Springboard Data Science Career Track

Capstone Project Report:

Sales Prediction
for an
Online Retail Store

Submitted By:
Koshika Agrawal

May 22, 2019

Table of Contents

1. Introduction

1.1. Problem Statement

Predicting sales is one of the most important business problems for any retail entity. If a business can predict the how much of each item it will sell in each month, it can manage its inventory better. Sales predictions also help in directing the marketing efforts in right direction to increase the chances of sale.

1.2. Objective

To forecast item-wise sales for an online retail store

1.3. Client

The client in this case is a UK-based and registered non-store online retail.The company mainly sells unique all-occasion gifts. Many customers of the company are wholesalers.

This model can be replicated to any similar online or physical store selling any kind of product.

2. Data Acquisition and Wrangling

2.1 Data Acquisition

The dataset has been taken from UCI Machine Learning Repository at:
https://archive.ics.uci.edu/ml/datasets/Online%20Retail

The dataset contains following attributes:

Attribute name	Type	Description
InvoiceNo	Nominal	A 6-digit integral number uniquely assigned to each transaction. If this code starts with letter 'C', it indicates a cancellation
StockCode	Nominal	A 5-digit integral number uniquely assigned to each distinct product

Description	Nominal	Product (item) name
Quantity	Numeric	The quantities of each product (item) per transaction
InvoiceDate	Numeric	The day and time when each transaction was generated
UnitPrice	Numeric	Product price per unit in sterling
CustomerID	Nominal	A 5-digit integral number uniquely assigned to each customer
Country	Nominal	Name of the country where each customer resides

2.2. Data Wrangling

Initial loading and inspection of datasets exposed some challenges in it for our study. We wrangled the data to make it fit for our analysis.

1. Description' and 'CustomerID' columns have null values

    - Each InvoiceNo should be linked to a single CustomerID. So we tried using InvoiceNo and CustomerId linkage to fill missing values. As we could not find the linkage, we dropped the null values

    - Deleting missing CustomerId removed all missing Description rows too.

2. StockCode does not identify a unique Description

    - Each StockCode should uniquely represent an item Description. But the original dataset has multiple Description for same StockCode. This is because there are data entry errors in the description as shown below

	StockCode	Description
48	20622	VIP PASSPORT COVER
49	20622	VIPPASSPORT COVER
101	20725	LUNCH BAG RED RETROSPOT
102	20725	LUNCH BAG RED SPOTTY
194	20914	SET/5 RED RETROSPOT LID GLASS BOWLS
195	20914	SET/5 RED SPOTTY LID GLASS BOWLS

    The data was wrangled to contain one to many mapping between StockCode and Description.

3. Some descriptions contain incidental charges like postage/shipping charges, discounts etc.
    - As these charges are not related to our analysis, we dropped these observations

4. Some of the item quantities are negative
    - These are cancelled orders

5. Some CustomerID linked with 2 countries

    As per the data attribute description: 'Country' column is the name of the country where each customer resides. But we don't have any information on how is this data being captured. Is it through IP address of the country while creating account, or may be based on the shipping address, or may be something else.

    Logically, each CustomerID should be linked to one country only. The reason for having more than one country could be:

    a. Data entry error
    b. Customer has moved to another country, and has got the address changed in his account
    c. In case this attribute reflects the shipping address, the customer has shipped the order to an address different from his own.
    d. In case this attribute is captured through the IP address while ordering, the customer might be ordering while travelling to another country.

    Further analysis of data does not make it clear what is the reason behind 2 countries for a CustomerID, so for now, we are not making any changes in the CustomerID and country linkage.

6. Different unit price of same item for different transactions

    Unit price of an item keeps changing for different transactions. This poses problem while aggregating the data. We added another column for total price (which is quantity multiplied by unitPrice). While aggregating we added the quantity and total price, and find the unit price from the aggregated values.

2.3. Derived Features

We derived following features from the already existing ones to aid in our analysis.

1. CancelledOrder containing boolean values, 1 if order was cancelled, 0 otherwise
2. InternationalOrders containing boolean values, 0 if order came from UK, 1 if the order came from outside UK

3. TotalPrice containing float values = unitPrice * Quantity

2.4. Data Wrangling Summary

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

Shape: (541909,8)

	InvoiceNo	StockCode	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	Description	CancelledOrder	InternationalOrders	TotalPrice
0	536365	85123A	6	2010-12-01 08:26:00	2.55	C17850	United Kingdom	CREAM HANGING HEART T-LIGHT HOLDER	0	0	15.30
1	536365	71053	6	2010-12-01 08:26:00	3.39	C17850	United Kingdom	WHITE METAL LANTERN	0	0	20.34
2	536365	84406B	8	2010-12-01 08:26:00	2.75	C17850	United Kingdom	CREAM CUPID HEARTS COAT HANGER	0	0	22.00
3	536365	84029G	6	2010-12-01 08:26:00	3.39	C17850	United Kingdom	KNITTED UNION FLAG HOT WATER BOTTLE	0	0	20.34
4	536365	84029E	6	2010-12-01 08:26:00	3.39	C17850	United Kingdom	RED WOOLLY HOTTIE WHITE HEART.	0	0	20.34

Shape: (404618,11)

The wrangled dataset is saved to a csv file.

The code for data acquisition and wrangling can be accessed in this ipython notebook.

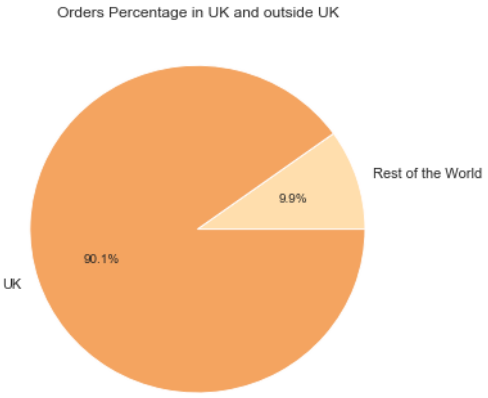
3. Data Exploration

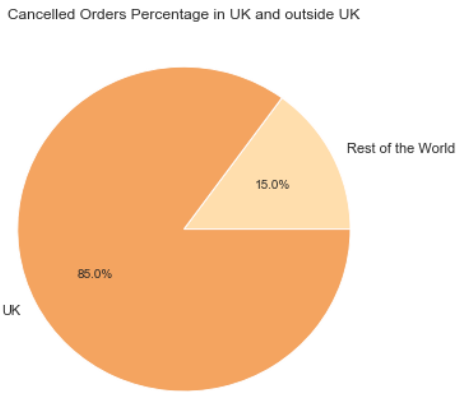
The code for EDA can be found in this ipython notebook.

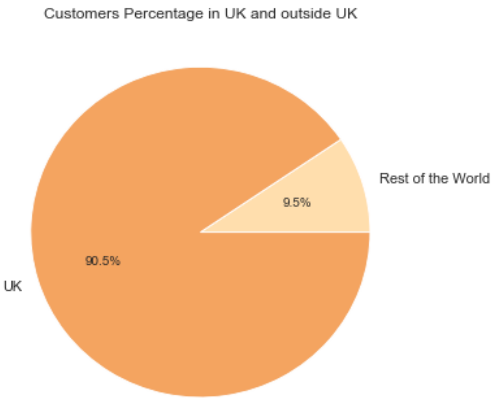
3.1. Worldwide distributions

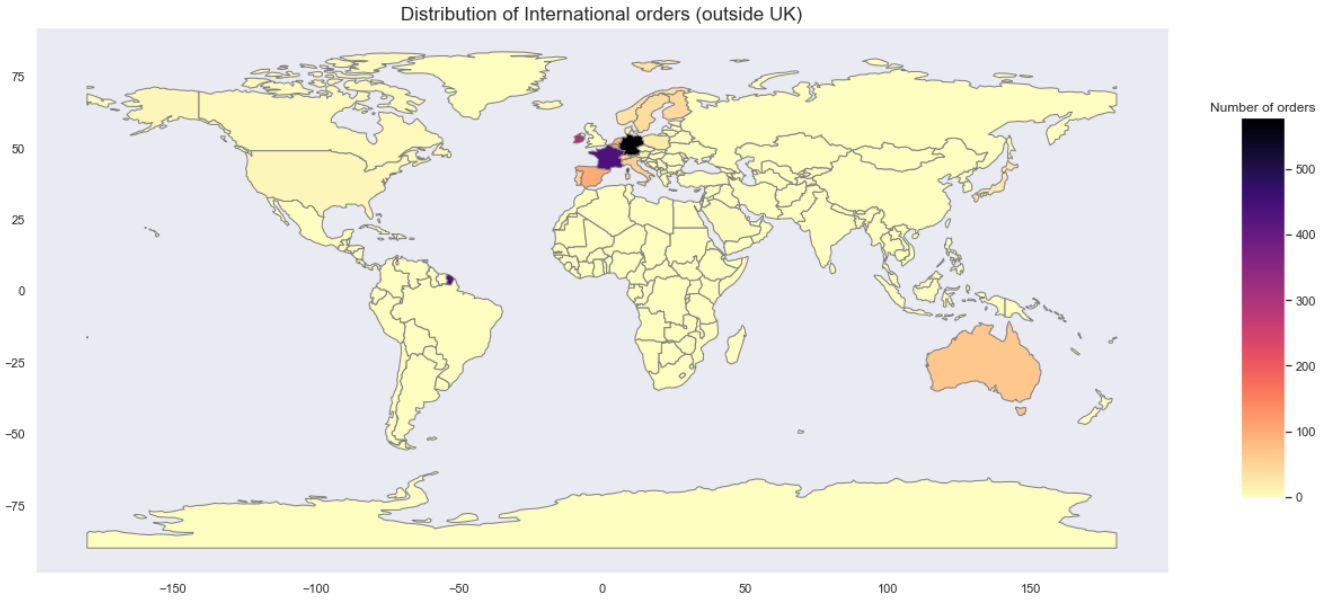
The retailer has its customers all over the world. We will take a look at following distributions on a world map:

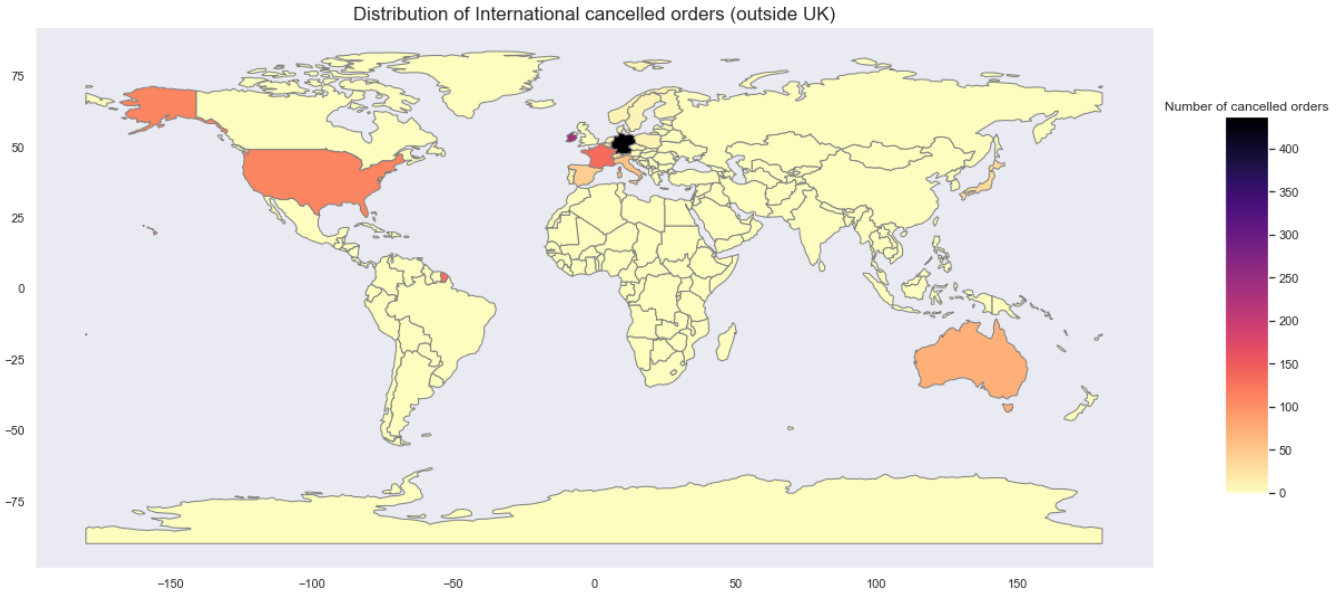
a. Orders
b. Customers
c. Cancelled orders

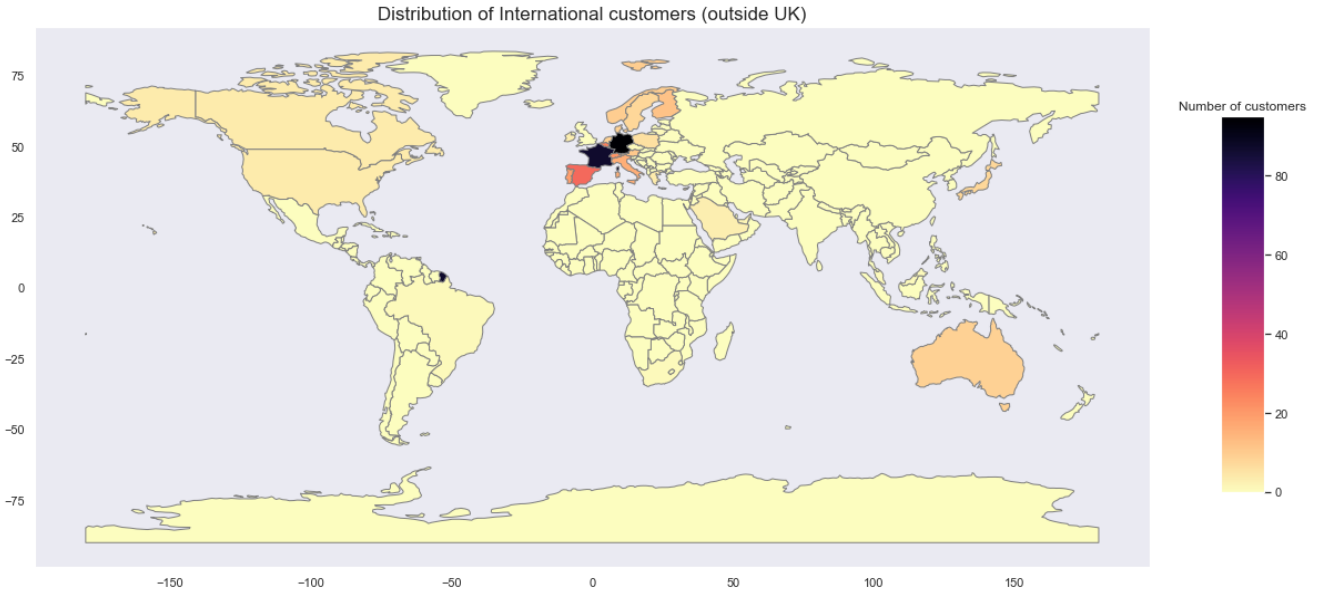

Orders Percentage in UK and outside UK
Rest of the World
9.9%
90.1%
UK


Cancelled Orders Percentage in UK and outside UK
Rest of the World
15.0%
85.0%
UK


Customers Percentage in UK and outside UK
Rest of the World
9.5%
90.5%
UK


Distribution of International orders (outside UK)
75
50
25
0
−25
−50
−75
−150
−100
−50
0
50
100
150
Number of orders
500
400
300
200
100
0

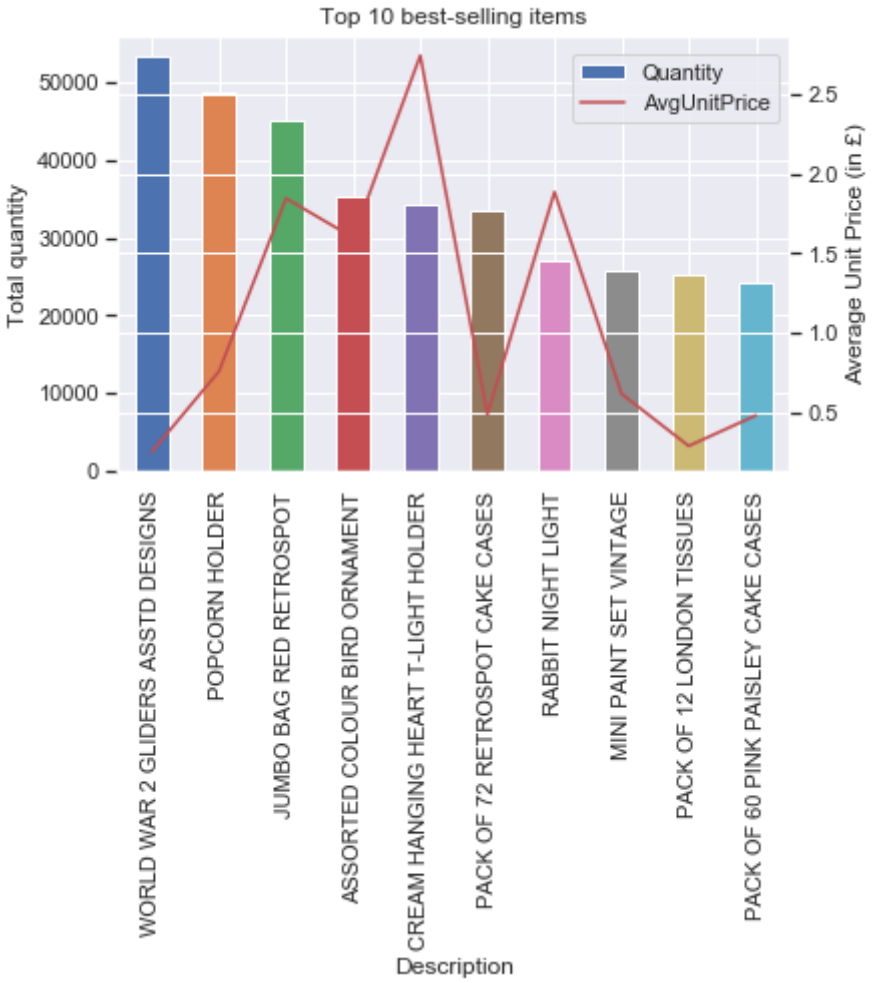


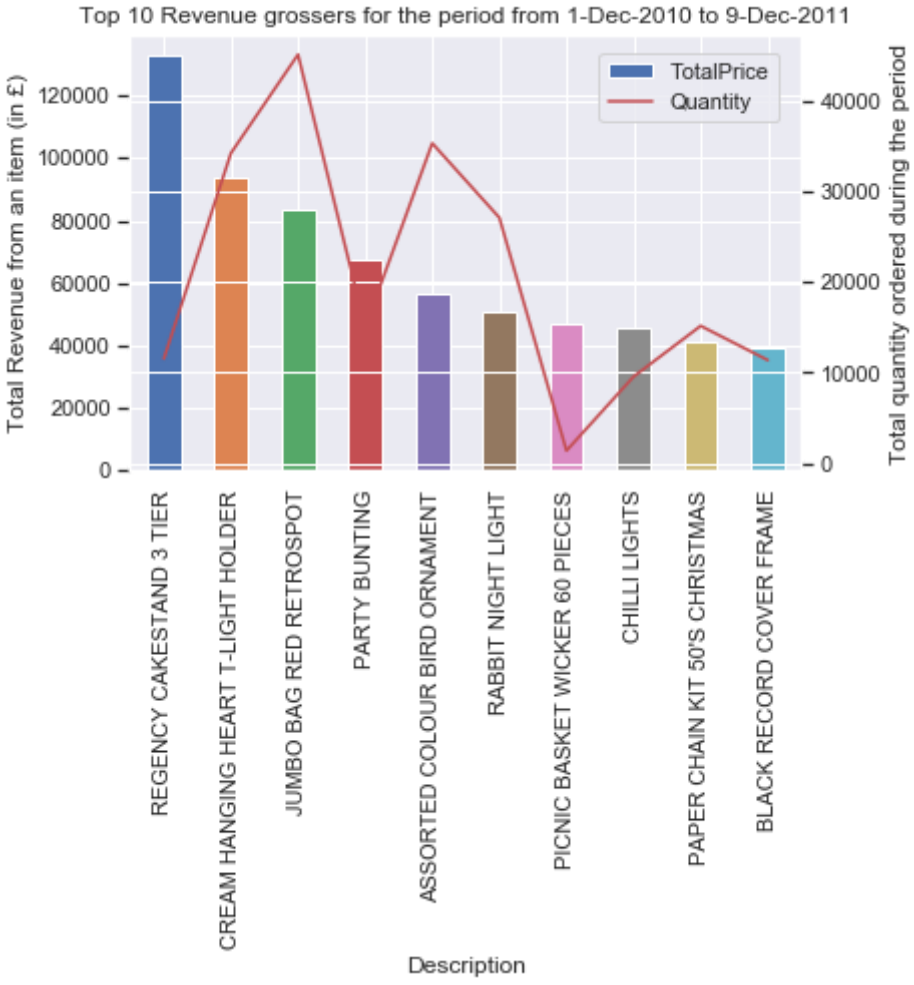


Observations

1. 90% sales comes from UK and 90% customers also are from UK
2. Outside UK, most of the sales is from Europe
3. In international sales, Germany, France and Ireland are among the highest
4. Outside UK, most customers are from Germany, France and Spain
5. Outside UK, Germany, Ireland, France and US show highest number of cancelled orders
6. Outside Europe, highest sales comes from Australia, while the highest number of cancellations come from United States.

3.2. Best Sellers





Observations

1. The best selling products vary in their average unit price. So it doesn't seem to have any relation with its price.
2. The No. 1 best selling product sells almost double the quantity of the 10th best seller.
3. The no. 1 revenue grosser leads its immediate follower by 26%.

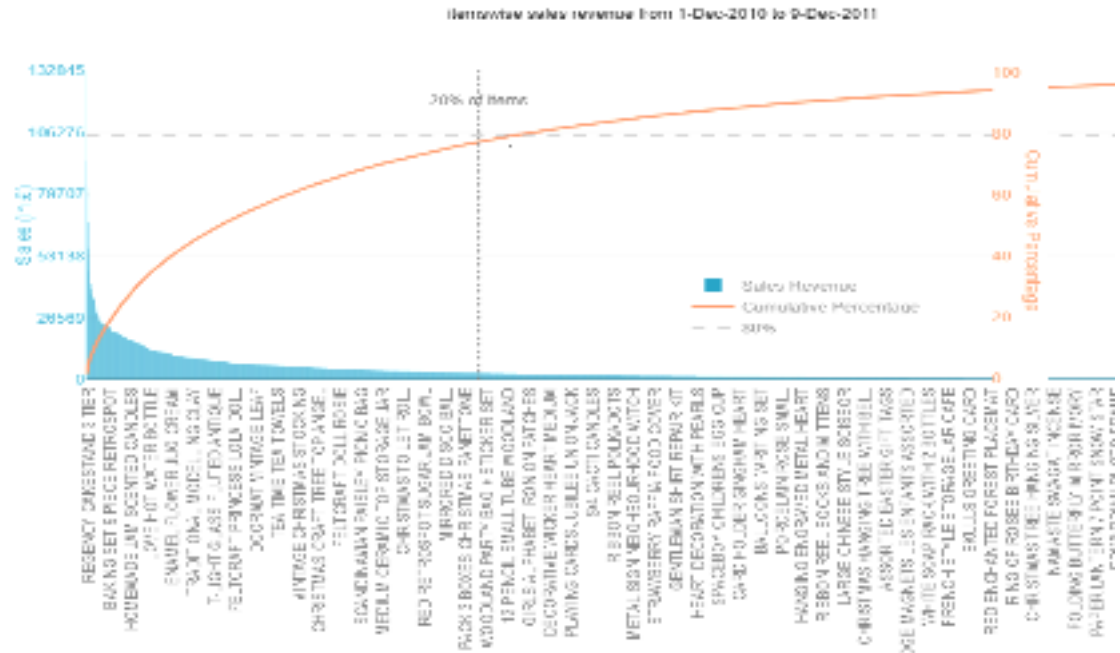
3.3 Pareto Principle (80-20 rule)

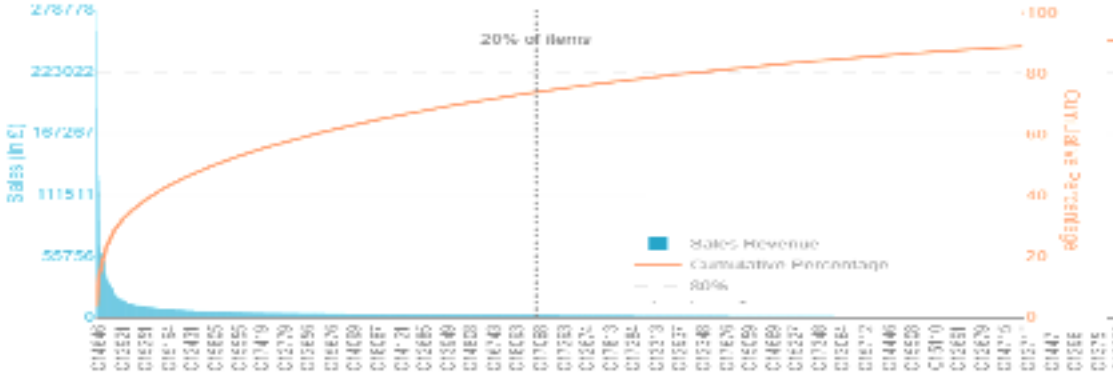
The Pareto principle (also known as the 80/20 rule) states that, for many events, roughly 80% of the effects come from 20% of the causes.(Source: wikipedia)

For a sales entity, Pareto principle could suggest that 80% sales of a company comes from 20% of its products and/or 80% of its sales comes from 20% of its customers.

Our online retail store has 3652 unique items for sale, and 4357 unique customers for the period from Dec 1 2010 to Dec 9 2011. Here, Pareto principle is of value because instead of focussing on such a huge number of items and customers, the company can just focus on 20% of these in order to effect 80% of its sales.

We can see the applicability of Pareto principle to our dataset.

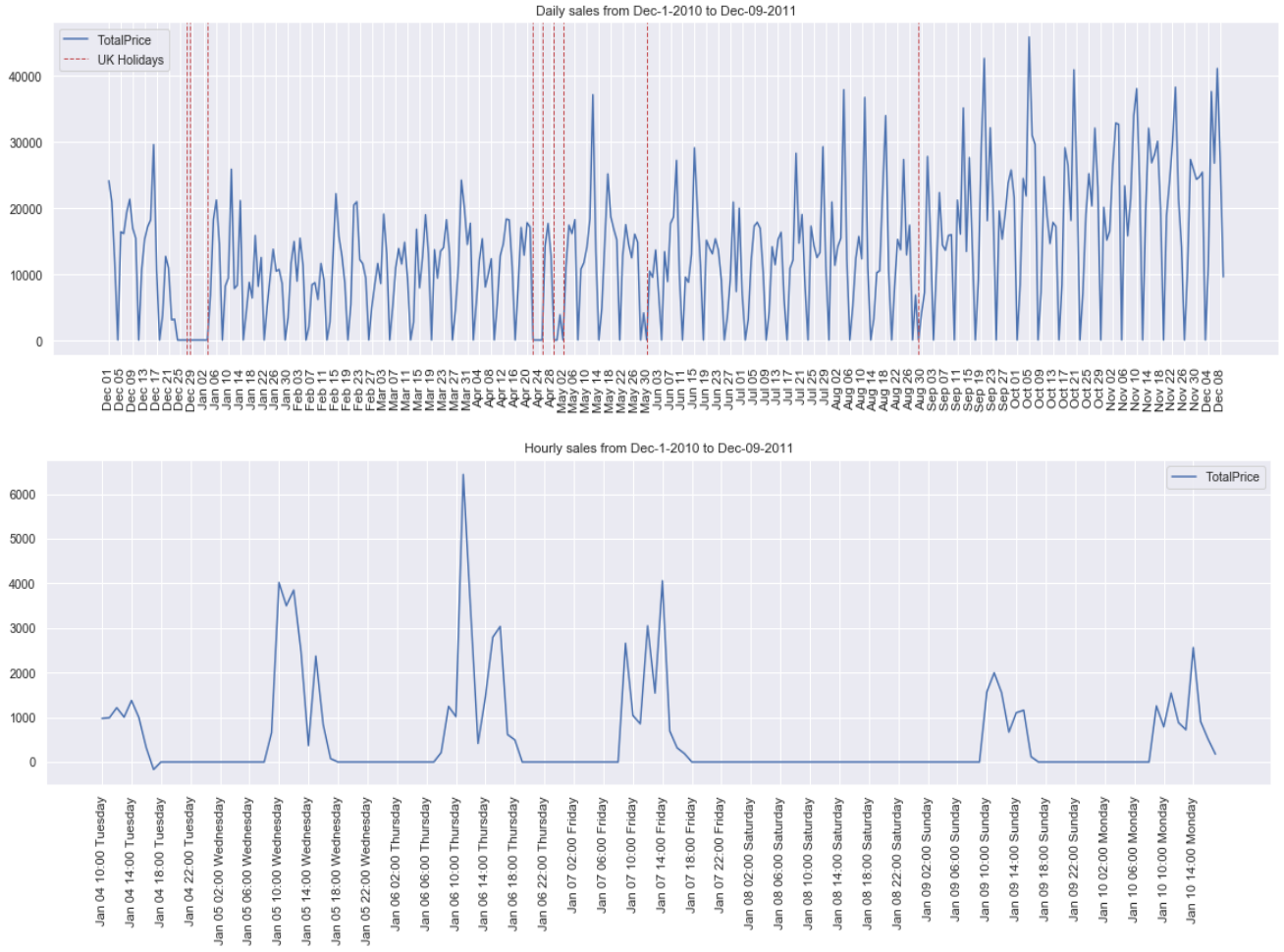




Observations

1. Pareto principle holds true for items in our dataset, as 22% of all the items are contributing to 80% of the sales revenue. 22% of all items means 803 items.
2. 27% of all the customers are contributing to 80% of the sales revenue. 27% of all items means 1176 items

3.4 Time series plots



Observations

1. There is no sales happening on Saturdays.
2. Sales falls down during the holidays
3. There is zero sales happening on Boxing day (Dec 26th) which probably means that the company does not offer any promotions during the holidays/special days.
4. Sales happens only during the working hours - between 8am and 6pm.
5. There is no particular trend seen in monthly and weekly sales. The overall sales has increased through the months.

4. Descriptive and Inferential Statistics

We dropped following columns:

1. InvoiceNo and CustomerID: As we have to predict itemwise sales, we don't need these columns
2. Country, InternationalOrders: We are not making distinction between international or domestic sales in the predictions, so we don't need this column
3. Description: We have the StockCode, so this is repetitive
4. CancelledOrder: Cancelled orders have negative quantity and price. So while aggregating the dataset, cancellations will be taken into account. We don't need this column.
5. UnitPrice: In our dataset, same item has different prices. So we will keep TotalPrice and drop UnitPrice. While aggregating we will calculate the UnitPrice from TotalPrice and Quantity for each item and month again.

We added year, quarter, month, week, weekday, day, unitPrice derived from the InvoiceDate:

4.1 Summary Statistics

	Quantity	TotalPrice	Year	Quarter	Month	Week	Weekday	Day	Dayofyear	UnitPrice
count	224628.000000	224628.000000	224628.000000	224628.000000	224628.000000	224628.000000	224628.000000	224628.000000	224628.000000	2.240890e+05
mean	21.755712	36.874346	2010.936713	2.739730	7.253993	29.413662	2.617821	15.114460	204.576740	NaN
std	65.986694	136.945165	0.243479	1.130302	3.448225	14.901138	1.931312	8.647994	104.412414	NaN
min	-8974.000000	-3825.360000	2010.000000	1.000000	1.000000	1.000000	0.000000	1.000000	4.000000	-inf
25%	3.000000	6.250000	2011.000000	2.000000	4.000000	16.000000	1.000000	7.000000	111.000000	8.500000e-01
50%	8.000000	15.300000	2011.000000	3.000000	8.000000	32.000000	2.000000	15.000000	220.000000	1.663433e+00
75%	24.000000	33.000000	2011.000000	4.000000	10.000000	43.000000	4.000000	22.000000	299.000000	3.750000e+00
max	4848.000000	39619.500000	2011.000000	4.000000	12.000000	51.000000	6.000000	31.000000	357.000000	inf

Observations:

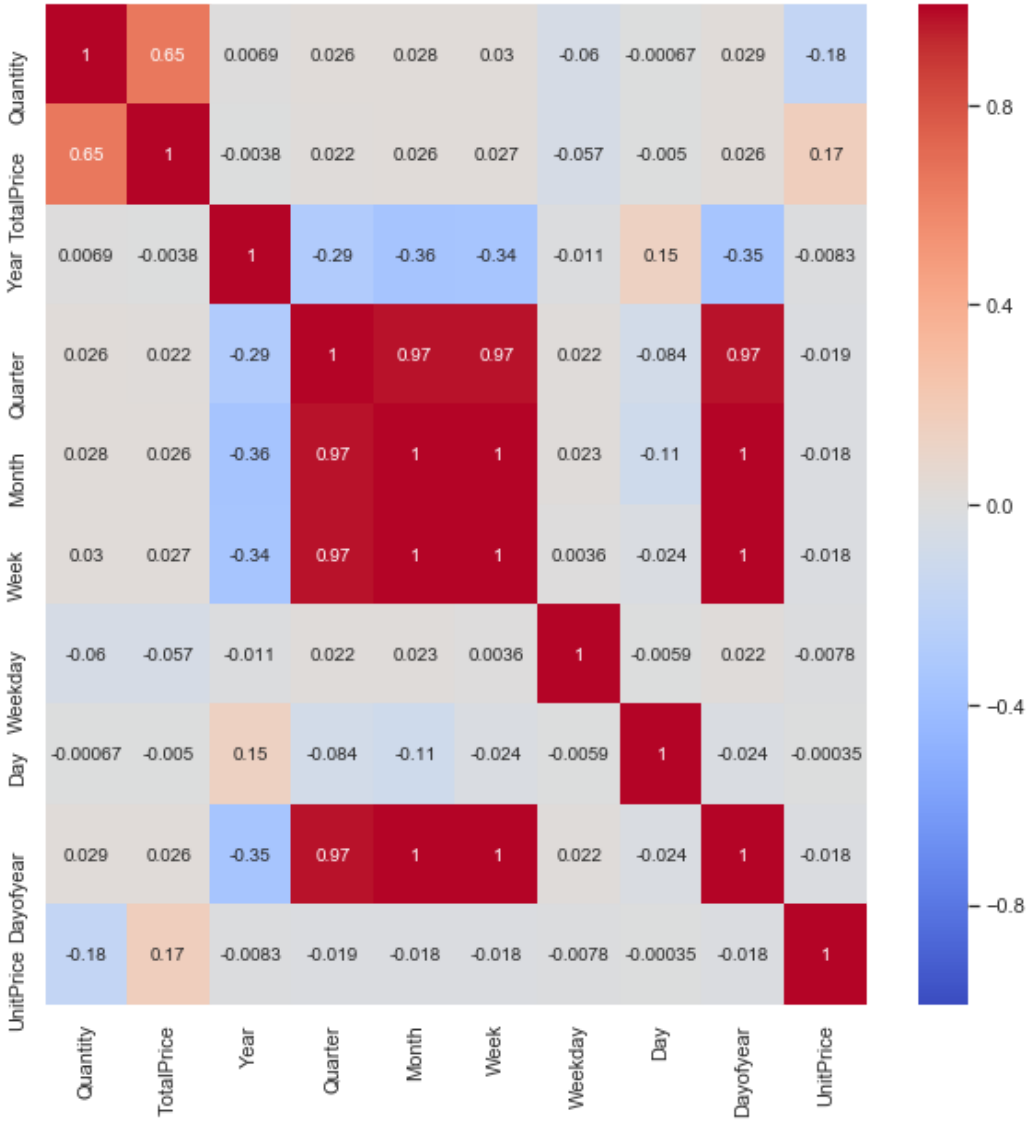
- There are some negative values for Quantity and TotalPrice. We drop these observations.
- The maximum value for Quantity and TotalPrice is very high, so we should do outlier analysis to remove the outliers.

4.2 Outlier Analysis

We used z-score to identify outliers from Quantity and TotalPrice and dropped the observations which are more than 3 standard deviations away. This is how the summary statistics looks like after removing outliers.

	Quantity	TotalPrice	Year	Quarter	Month	Week	Weekday	Day
count	216753.000000	216753.000000	216753.000000	216753.000000	216753.000000	216753.000000	216753.000000	216753.000000
mean	17.320586	29.372231	2010.937057	2.738772	7.251461	29.402726	2.630510	15.127034
std	25.659530	42.252063	0.242860	1.130421	3.447829	14.900267	1.942232	8.651776
min	1.000000	0.060000	2010.000000	1.000000	1.000000	1.000000	0.000000	1.000000
25%	3.000000	6.640000	2011.000000	2.000000	4.000000	16.000000	1.000000	7.000000
50%	8.000000	15.300000	2011.000000	3.000000	8.000000	32.000000	2.000000	15.000000
75%	23.000000	31.600000	2011.000000	4.000000	10.000000	43.000000	4.000000	22.000000
max	211.000000	343.050000	2011.000000	4.000000	12.000000	51.000000	6.000000	31.000000

4.3 Correlation between features - Heatmaps



Above heatmap shows strong correlation between Quarter, Month, Dayofyear and Week. We will drop , Dayofyear, quarter and month, and just keep week, as week is most strongly correlated with week. Year has strong correlation with Day, and as it is not strongly correlated with Quantity, we will drop it.

The code for descriptive and inferential statistics can be viewed in this ipython notebook.

5. Machine Learning

The code for the machine learning procedure can be viewed in this ipython notebook.

5.1 Target and Feature Variables

We are building a machine learning algorithm to predict sale quantity of each item for a month. So, the target variable is Quantity. As it is a continuous variable, we will be using regression algorithms.

5.2 Train Test Split

Using the entire dataset to train our model might lead to data leakage and thus affect the performance of the trained model on unseen data. To address this issue, we split our dataset into train and test datasets wherein we train our model on the train dataset, and test its performance on the test dataset.

For our project, we will hold out the data for last month from Nov-01-2011 to Dec-09-2011 as our test set, and the remaining data will be used to train our model.

After the split, we have 180,661 observations in the train set, and 36,092 observations in the test set, which is a ratio of 83:17

5.3 Process and Methodology

To find the best model for our purpose, we train our data on different algorithms, compare them and then select the one that gives the best performance on our evaluation criteria.

We will use the following algorithms:

1. Linear Regression
2. Regularization Model - Ridge
3. Regularization Model - Lasso
4. Ensemble Model - Random Forest
5. Ensemble Model - Gradient Boost

5.4 Performance Evaluation

To compare the performance of different algorithms, we are using RMSE (Root Mean Square Error) of the prediction and time taken to fit/predict the model, and select the best. The model with lowest RMSE and time taken is desirable.

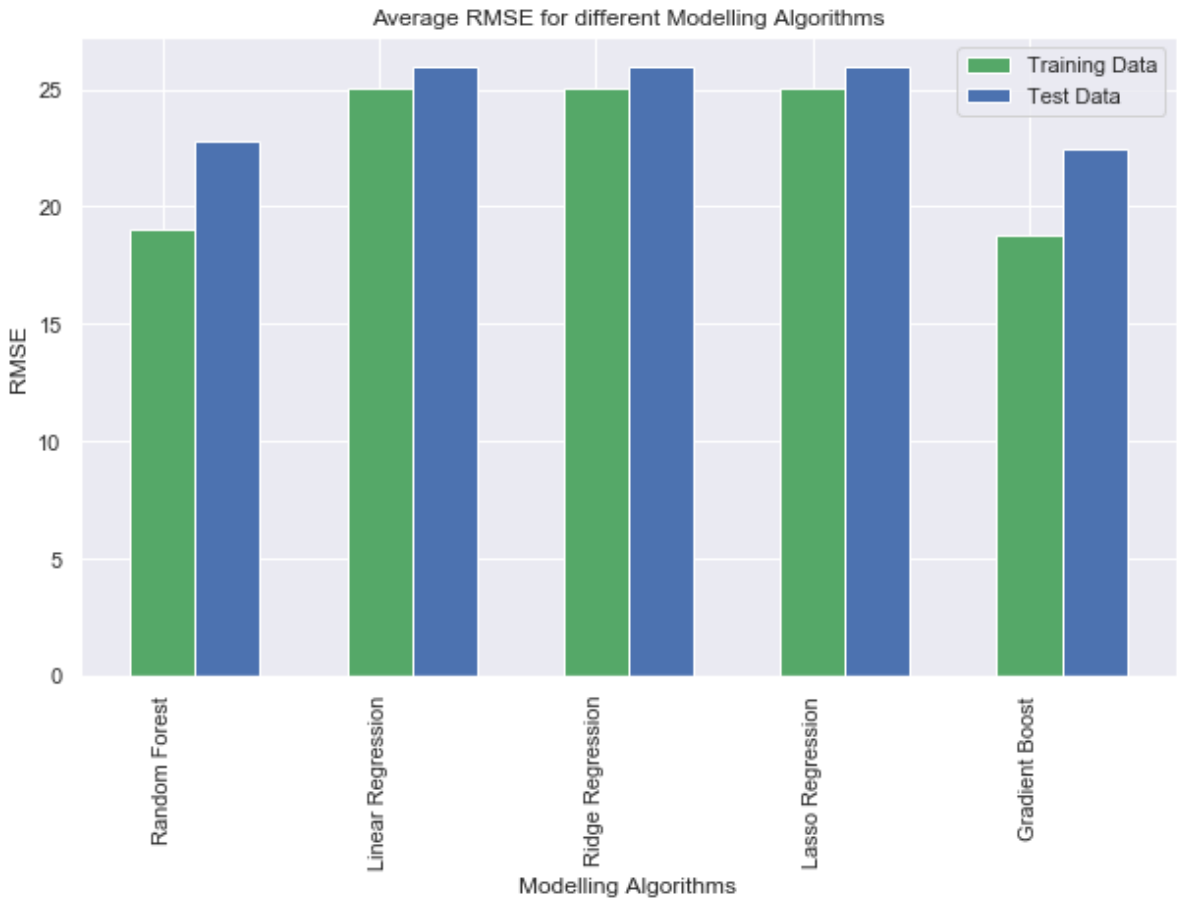
5.5 Model Comparison

We compare the RMSE for train and test datasets for different algorithms below.

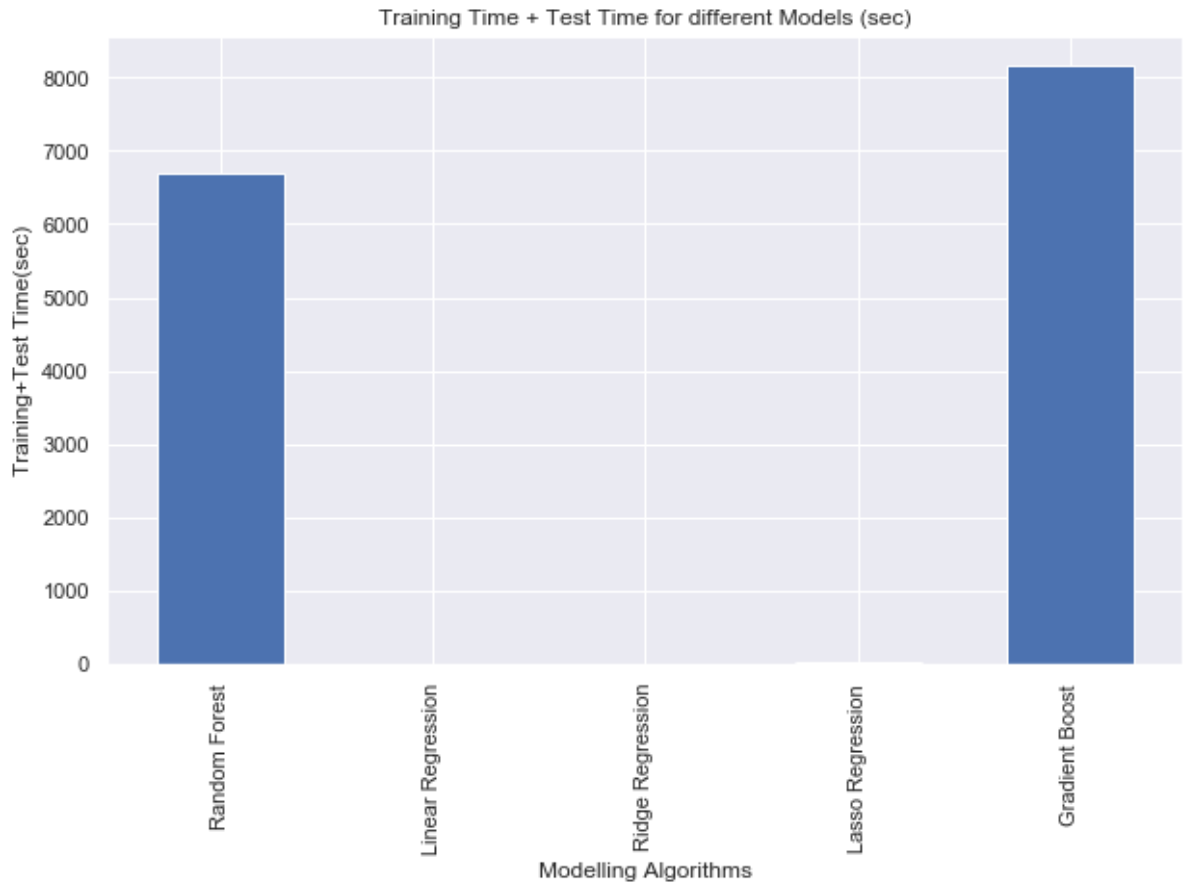
Modelling Algo	Train RMSE	Test RMSE	Hyperparameters	Training+Test Time(sec)
Random Forest	19.004094	22.758548	{'n_jobs': -1, 'n_estimators': 500, 'min_sampl...	6706
Linear Regression	25.033909	25.931982		0.516
Ridge Regression	25.033909	25.931989	{'alpha': 145}	6.917
Lasso Regression	25.034230	25.933011	{'alpha': 0.22}	31.328
Gradient Boost	18.744075	22.420339	{'n_estimators': 1000, 'min_samples_split': 5,...	8155

6. Summary and Conclusion

Below is the graphical representation of training and test RMSE for all algorithm used. The best performing algorithms are also overfitted, but that does not concern us much, as despite overfitting, we are getting low RMSE for test dataset.



Random Forest and Gradient Boost show the best performance on the test dataset. In order to decide one of these, we look at the time taken in fitting the model. In all the algorithms, the prediction time was very small in comparison to the fit time.



As seen above, Random Forest is taking much lesser time as compared to Gradient Boost. So we choose Random Forest as our final algorithm. Random Forest gives an RMSE of 22.7 on our test data.

7. Result

The objective of this project was to predict sales for each item in a month. The model we came up with gives us decent results with RMSE of 23 on test data.

8. Scope of further study

1. Predict customer churn and suggest ways to prevent the churn
2. Customer segmentation and buying behavior analysis