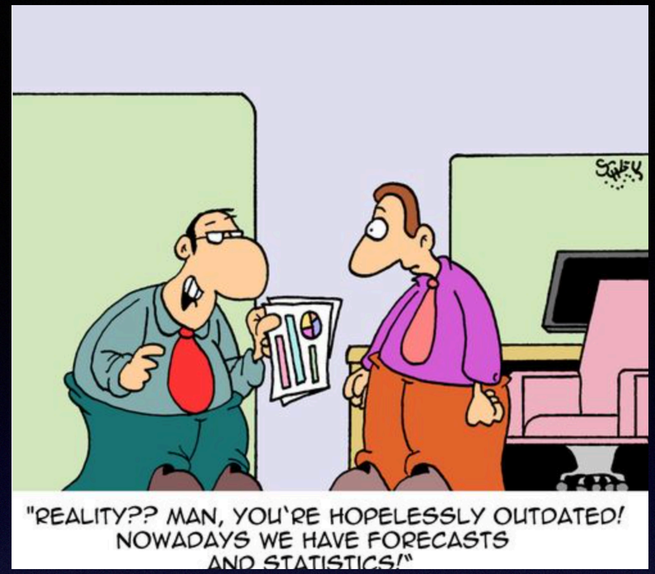


Capstone_Presentation

Demand Forecast Project

Presented by - Swati Sharma

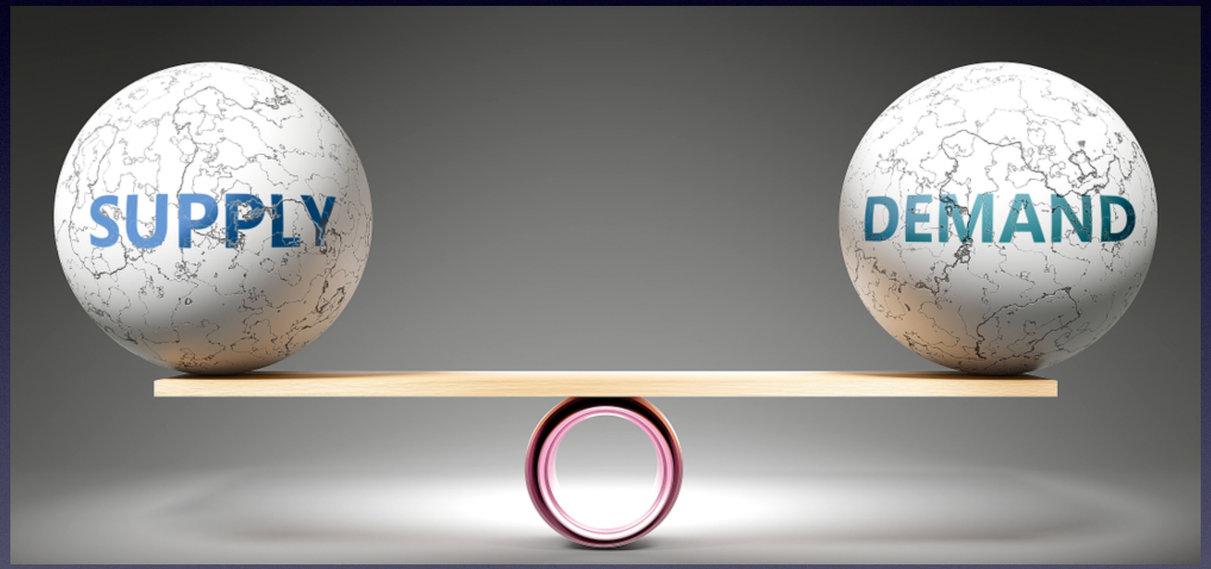
Context

Fresh foods, a challenge to any grocer or retailer

- Short shelf life,
- Appearance,
- Season dependency,
- Specific transportation,
- Storage requirements

Led to Demand and Supply Imbalance.

Food wastage, - 2.5 billion tons every year



Accurate Demand forecasting helps optimize inventory, minimize costs, and ensure better customer satisfaction.

What is the Problem?

What opportunities exist for grocers or retailers to effectively develop and implement a new “Forecasting ML model” to better forecast the demand for perishable goods.

Data Collection

- **Gather historical data:** Collected raw data as a csv file with columns such as record_ID, week, store_id, sku_id, total_price, base_price, is_featured_sku, is_display_sku and units_sold.

- **Data sources:** Kaggle, an external data source.

- **Data granularity:** The idea was to divide the data on the basis of day, month and year.

Step 1:

	record_ID	week	store_id	sku_id	total_price	base_price	is_featured_sku	is_display_sku	units_sold
0	1	17/01/11	8091	216418	99.0375	111.8625	0	0	20
1	2	17/01/11	8091	216419	99.0375	99.0375	0	0	28
2	3	17/01/11	8091	216425	133.9500	133.9500	0	0	19
3	4	17/01/11	8091	216233	133.9500	133.9500	0	0	44
4	5	17/01/11	8091	217390	141.0750	141.0750	0	0	52
...	...	...	...	...	...	...	...	...	...
150145	212638	09/07/13	9984	223245	235.8375	235.8375	0	0	38
150146	212639	09/07/13	9984	223153	235.8375	235.8375	0	0	30
150147	212642	09/07/13	9984	245338	357.6750	483.7875	1	1	31
150148	212643	09/07/13	9984	547934	141.7875	191.6625	0	1	12
150149	212644	09/07/13	9984	679023	234.4125	234.4125	0	0	15

150150 rows × 9 columns

Independent vs Dependent Variables

Independent Variables

Dependent Variable

Features/ Independent Variables

Variables that is changed

	record_ID	week	store_id	sku_id	total_price	base_price	is_featured_sku	is_display_sku	units_sold
0	1	17/01/11	8091	216418	99.0375	111.8625	0	0	20
1	2	17/01/11	8091	216419	99.0375	99.0375	0	0	28
2	3	17/01/11	8091	216425	133.9500	133.9500	0	0	19
3	4	17/01/11	8091	216233	133.9500	133.9500	0	0	44
4	5	17/01/11	8091	217390	141.0750	141.0750	0	0	52
...	...	...	...	...	...	...	...	...	...
150145	212638	09/07/13	9984	223245	235.8375	235.8375	0	0	38
150146	212639	09/07/13	9984	223153	235.8375	235.8375	0	0	30
150147	212642	09/07/13	9984	245338	357.6750	483.7875	1	1	31
150148	212643	09/07/13	9984	547934	141.7875	191.6625	0	1	12
150149	212644	09/07/13	9984	679023	234.4125	234.4125	0	0	15

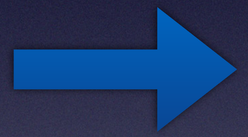
150150 rows × 9 columns

Target Variables/ Label/ Dependent Variables

Variables affected by the change

Data Wrangling / Data Cleaning

Week
2011-01-17
2011-01-17
2013-09-07

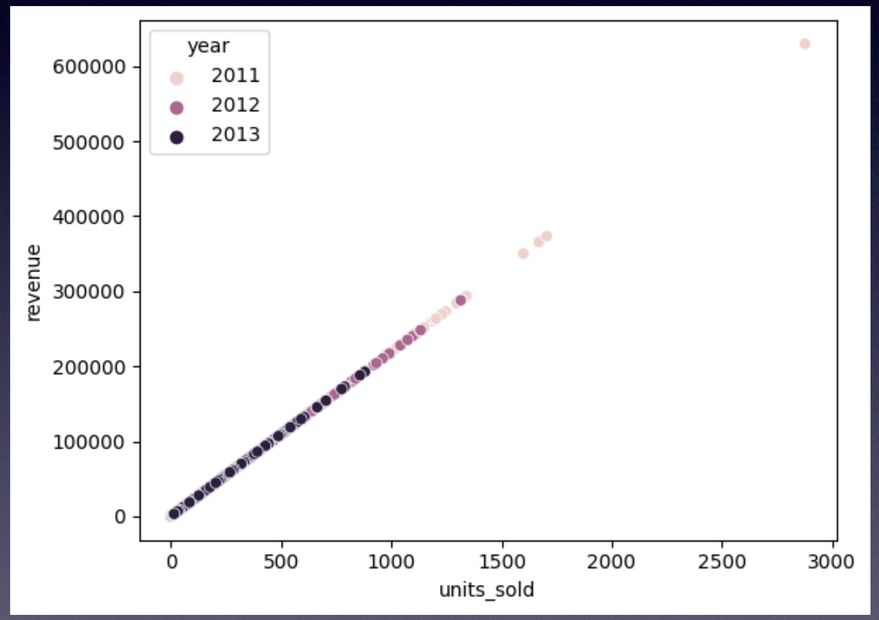


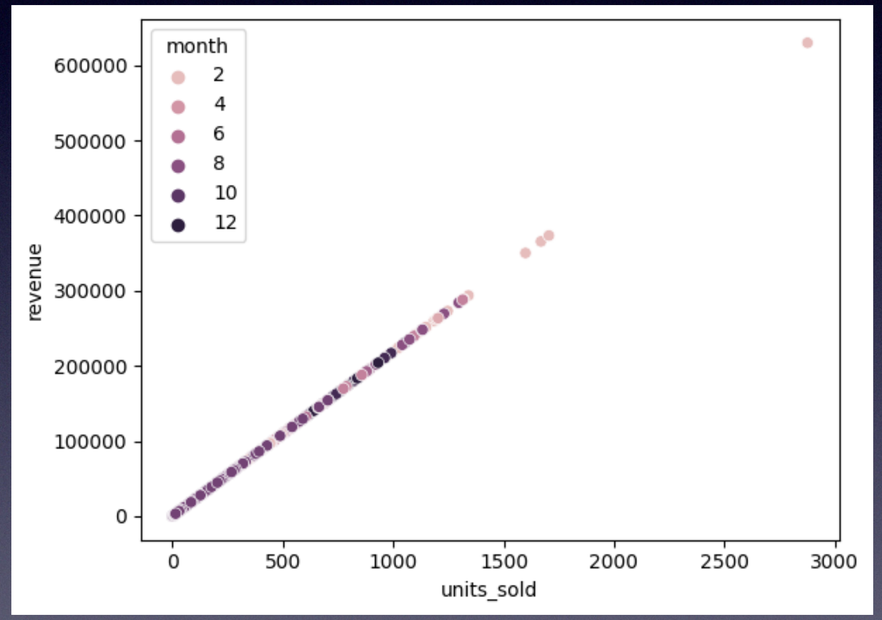
Year	Month	Day
2011	1	17
2011	1	17
2013	9	7

Converted the **week** column to datetime column
Extract the corresponding **day**, **month**, and **year** for each week
Drop the original week column

Identify Trends

Created a column “revenue” to understand the Target variable distribution - Multiply units sold by average price

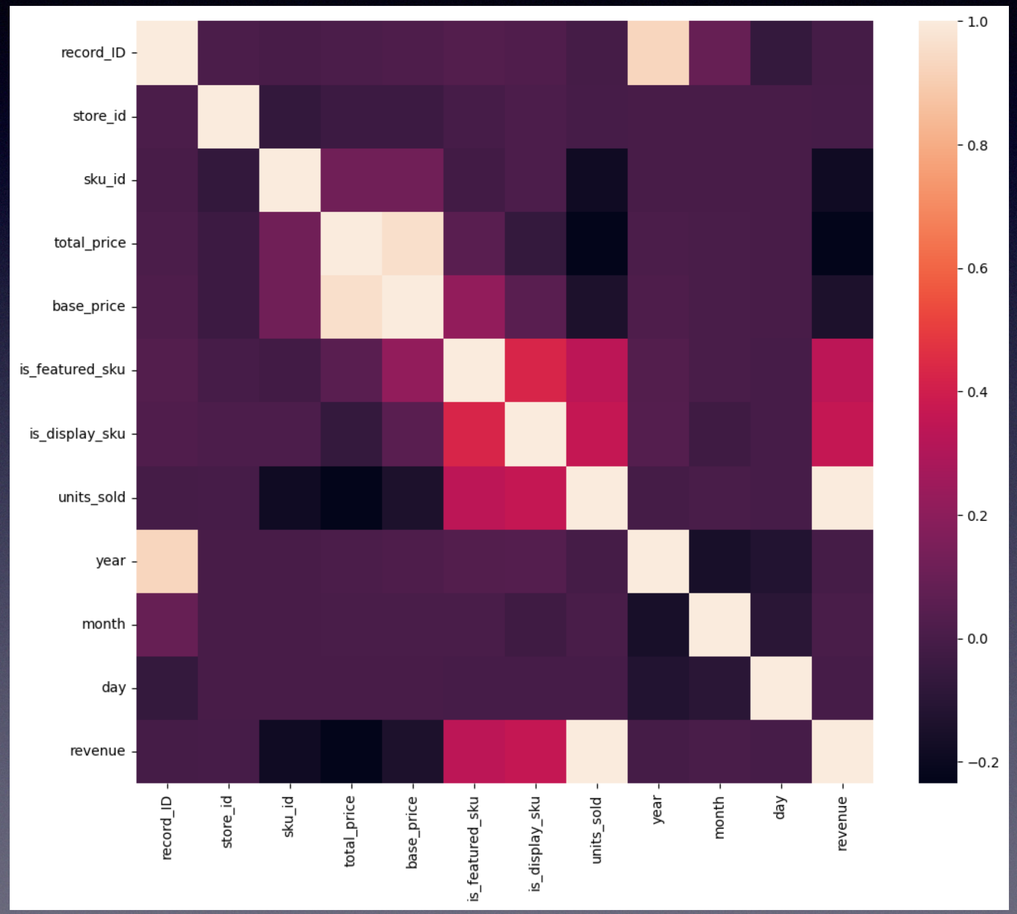




Summary: Decline in the sales per units sold between 2011 and 2013
February (Month 2) saw the maximum purchases
August (Month 8) saw the least amount of sales

Exploratory Data Analysis

Feature correlation heatmap - Gain a high level view of relationships amongst the features



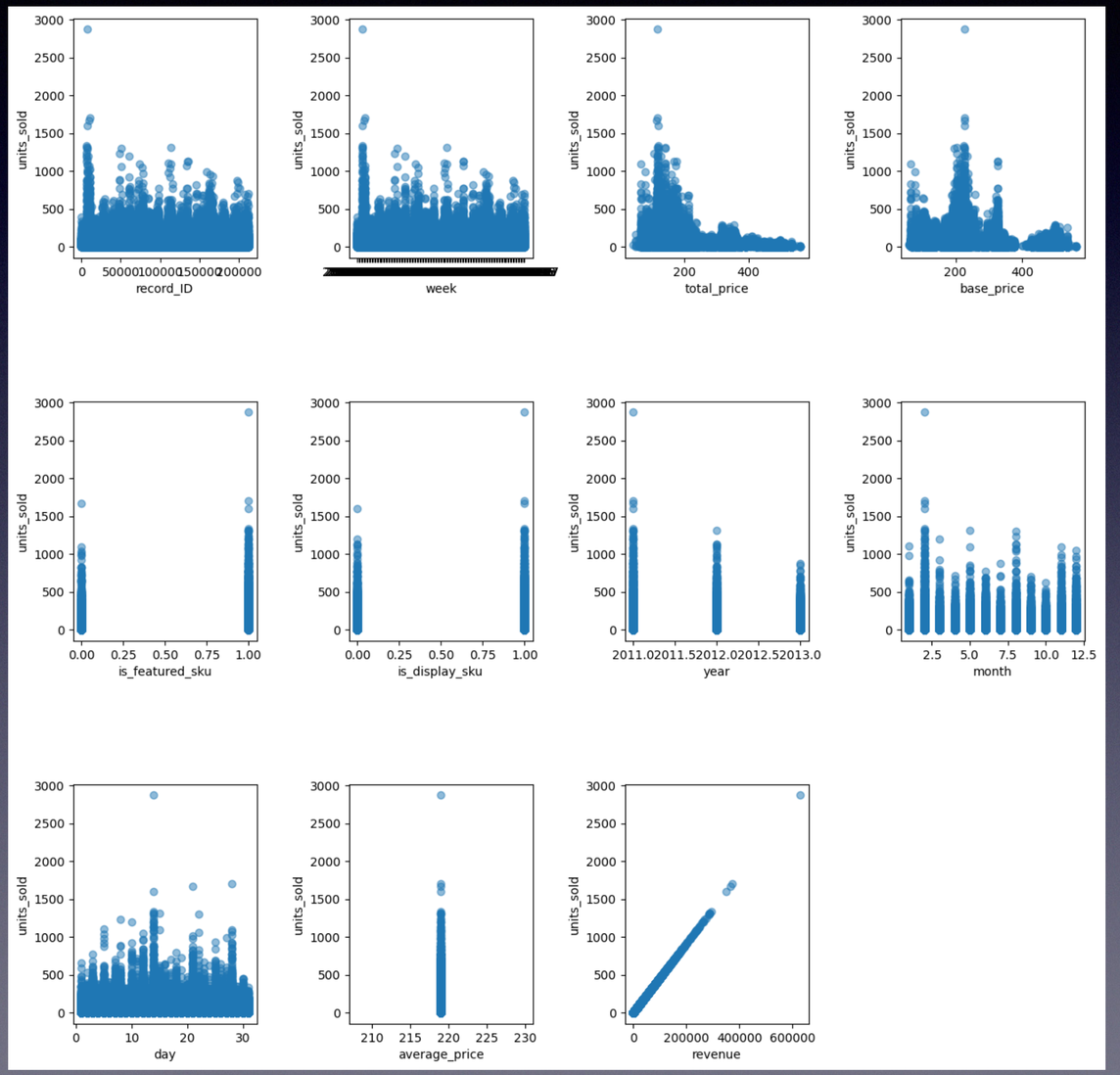
- There's a strong positive correlation between units_sold and revenue.
- record_id and year are positively correlated.
- Strong positive correlation between sku_id and total_price and base_price.

Distribution of units_sold per Store ID
Store ID's 9845, 9823 and 8869 dominates per units_sold.

Exploratory Data Analysis - Continue

Create Scatter plots for visualizing the relationship between a numeric feature against target variable, units sold.

- Year, Month and Day seems useful to understand the seasonality and behavioral aspect.
- is_featured_sku and is_display_sku appear quite similar.
- There are some outliers present in almost all columns.



Pre-processing and Training data

Drop Insignificant Columns
Week
is_featured_sku
is_display_sku
Average_price

Categorical Columns (One-hot encoding)
month
store_id
sku_id
year

Split Dataset

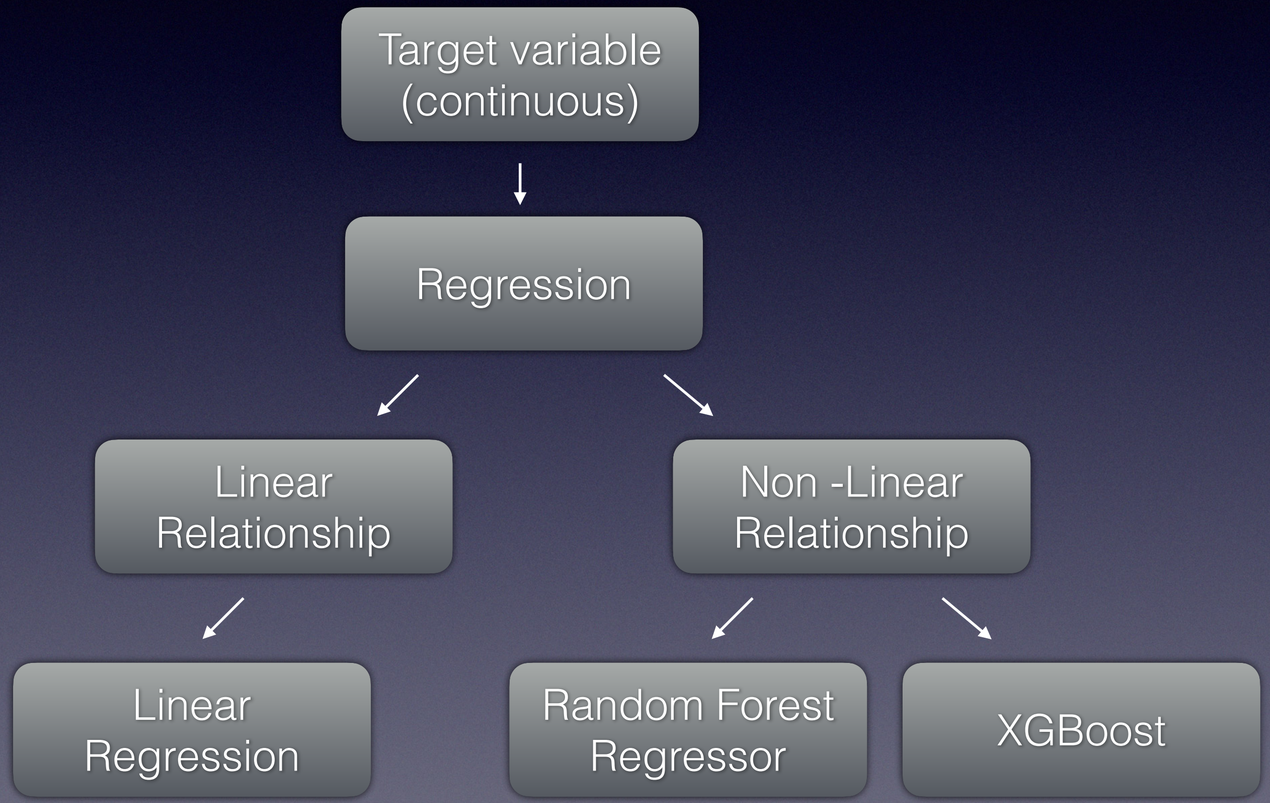
Training	Test
70	30

Machine Learning Algorithms

Choosing the right ML algorithm is crucial.

The choice depends on Problem Type:

- **Type of problem: Regression Problem** (Target, units_sold is continuous)
- **Nature of the data: Supervised learning as Labeled data**
- **Goals of the analysis: Predict a continuous numerical value**
- **Model performance: Mean Squared Error (MSE), R-squared (R^2)**



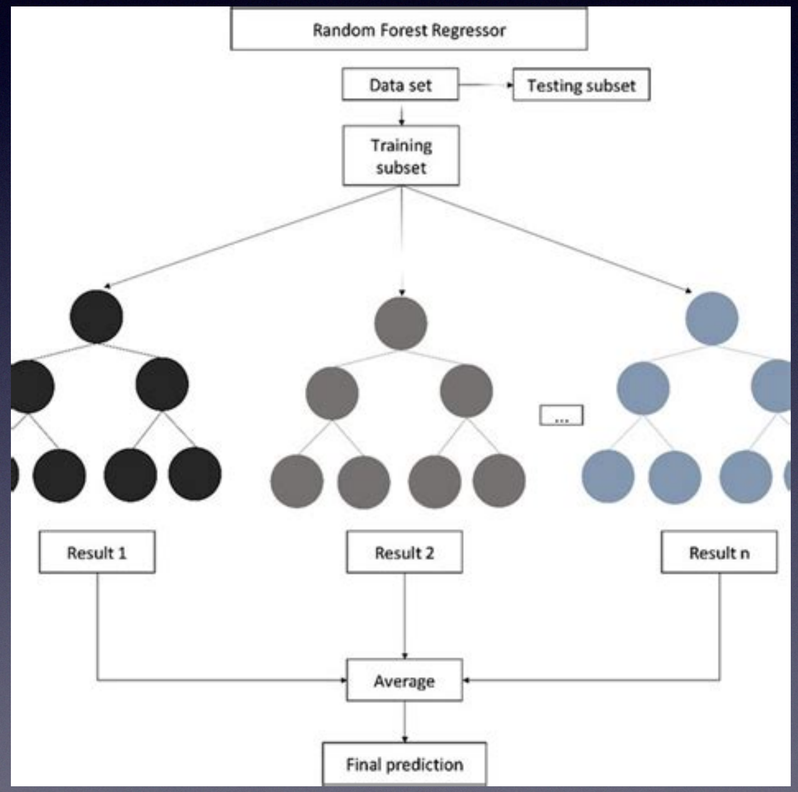
ML steps

	Linear Regression	Random Forest	XGBoost
Import libraries	scikit-learn	scikit-learn	xgboost
Initialize the model	LinearRegression()	RandomForestRegressor()	xgb.XGBRegressor()
Train the model	model.fit(X_train, y_train)	model.fit(X_train, y_train)	model.fit(X_train, y_train)
Predict on test data	model.predict(X_test)	model.predict(X_test)	model.predict(X_test)
Evaluate the model	Mean absolute error: 30.83 R^2: 0.22	Mean absolute error: R^2:	Mean absolute error: 20.11 RMSE: 34.96
Hyperparameter Tuning	GridSearchCV cv=3, n_jobs=-1	GridSearchCV n_est = [10,50,100]	GridSearchCV n_est = [10,50,100]
Model Performance	mean_absolute_error: 25.89	mean_absolute_error: 13.44	mean_absolute_error: 13.96

Random Forest Algorithm Overview

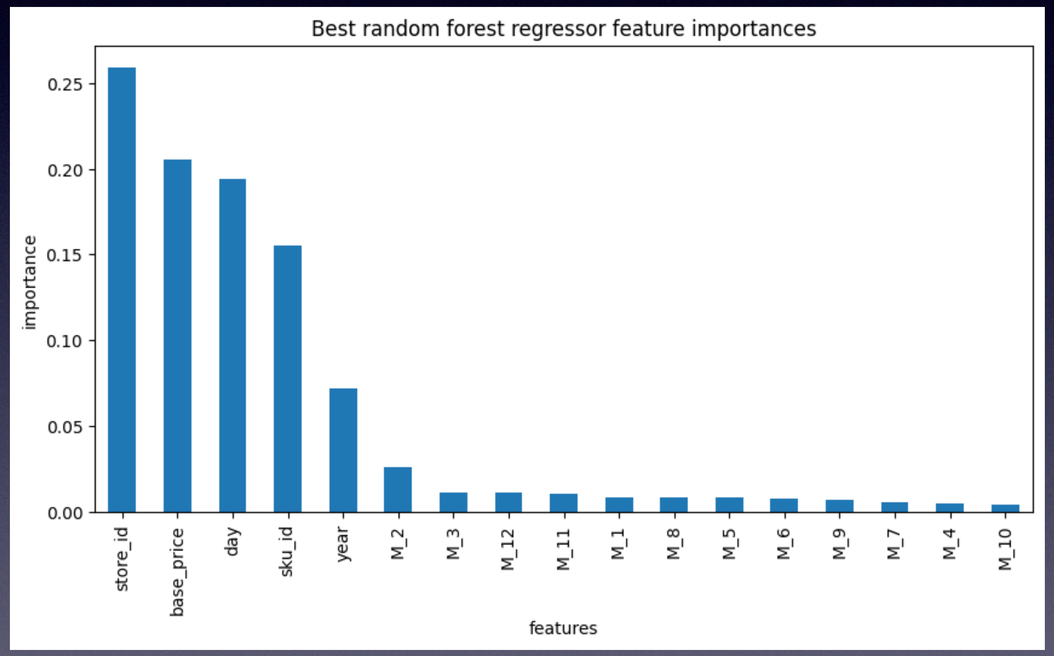
Key Features:

- An ensemble learning method based on decision trees.
- Combines multiple decision trees to improve predictive accuracy.
- Suitable for both classification and regression tasks.
- Handles large datasets well and can model non-linear relationships too.
- Robust to overfitting (due to averaging multiple trees).
- Provides feature importance metrics.



Modeling Results and Analysis

Feature Importance	Feature name
1	total_price
2	base_price
3	sku_id_219009
4	day
5	store_id_9845
6	store_id_8023
7	sku_id_222087
8	month_2
9	sku_id_216418
10	store_id_9112



Identified dominant features using Random forest regression model

Model Evaluation: The random forest model has a lower cross-validation with least mean absolute error of 13.45

Exhibits less variability. Robust to outliers too.

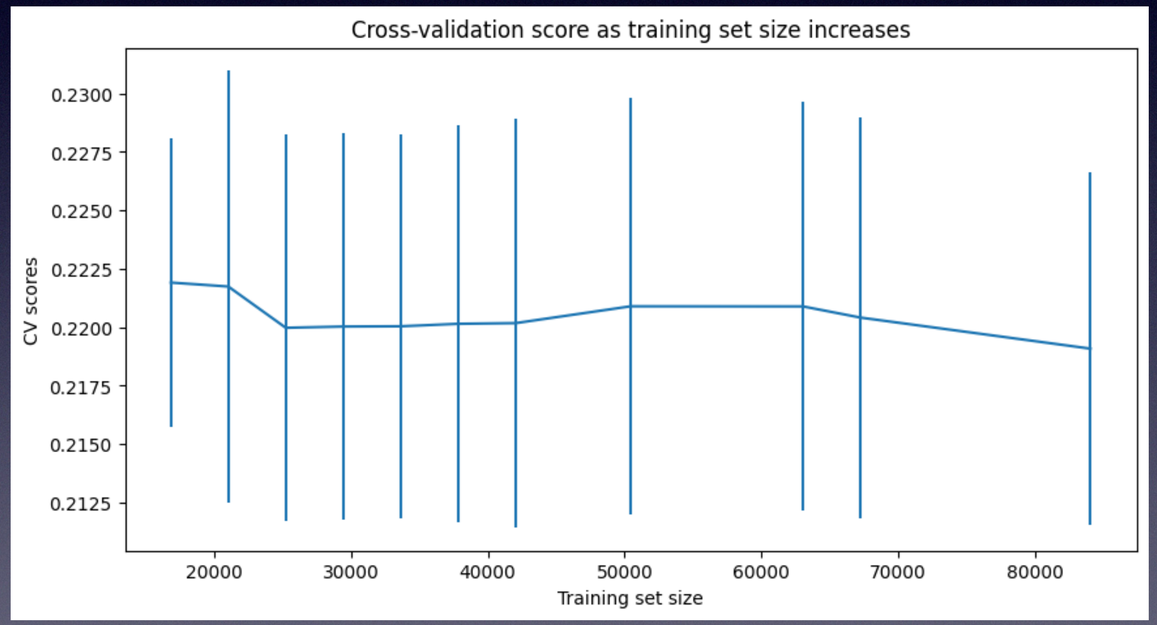
Data quantity assessment

Large Dataset, ideal for Random forest model

Data sufficiency crucial to capture the patterns

Per model CV score, the model performance quite leveled off.

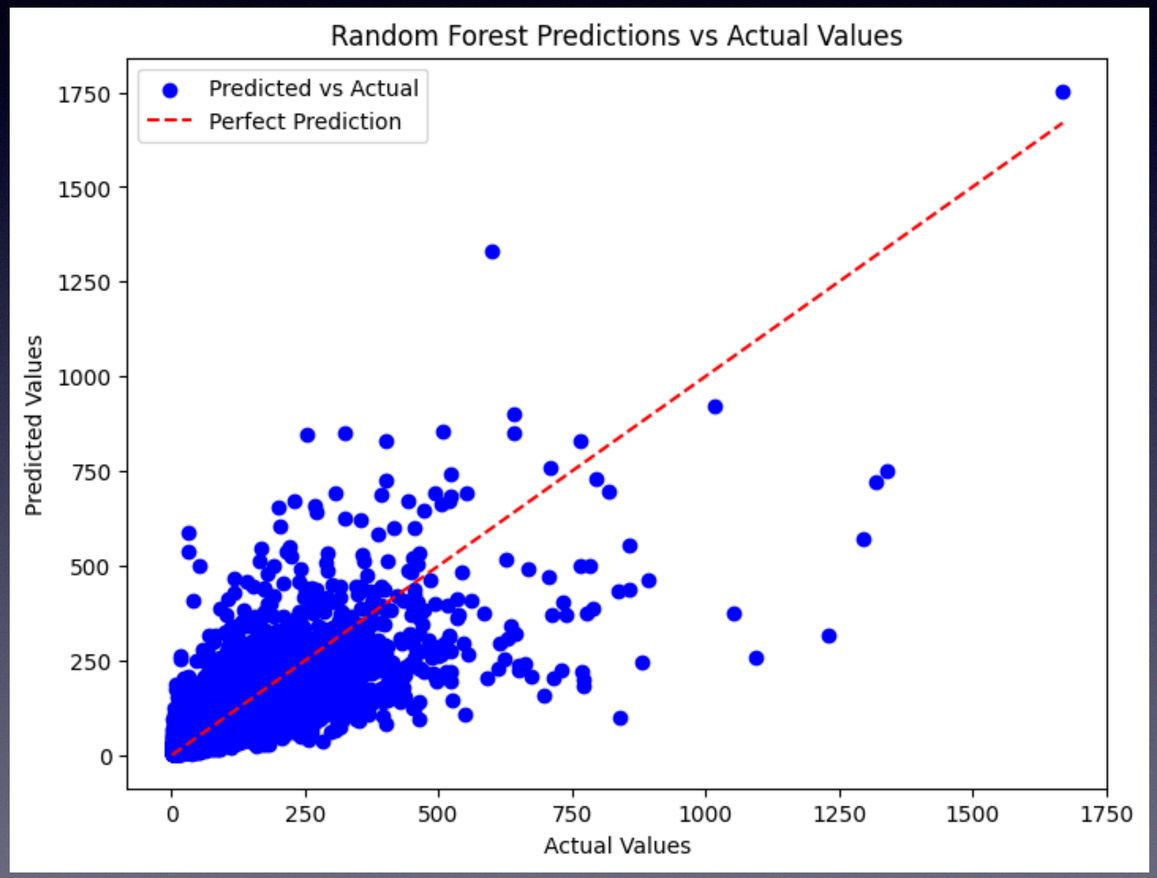
- Saw an increase initially in model performance which then reduced slightly, then stayed same.



Visualize the Predictions

Plotted the predicted values against the actual values (y_test).

- Each point on the scatterplot represents a pair of **actual** and **predicted** values.
- The **red dashed line** represents perfect predictions (y = x). If all the points lie on this line, the model has made perfect predictions.



Recommendation and Key Findings

- Add more relevant features like external factors (economic indicators, holidays, etc.) to improve predictions.
- Determine specific products belong to sku id’s, 219009, 222087 and 216418 to predict future demand accurately.
- Research economical and environmental factors highlighting store id’s 9845, 8023, 9112.
- Include external data sources such as marketing campaigns, competitor actions to stay updated.
- Tune the model periodically with fresh data to keep it up to date.