 unique value

- Columns with only 1 unique value will not add value to the model, hence dropping them out.

✓ Engineer new feature

- Importing “unique-categories.sorted-by-count.csv” that has the unique categories of tags sorted by count.
- Aim: To find out the percentage the of total number of tags available for a particular product.
- New feature will be 'tags_percentage'.
- More tags a product has, the more it will turn up in searches. Hence the probability of its units being sold will be high.
- Dropping the 'tags' feature because it is not needed for the model.

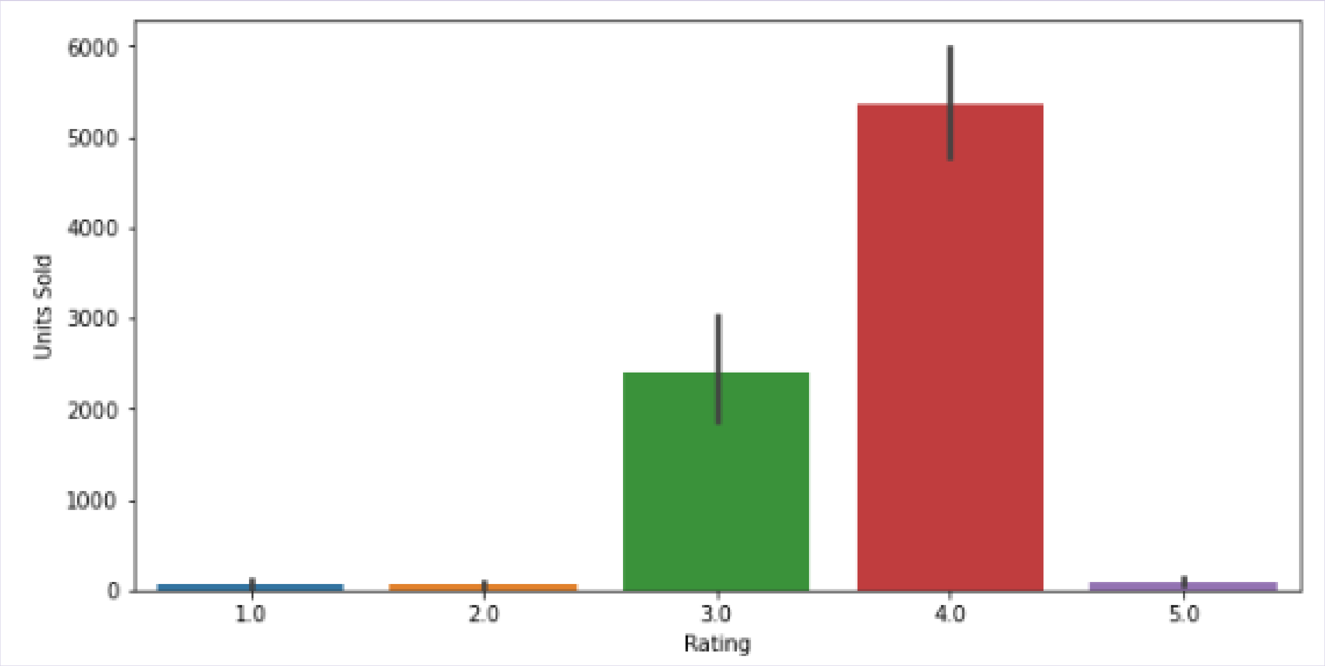
✓ Remove unnecessary features

- Columns: title, title_orig, merchant_profile_picture, product_url, product_picture, product_id, merchant_id, merchant_info_subtitle, merchant_name, merchant_title, shipping_option_name, urgency_text
- These will be dropped for now, as the likelihood of these affecting the number of units sold is less. For some of the features present above, a corresponding feature already exists in the dataset that provides more relevant information.
- The rating_count will also be removed since features of the distribution of rating count across (5/4/3/2/1) gives more detailed information than 'rating count'

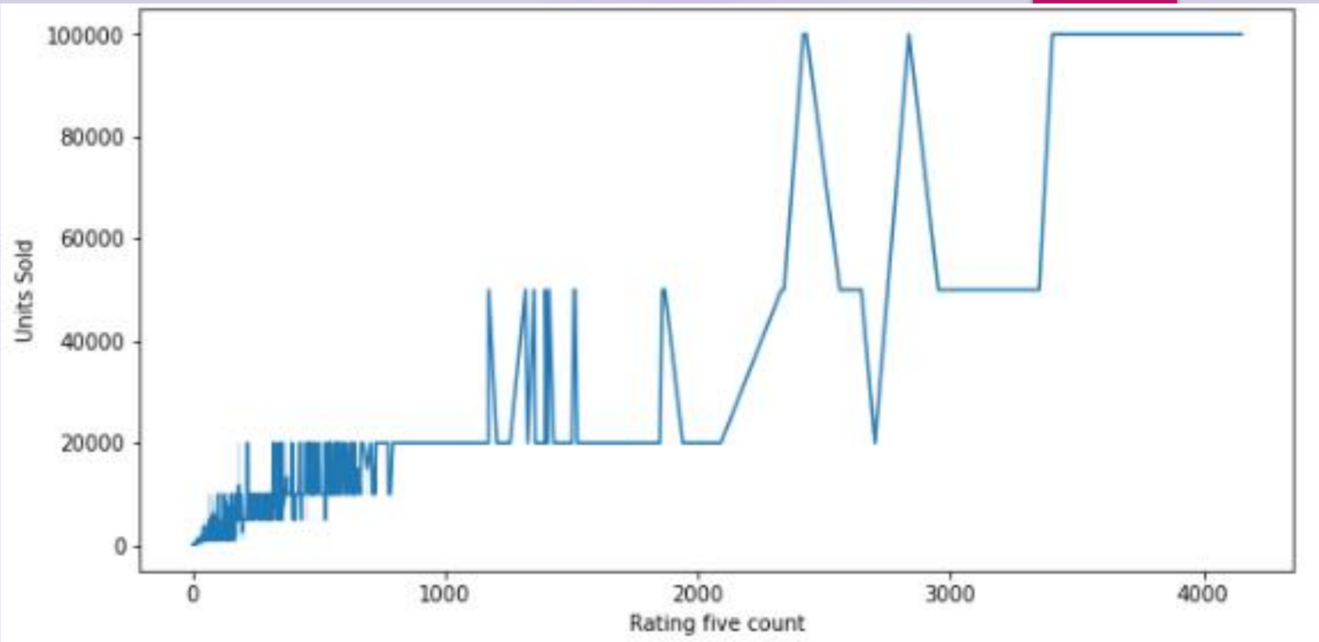
DATA ANALYSIS

Relationship between Ratings and Units sold

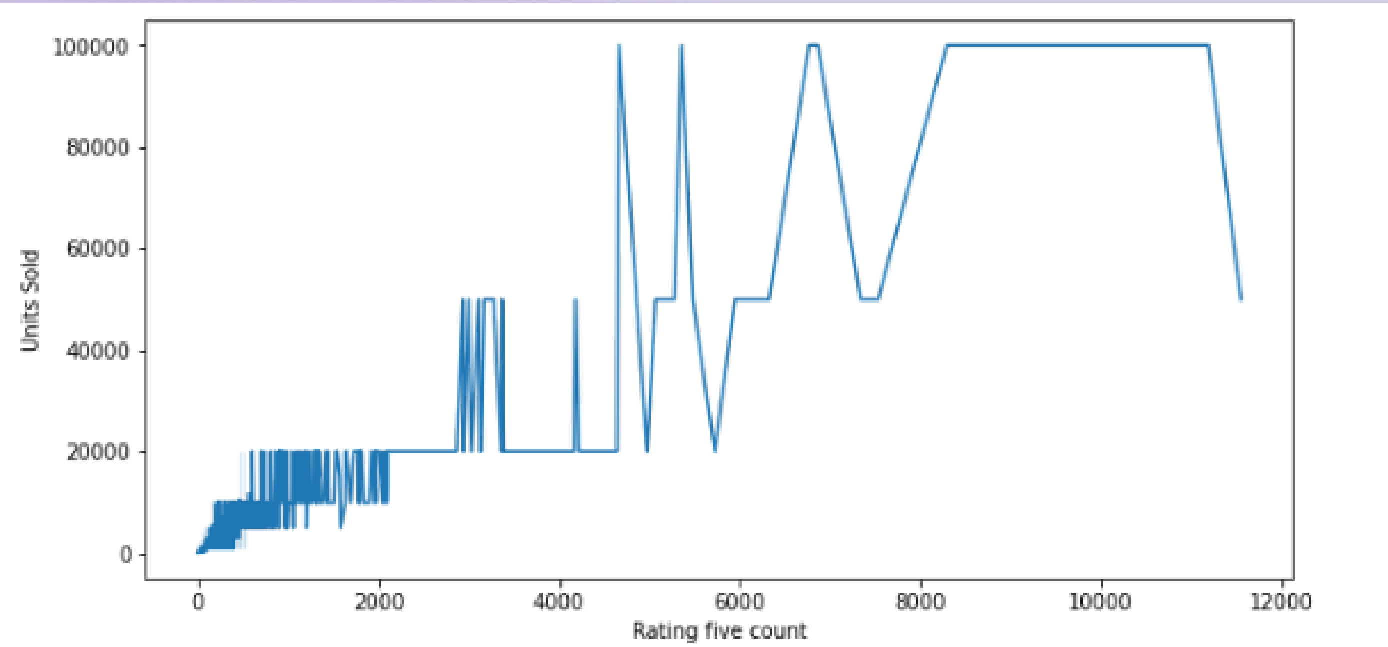
- ✓ Figure shows the effect of ratings to boost the units sold. The maximum sales occurred when the rating is 4 stars (good). Customers require high-quality of products sold



Relationship between 5-star-rating and Units sold



✓ Figure shows the effect of 5-star-rating and Units sold. From the chart, the seller gets higher sales as the number of 5-star increases

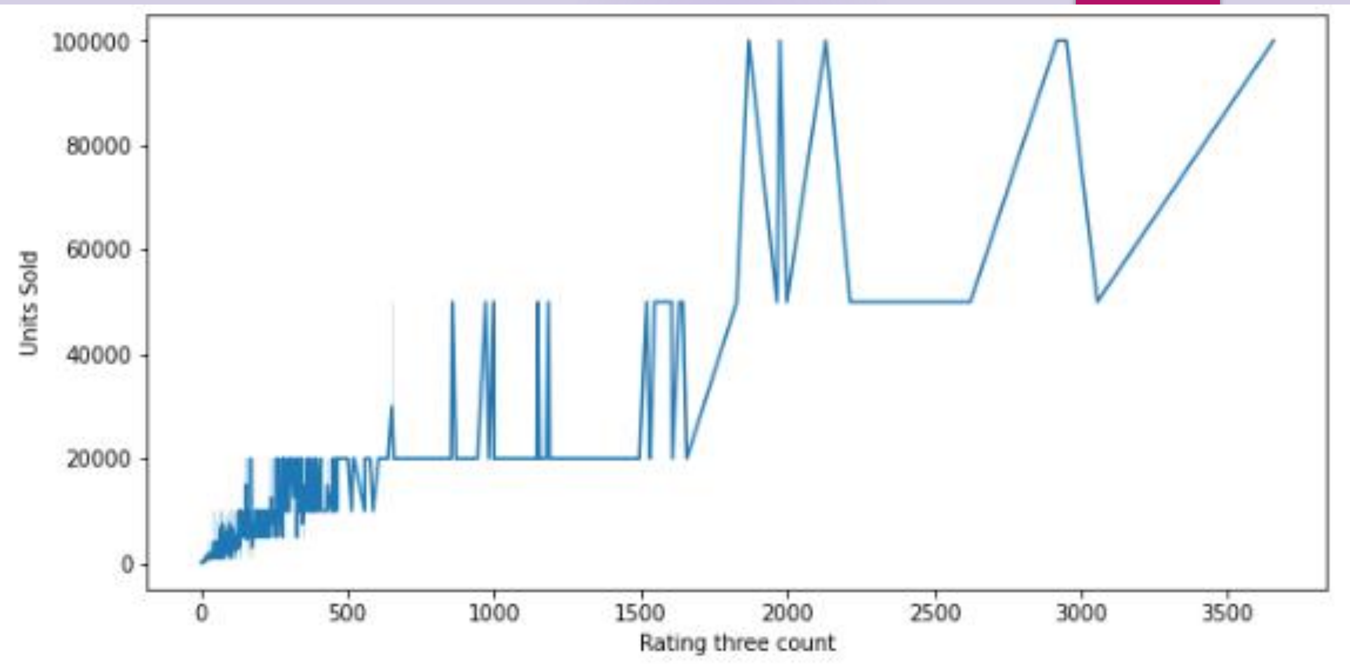


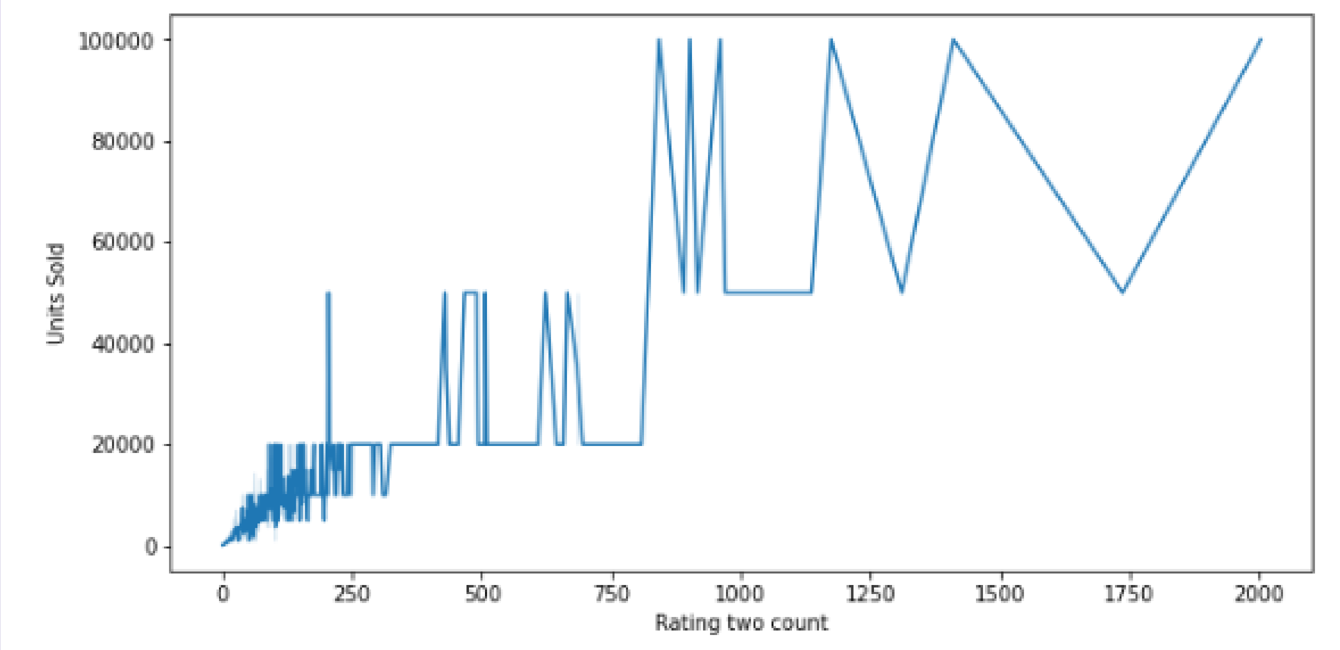
Relationship between 4-star-rating and Units sold

✓ Figure shows the effect of 4-star-rating and Units sold. From the chart, the seller gets higher sales as the number of 4-star increases

Relationship between 3-star-rating and Units sold

✓ Figure shows the effect of 3-star-rating and Units sold. From the chart, the seller gets higher sales as the number of 3-star increases



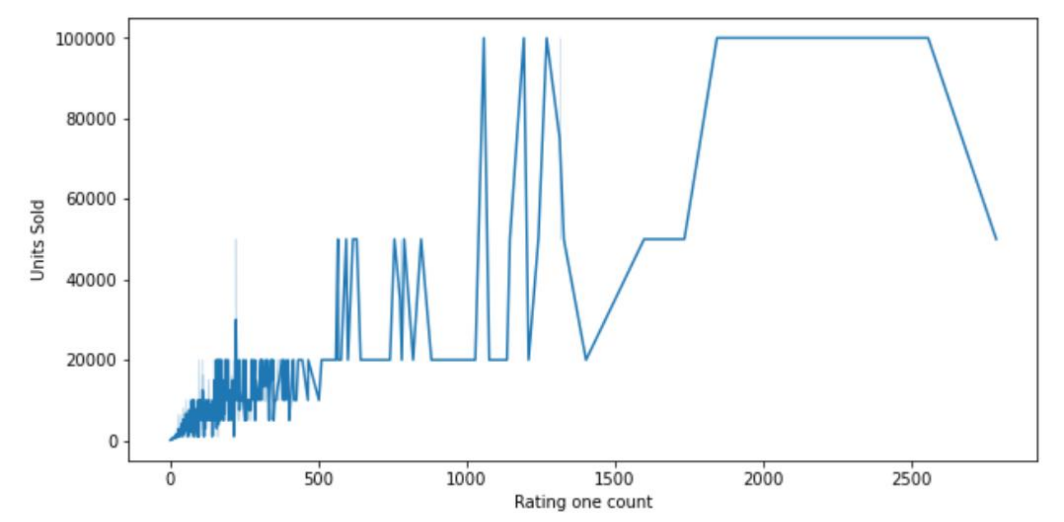


Relationship between 2-star-rating and Units sold

✓ Figure shows the effect of 2-star-rating and Units sold. From the chart, the seller gets higher sales as the number of 2-star increases

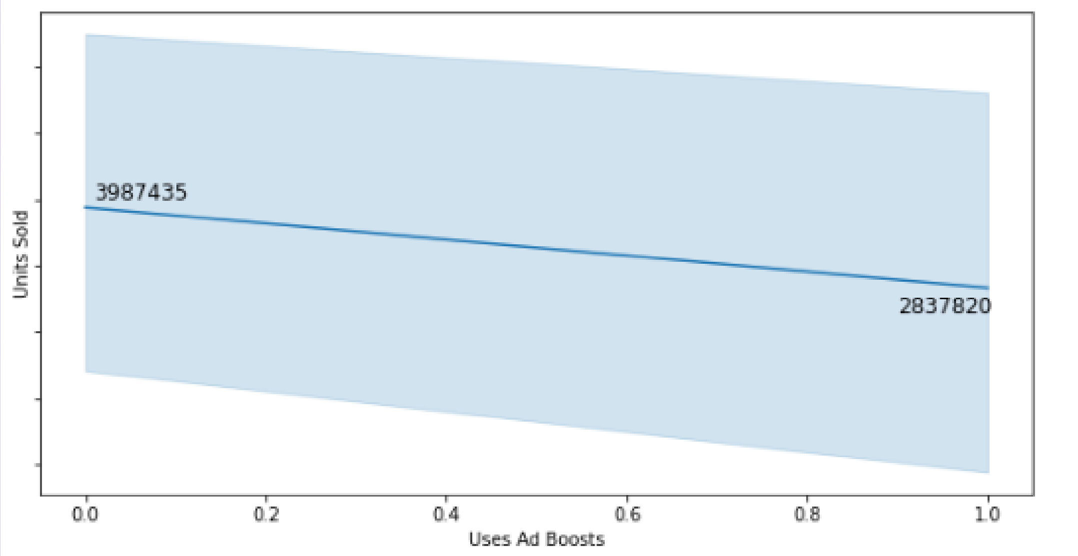
Relationship between 1-star-rating and Units sold

- ✓ Figure shows the effect of 1-star-rating and Units sold. From the chart, the seller gets higher sales as the number of 1-star increases



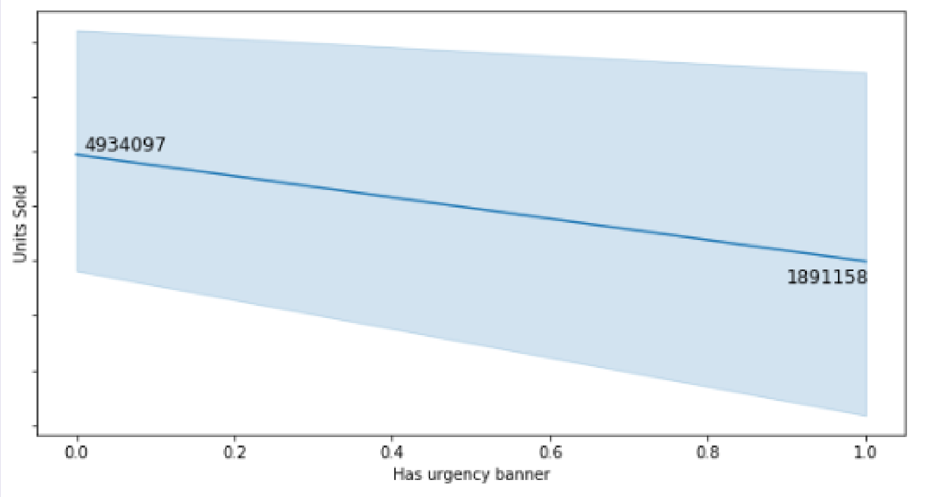
Relationship between Uses of Ad boosts and Units sold

- ✓ Figure shows the effect of using advertisements to boost the units sold. From the chart, the seller gets higher sales without using advertisements.



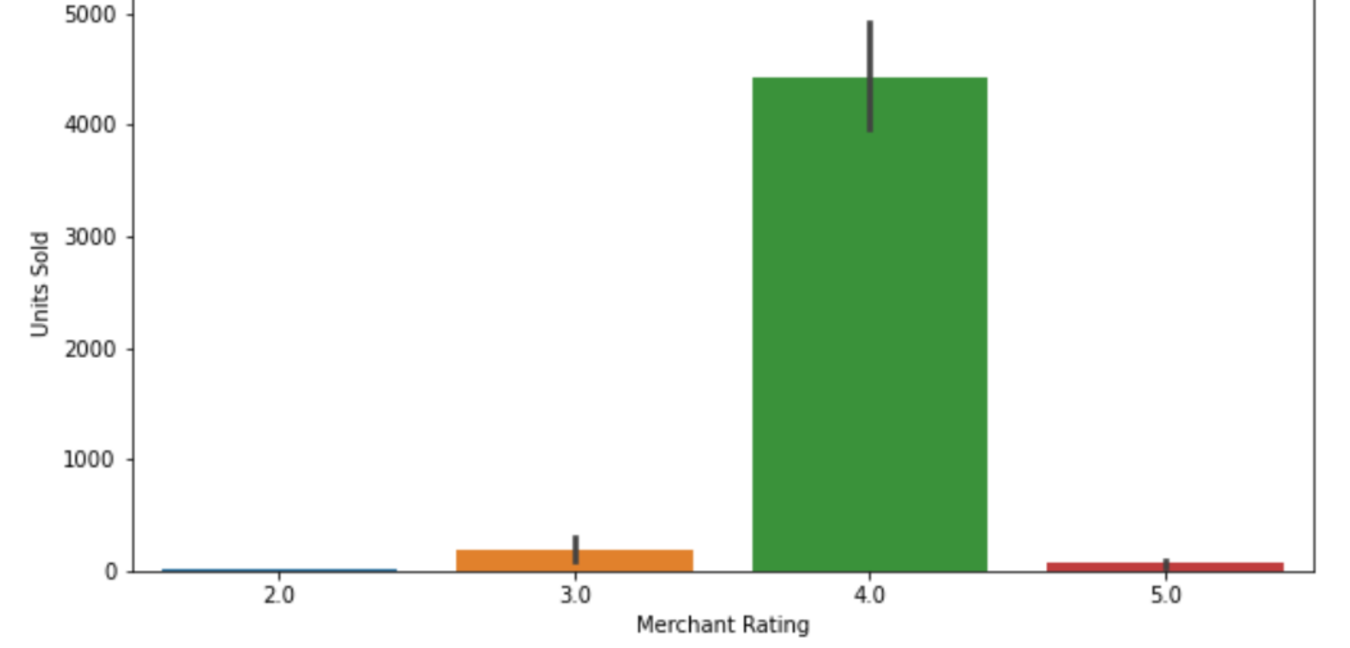
Relationship between urgency banner and Units sold

- ✓ Figure shows the effect of the urgency banner and Units sold. From the chart, the seller gets much higher sales if the urgency banner is not present



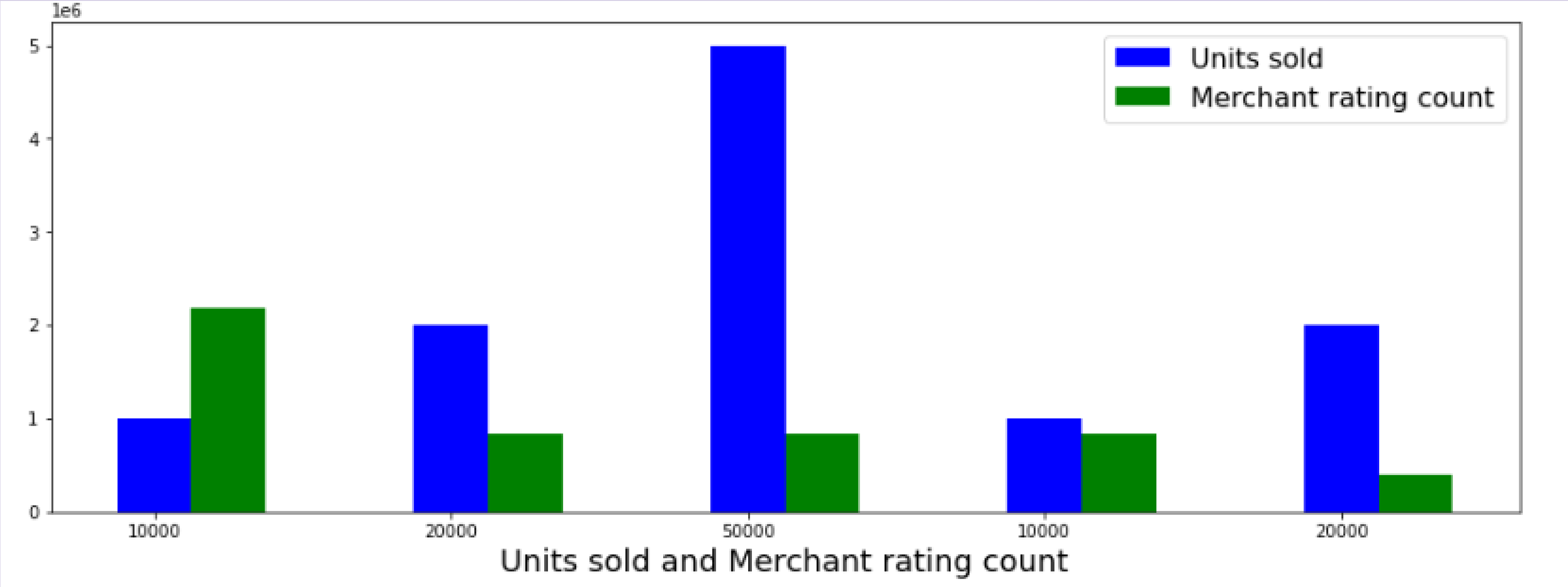
Relationship between Merchant Rating and Units sold

- ✓ Figure shows the effect of merchant ratings to boost the units sold. The maximum sales occurred when the merchant rating is 4 stars (good).



Relationship between merchant rating count and Units sold

- ✓ Figure shows the effect of merchant rating count and Units sold. Merchant's rating count is important in the seller's choice of purchase, but it is not the final factor.



Relationship between Product Color, Product Size, and Origin Country with Units sold

- ✓ Figure 1 shows that black colored clothes have the most sales
- ✓ Figure 2 shows the Small sized clothes have the most sales
- ✓ Figure 3 shows the clothes made in the country China has the most sales

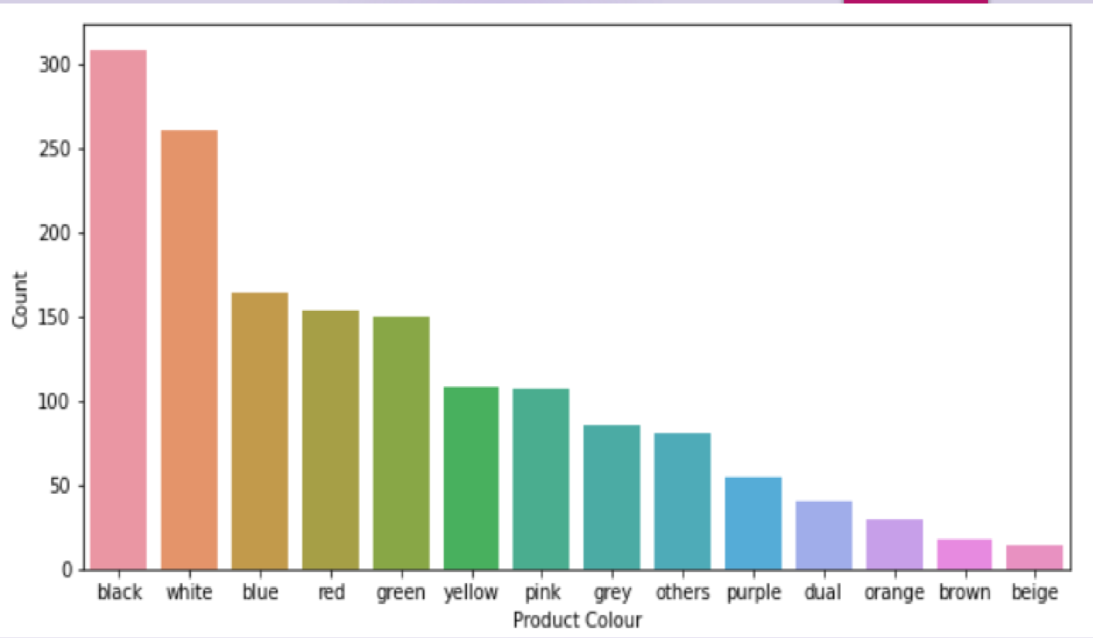


Figure 1

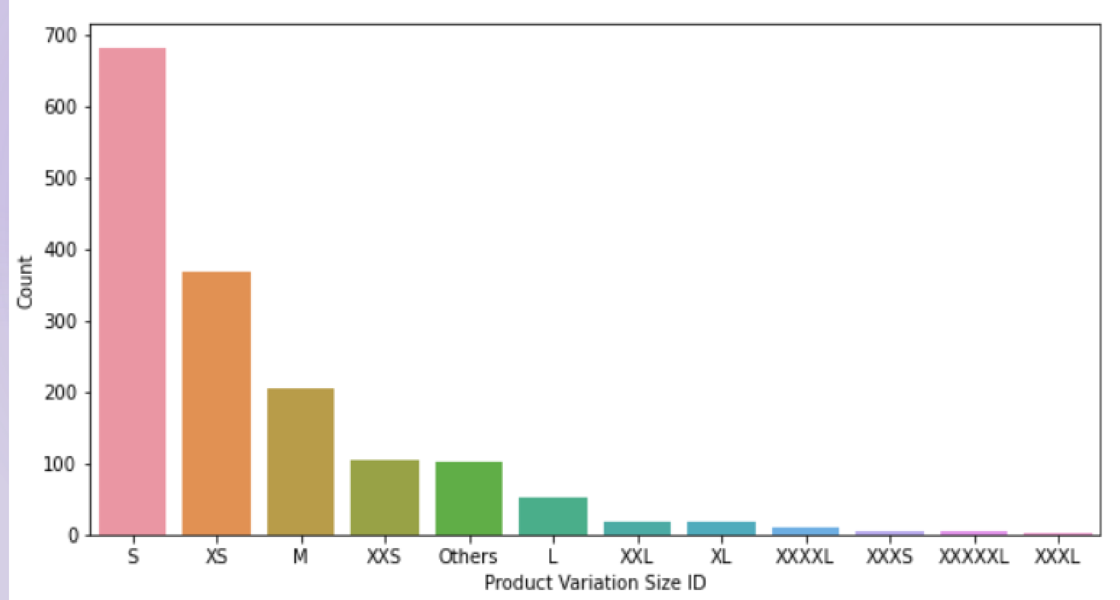


Figure 2

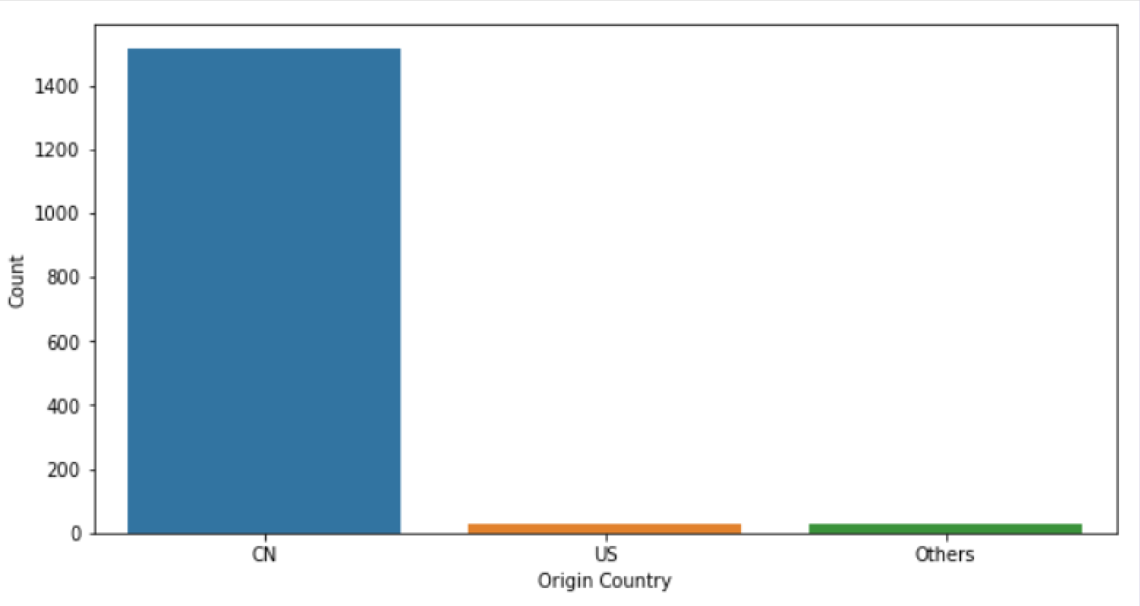


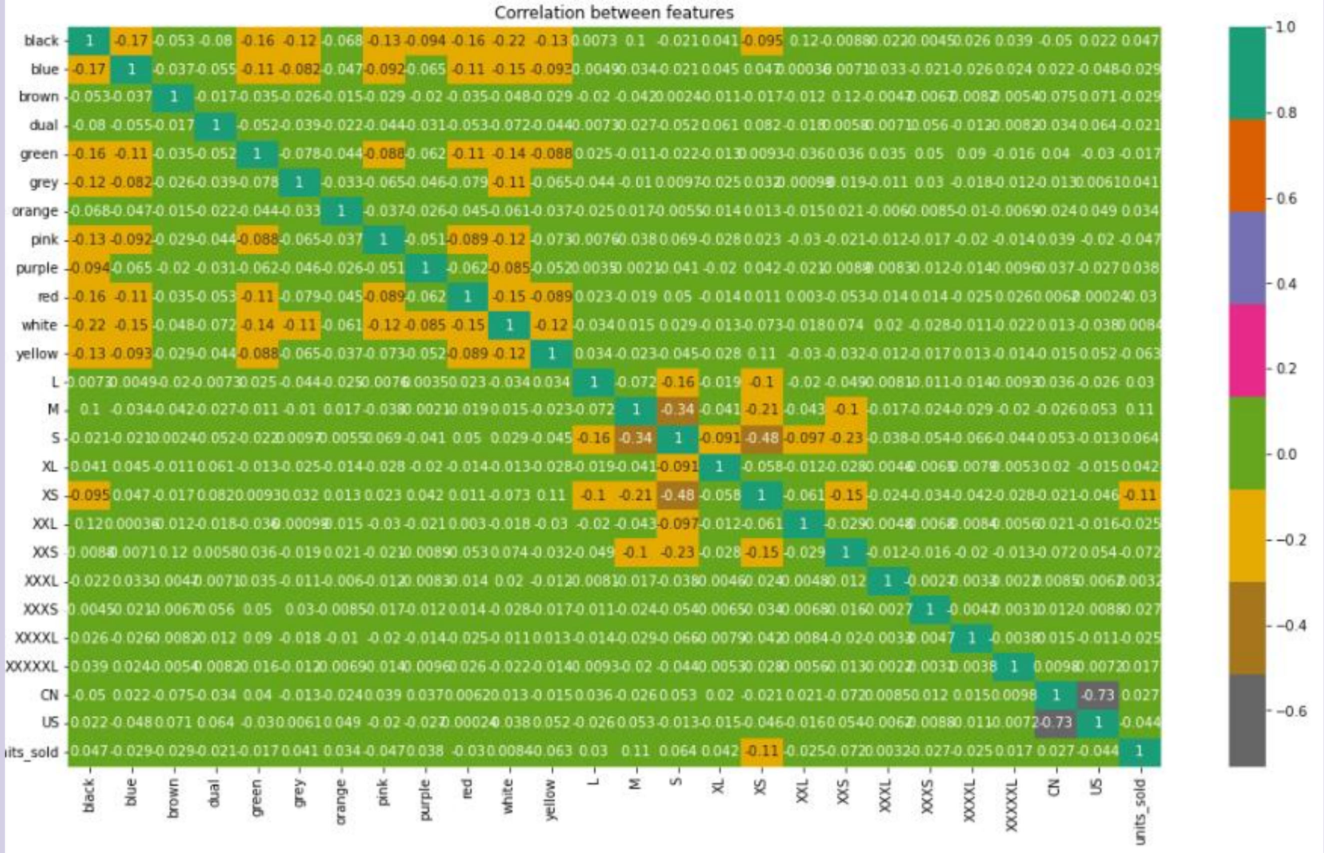
Figure 3

CORRELATION BETWEEN FEATURES

- ✓ Checking three categorical variables (product color, variation size, and origin country) of correlation using the one-hot encoded format with the units sold.

```python
feat_onehot_corr = feat_onehot.corr()

feat_onehot_corr['units_sold'].sort_values(ascending=False)
```

```
units_sold    1.000000
M             0.107101
S             0.063655
black         0.046767
XL            0.042029
grey          0.041432
purple        0.037679
orange        0.034395
L             0.029851
CN            0.026664
XXXXXL        0.017198
white         0.008433
XXXL          0.003245
green        -0.016570
dual         -0.021030
XXL          -0.024689
XXXXL        -0.024773
XXXS         -0.027051
blue         -0.028643
brown        -0.028790
red          -0.030292
US           -0.044473
pink         -0.046868
yellow       -0.063270
XXS          -0.072489
XS           -0.112293
Name: units_sold, dtype: float64
```

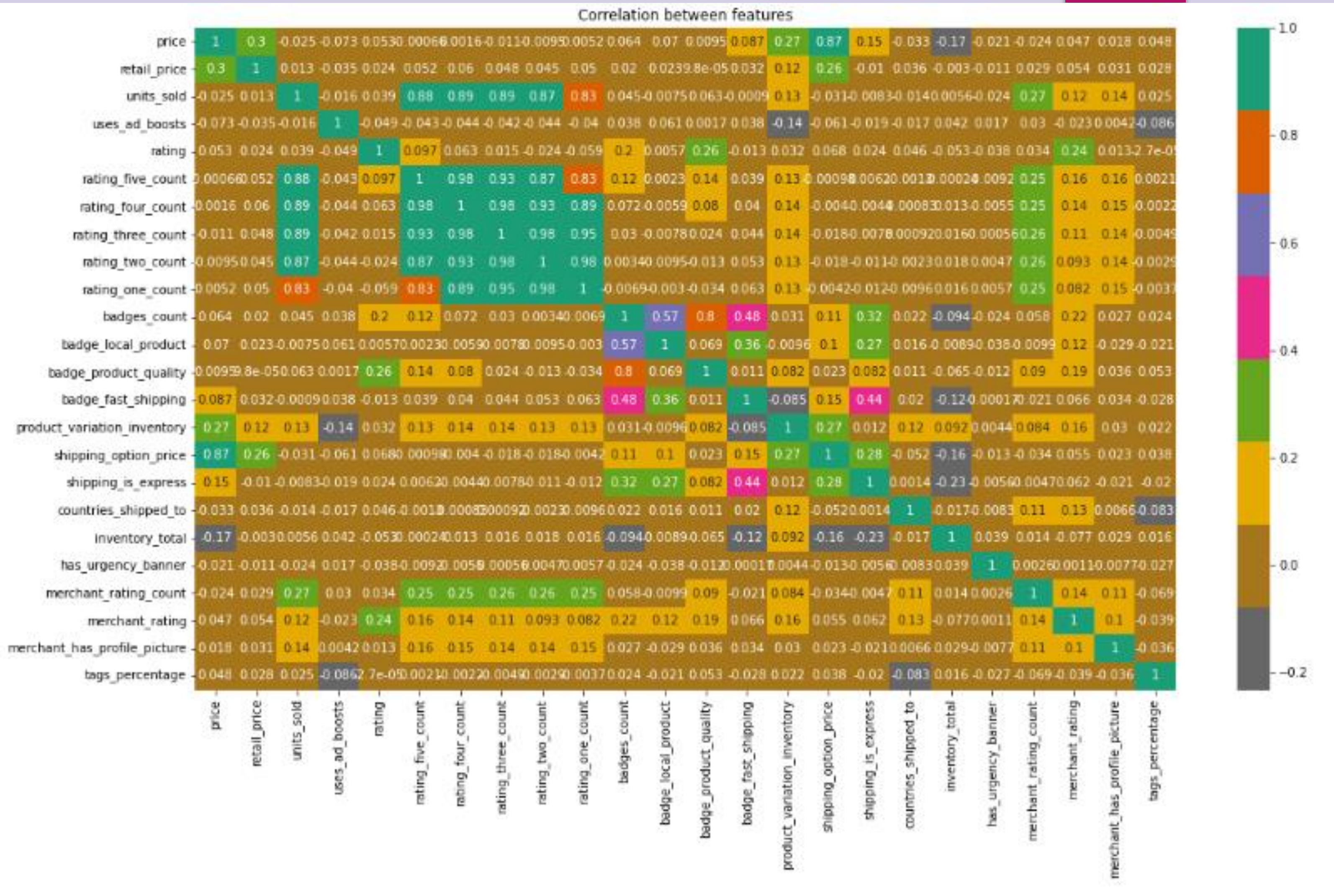


- ✓ From the above result we can safely say that the dependency of units sold on the product color, variation size or origin country is very unlikely.
- ✓ For the same reason, we will DROP these three features.

CORRELATION BETWEEN OTHER FEATURES

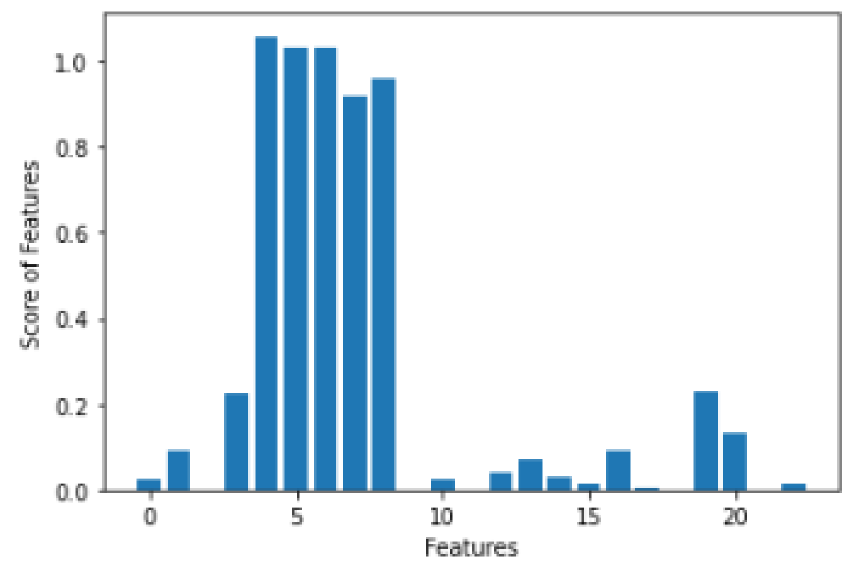
```
sales_corr['units_sold'].sort_values(ascending=False)
```

```
units_sold                      1.000000
rating_three_count              0.894835
rating_four_count               0.891761
rating_five_count               0.876972
rating_two_count                0.867406
rating_one_count                0.833807
merchant_rating_count           0.272897
merchant_has_profile_picture    0.143529
product_variation_inventory     0.133846
merchant_rating                 0.122504
badge_product_quality           0.063187
badges_count                    0.045402
rating                          0.039478
tags_percentage                 0.025363
retail_price                    0.012638
inventory_total                 0.005608
badge_fast_shipping            -0.000898
badge_local_product            -0.007544
shipping_is_express            -0.008308
countries_shipped_to           -0.013553
uses_ad_boosts                 -0.016055
has_urgency_banner             -0.023891
price                          -0.024815
shipping_option_price          -0.030987
Name: units_sold, dtype: float64
```



FEATURE SELECTION

- ✓ Feature Selection is done to get the best features that would help in predictions.
- ✓ Using the **SelectKBest** method to capture the best features for the model. It selects features according to the k highest scores.
- ✓ Scoring function used here is **Mutual Info Regression.**
    - Mutual Information Regression: It is between two random variables is a non-negative value, which measures the dependency between the variables. It is equal to zero if and only if two random variables are independent, and higher values mean higher dependency. It can capture any type of dependency between variables.
    - The function relies on nonparametric methods based on entropy estimation from k-nearest neighbors distances.

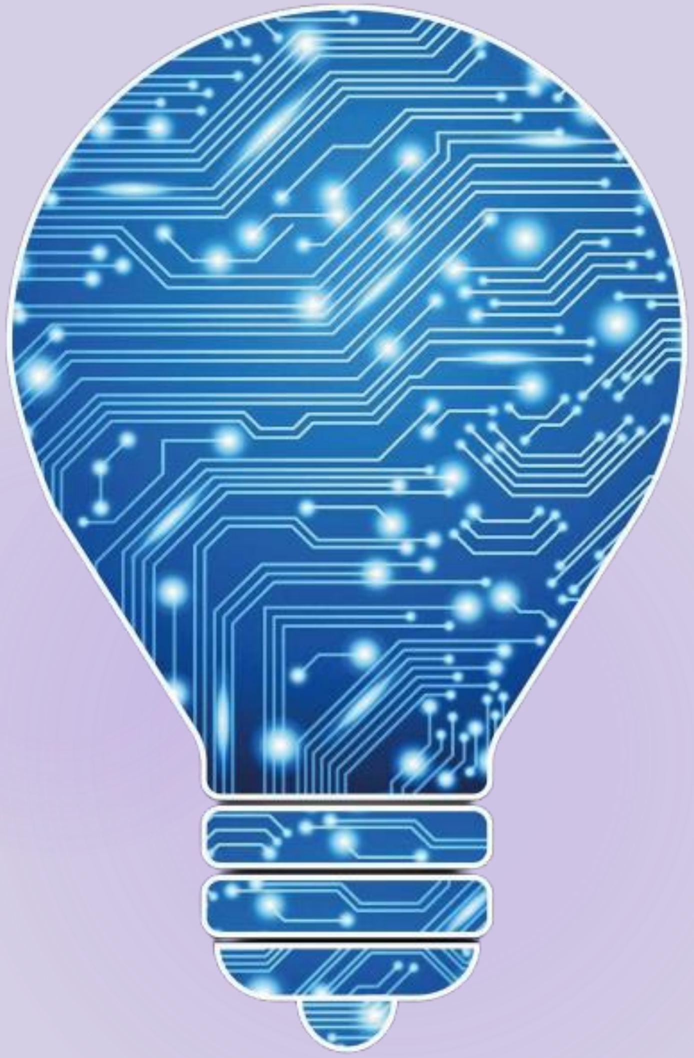


Best 8 features for model

```
a = fs.get_feature_names_out()

print('Best columns that we are using for our model\n')

for i in a:
    print(i)
```

```
Best columns that we are using for our model

rating
rating_five_count
rating_four_count
rating_three_count
rating_two_count
rating_one_count
merchant_rating_count
merchant_rating
```

MACHINE LEARNING TECHNIQUES



- ✓ Machine Learning is the area of study which enables machines to learn without being explicitly programmed. Machine Learning is defined as the computer program that learns from experience E with respect to some class of tasks T and performance measure P when its performance at tasks in T, as measured by P, strengthens with experience E.

- ✓ In general, Machine Learning is a program that can manage various tasks by analyzing and exploring data

- ✓ In this project, five different algorithms are used for analysis and comparison.

I . Linear Regression:

Analysis is used to predict the value of a variable based on the value of another variable.

II . Polynomial Regression:

Form of regression analysis in which the relationship between the independent variable x and the dependent variable y is modeled as an nth degree polynomial in x

III. Support Vector Regression Or SVR

SVR is a regression algorithm, used for working with continuous values instead of Classification which is SVM(Support vector machines).

IV. Decision Forest Regression

A decision Tree is a non-parametric model that performs a sequence of simple tests for each instance, traversing a binary tree data structure until a leaf node (decision) is reached. The decision Forest Regression model consists of an ensemble of decision trees. Each tree in a regression decision forest outputs a Gaussian distribution as a prediction. Aggregation is performed over the ensemble of trees to find a Gaussian distribution closest to the combined distribution for all trees in the model.

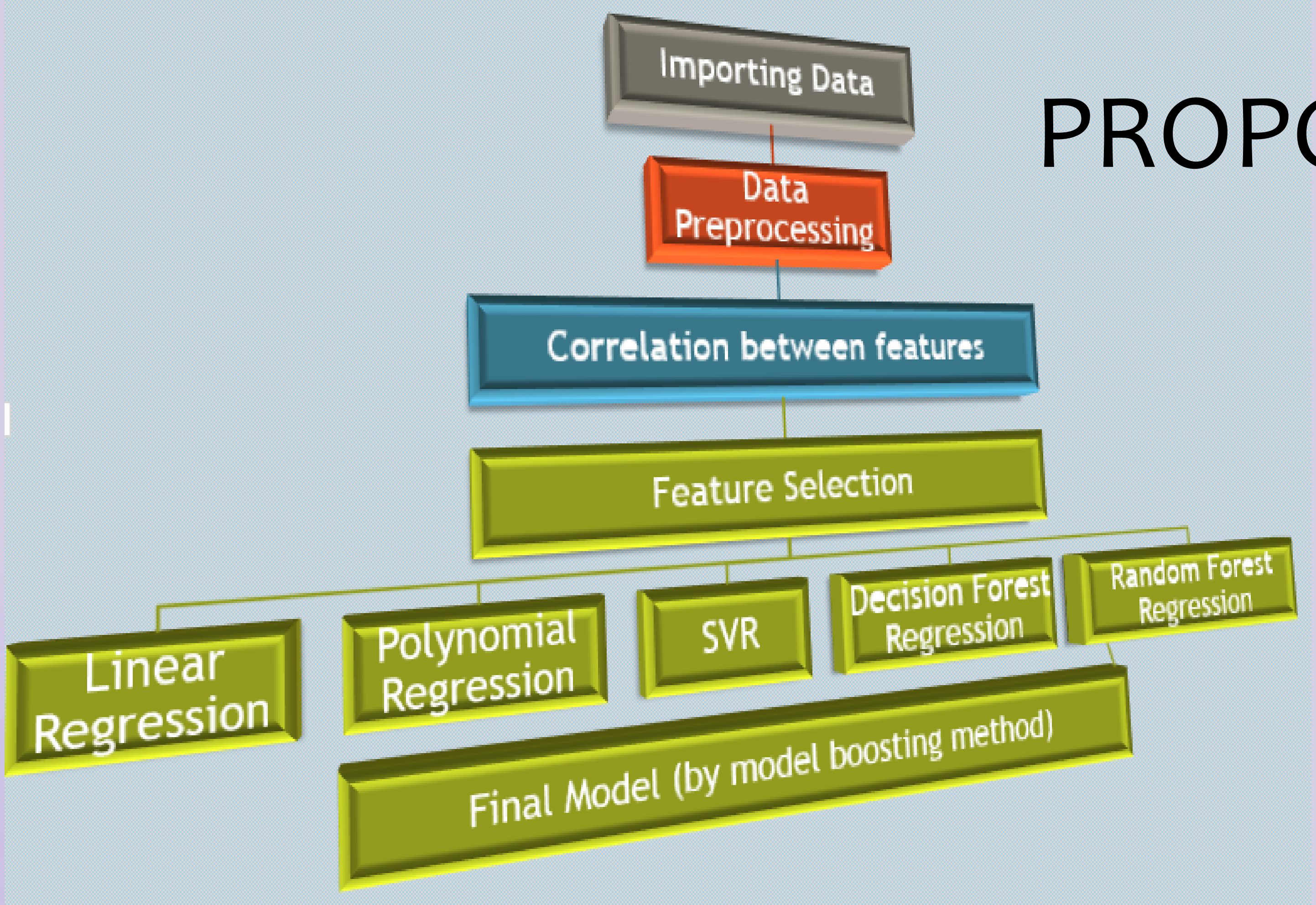
V. Random Forest Regression

Random Forest Regression is a supervised learning algorithm that uses the ensemble learning method for regression. The ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model.

Model Boosting

We have used VotingRegressor to boost our results. It is an ensemble meta-estimator that fits several base regressors, each on the whole dataset. Then it averages the individual predictions to form a final prediction. It uses linear regressor and the best possible random forest regressor to give predictions.

PROPOSED MODEL



I. Task is to predict the number of units sold of the products.

II. The impetus has been given to data preprocessing

III. Feature Selection has also been done to get the best features that would help us in our predictions.

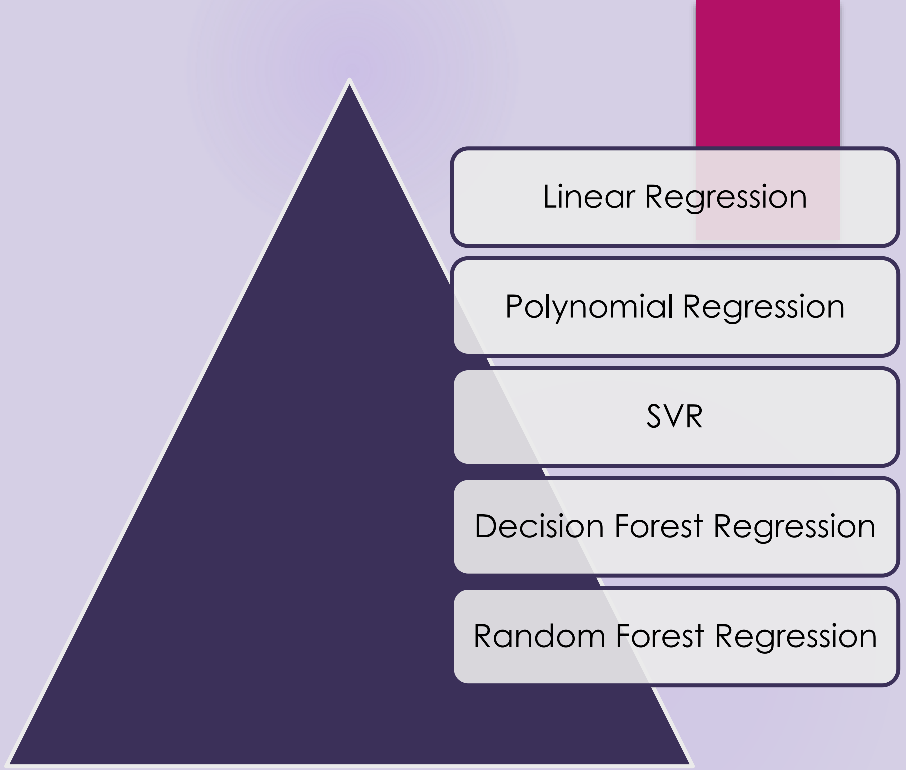
IV. Build Machine Learning Algorithms to predict the number of units sold of the products.

V. GridSearch has been performed on the best model to get more optimized parameters of our model.

VI. An attempt at model boosting has been done to get even better predictions.

MODEL COMPARISON

- ✓ Machine Learning techniques for good decision-making in the field of sales are namely Linear Regression, polynomial Regression, SVR, Decision Forest Regression, Random Forest Regression, and VotingRegressor



	Name	Train Score	Test Score	Mean Absolute Error	Mean Squared Error	Cross Validation Score (Mean Accuracy)	R2 Score
0	LinearRegression	0.815661	0.828781	1588.205844	8891266.650267	71.955947	0.828781
1	DecisionTreeRegressor	1.0	0.479628	1529.406091	27022509.482234	64.775637	0.479628
2	RandomForestRegressor	0.974564	0.824763	1311.001523	9099893.234416	74.832188	0.824763
3	LinearRegression (Poly)	0.940897	-3.176442	3882.206638	216879158.125377	71.955947	-3.176442
4	SVR	-0.117289	-0.14827	3639.465606	59628687.558016	-14.70638	-0.14827

FINAL MODEL: MODEL BOOSTING

- ✓ Using VotingRegressor to boost our results.
- ✓ A voting regressor is an ensemble meta-estimator that fits several base regressors, each on the whole dataset. Then it averages the individual predictions to form a final prediction.
- ✓ The voting regressor uses linear regressor and the best possible random forest regressor to give predictions.

	Name	Train Score	Test Score	Mean Absolute Error	Mean Squared Error	Cross Validation Score (Mean Accuracy)	R2 Score
0	LinearRegression	0.815661	0.828781	1588.205844	8891266.650267	71.955947	0.828781
1	DecisionTreeRegressor	1.0	0.479628	1529.406091	27022509.482234	64.775637	0.479628
2	RandomForestRegressor	0.974564	0.824763	1311.001523	9099893.234416	74.832188	0.824763
3	LinearRegression (Poly)	0.940897	-3.176442	3882.206638	216879158.125377	71.955947	-3.176442
4	SVR	-0.117289	-0.14827	3639.465606	59628687.558016	-14.70638	-0.14827
5	RandomForestRegressor (after GridSearchCV)	0.922982	0.769997	1385.861089	11943857.43647	76.956616	0.769997
6	VotingRegressor	0.888813	0.828968	1451.062478	8881543.623321	77.168112	0.828968

RESULTS

Specifications of the most optimum model:

Voting Regressor:

1. Linear Regressor
2. Random Forest Regressor (n_estimators=18, max_depth=4)

with results:

- Train Score: 0.88
- Test Score: 0.83
- MAE: 1451.06
- MSE: 8.88e+06
- CV Score (Mean Accuracy): 77.16
- R2 Score: 0.83

PREDICTION TESTING

- ✓ Algorithm predicted that **1161** units sold when the
  - Rating is 4.2
  - Rating five count is 66
  - Rating four count is 13
  - Rating three count is 18
  - Rating two count is 3
  - Rating one count is 7
  - Merchant rating count is 247
  - Merchant rating is 3.9433
- ✓ As per the data, the units sold are **1000** which is close.

```
y_test_df = y_test.to_frame().reset_index()
y_test_df.loc[y_test_df.index == 108]
```

	index	units_sold
108	1032	1000

	rating	rating_five_count	rating_four_count	rating_three_count	rating_two_count	rating_one_count	merchant_rating_count	merchant_rating
108	4.2	66.0	13.0	18.0	3.0	7.0	247	3.94332

```
prediction = regressor.predict([[4.2,66.0,13.0,18.0,3.0,7.0, 247.0,3.94331984]])
print(round(prediction[0]))
```

1162

CONCLUSION

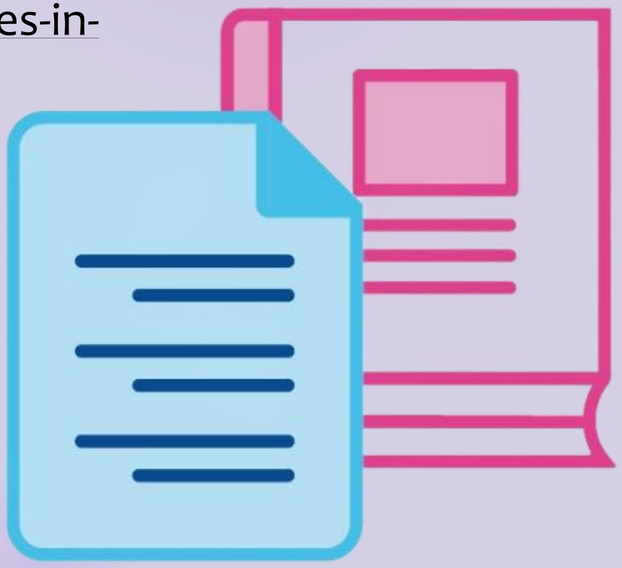
✓ This project identifies product listing strategies, primarily visual and textual presentation, that can help retailers to raise their product sales.

- Maximum sales occurred when the rating is 4 stars (good). Customers require high-quality products sold
- Black colored clothes have the most sales
- Small-sized clothes have the most sales
- Clothes made in China have the most sales
- Much higher sales if the urgency banner is not present
- Maximum sales occurred when the merchant rating is 4 stars (good)
- The merchant’s rating count is important in the seller’s choice of purchase
- Seller gets higher sales as the number of 5,4,3,2,1-star increases
- Seller gets higher sales without using advertisements as per the data

✓ Sales forecasting is an important field in the e-commerce sector and it has recently got immense popularity to boost market operations and productivity due to new technologies. To predict the sales, a Machine Learning Algorithm is built with an **accuracy of 77.16%.**

REFERENCES

- Haishan Gao, Zhaoqiang Bai, Jingqian Li. Sales Prediction Based On Product Titles and Images with Deep Learning Approaches, https://github.com/jqli0201/etsy-analysis.
- Pryzant, R., Chung, Y., Jurafsky, D. (2017). Predicting Sales from the Language of Product Descriptions eCOM@SIGIR.
- Xia, H., Pan, X., Zhou, Y., Zhang, Z. (2020). Creating the best first impression: Designing online product photos to increase sales. Decision Support Systems, Volume 131.
- Mabilama, J. M.. (2021) Sales of summer clothes in E-commerce Wish, Version 4. Retrieved November 1,2021 from https://www.kaggle.com/jmmvutu/summer-products-and-sales-in-ecommerce-wish.
- Pryzant, R., Chung, Y., Jurafsky, D. (2017). Predicting Sales from the Language of Product Descriptions. eCOM@SIGIR.
- Wei, J., Zou, K. (2019). EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. ArXiv, abs/1901.11196.



Thank you