ROSSMANN

Store Sales Prediction

Paul Marten

Context

Rossmann operates over 3,000 drug stores in 7 European countries. Currently, Rossmann store managers are tasked with predicting their daily sales for up to six weeks in advance. Store sales are influenced by many factors, including promotions, competition, school and state holidays, seasonality, and locality. With thousands of individual managers predicting sales based on their unique circumstances, the accuracy of results can be quite varied.

The Problem

How can we best predict the Rossmann stores' daily sales up to six weeks in advance, through means of using data about each location and past sales to formulate a regression model that will minimize percentage error?

Key Data Sources

The data is obtained from the Kaggle competition titled "Rossmann Store Sales." The data can be found here: https://www.kaggle.com/c/rossmann-store-sales/data

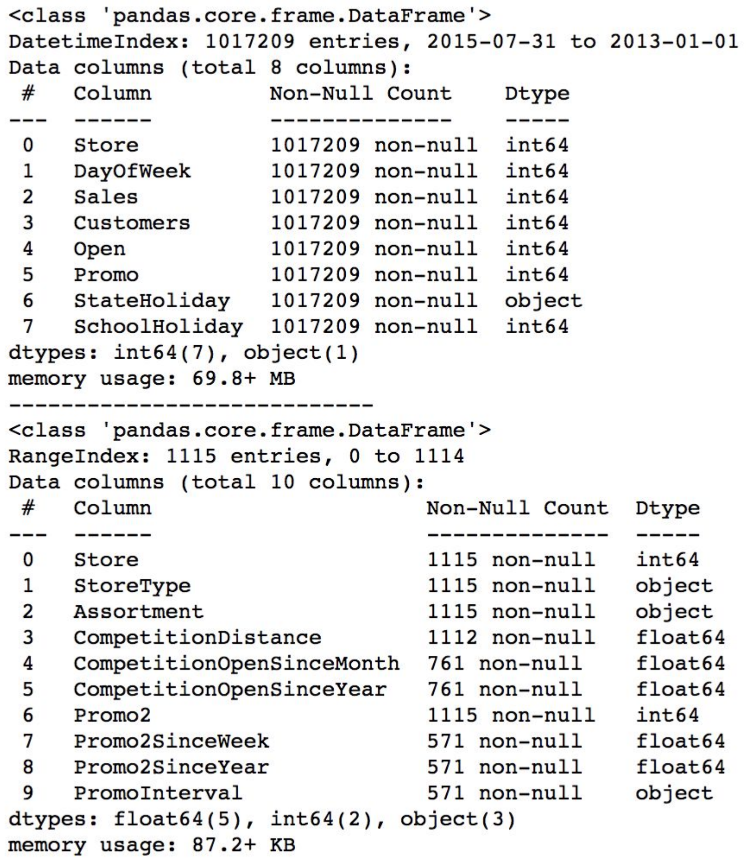
- **train.csv** - historical data including sales (9 features).
- **store.csv** - supplemental information about the stores (10 features).

Data Wrangling

★ Loading

★ Cleaning

★ Transforming

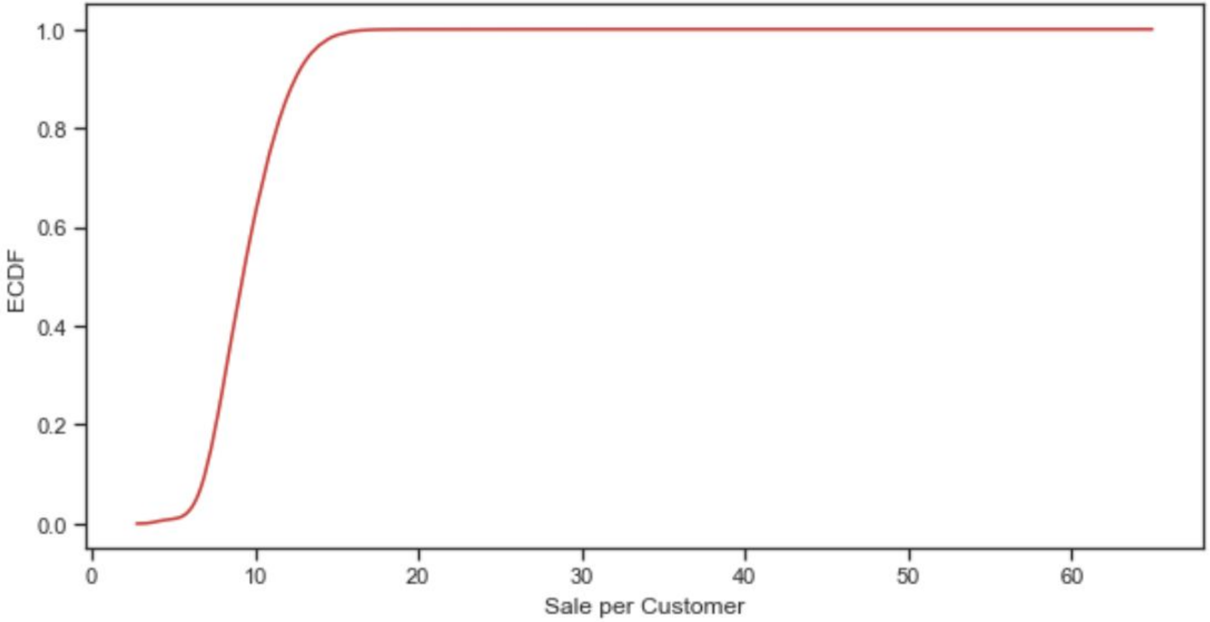


```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 1017209 entries, 2015-07-31 to 2013-01-01
Data columns (total 8 columns):
 #   Column         Non-Null Count    Dtype 
---  ------         --------------    ----- 
 0   Store          1017209 non-null  int64 
 1   DayOfWeek      1017209 non-null  int64 
 2   Sales          1017209 non-null  int64 
 3   Customers      1017209 non-null  int64 
 4   Open           1017209 non-null  int64 
 5   Promo          1017209 non-null  int64 
 6   StateHoliday   1017209 non-null  object
 7   SchoolHoliday  1017209 non-null  int64 
dtypes: int64(7), object(1)
memory usage: 69.8+ MB
----------------------------
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1115 entries, 0 to 1114
Data columns (total 10 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   Store                      1115 non-null   int64  
 1   StoreType                  1115 non-null   object 
 2   Assortment                 1115 non-null   object 
 3   CompetitionDistance        1112 non-null   float64
 4   CompetitionOpenSinceMonth  761 non-null    float64
 5   CompetitionOpenSinceYear   761 non-null    float64
 6   Promo2                     1115 non-null   int64  
 7   Promo2SinceWeek            571 non-null    float64
 8   Promo2SinceYear            571 non-null    float64
 9   PromoInterval              571 non-null    object 
dtypes: float64(5), int64(2), object(3)
memory usage: 87.2+ KB
```

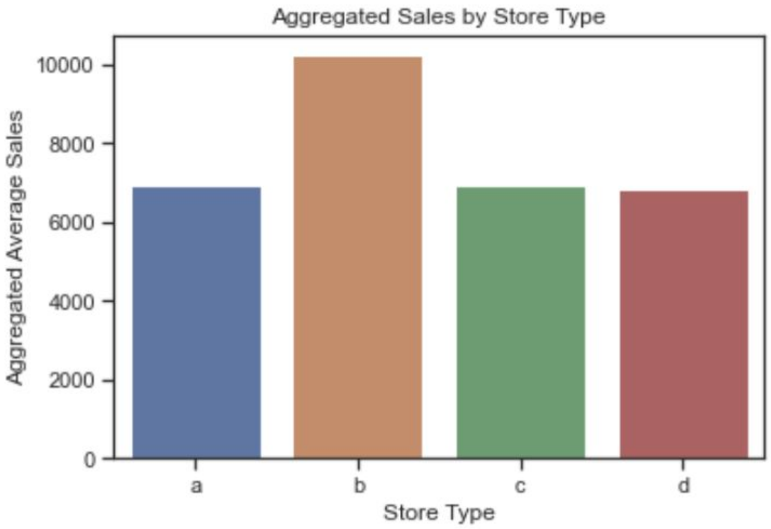
Exploratory Data Analysis

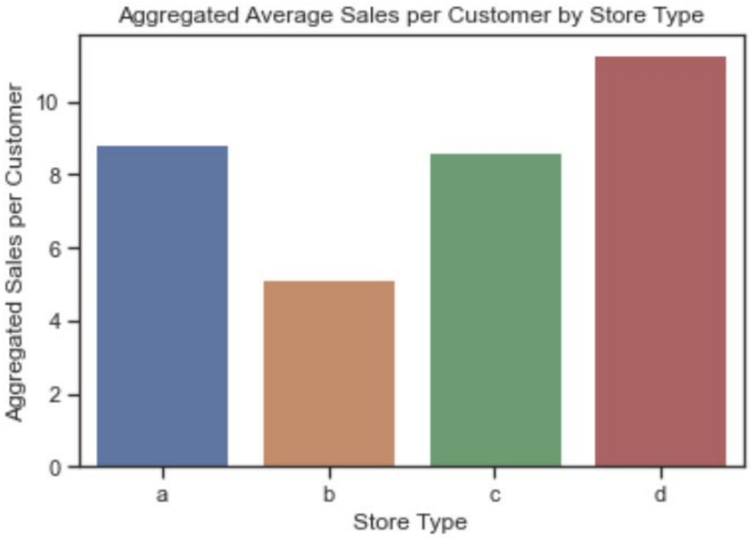
- ★ Understand how the sales are distributed given the influence of various features.
- ★ Determine if the data has consistent patterns that imply predictable outcomes.
- ★ Investigate how correlated the features are with the target variable as well as one another.

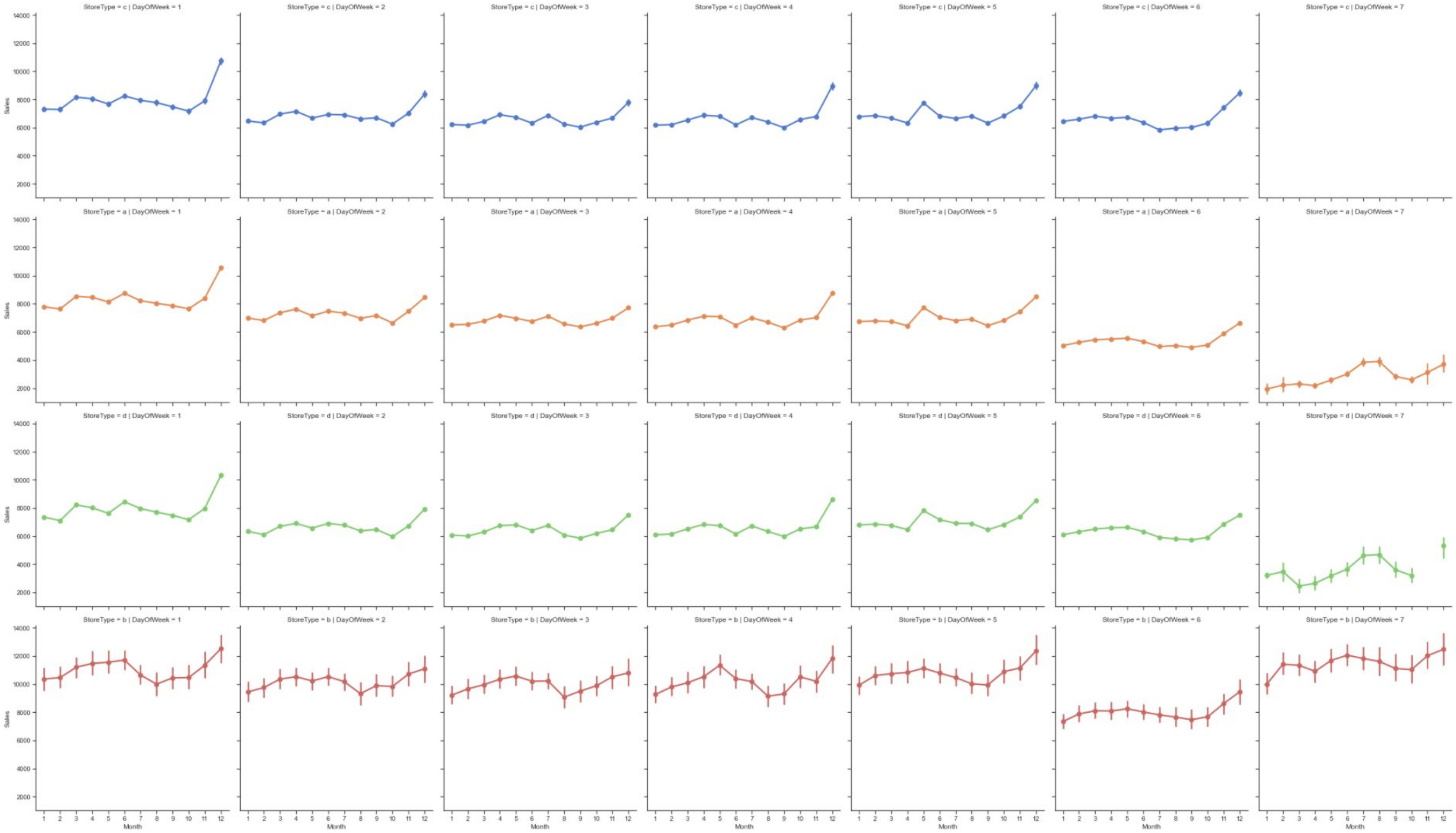
Exploratory Data Analysis



Exploratory Data Analysis

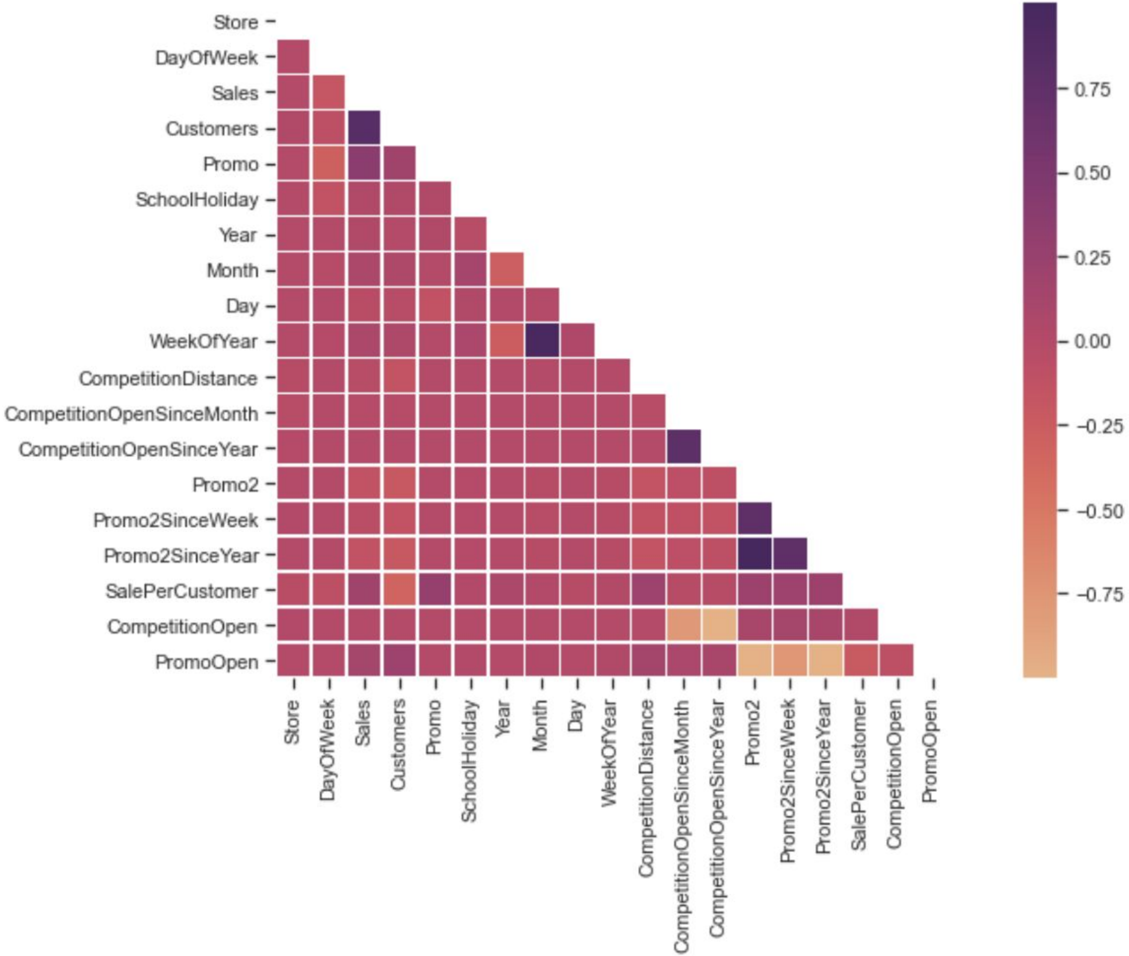





StoreType = c | DayOfWeek = 1
StoreType = c | DayOfWeek = 2
StoreType = c | DayOfWeek = 3
StoreType = c | DayOfWeek = 4
StoreType = c | DayOfWeek = 5
StoreType = c | DayOfWeek = 6
StoreType = c | DayOfWeek = 7
StoreType = a | DayOfWeek = 1
StoreType = a | DayOfWeek = 2
StoreType = a | DayOfWeek = 3
StoreType = a | DayOfWeek = 4
StoreType = a | DayOfWeek = 5
StoreType = a | DayOfWeek = 6
StoreType = a | DayOfWeek = 7
StoreType = d | DayOfWeek = 1
StoreType = d | DayOfWeek = 2
StoreType = d | DayOfWeek = 3
StoreType = d | DayOfWeek = 4
StoreType = d | DayOfWeek = 5
StoreType = d | DayOfWeek = 6
StoreType = d | DayOfWeek = 7
StoreType = b | DayOfWeek = 1
StoreType = b | DayOfWeek = 2
StoreType = b | DayOfWeek = 3
StoreType = b | DayOfWeek = 4
StoreType = b | DayOfWeek = 5
StoreType = b | DayOfWeek = 6
StoreType = b | DayOfWeek = 7
Sales
14000
12000
10000
8000
6000
4000
2000
1 2 3 4 5 6 7 8 9 10 11 12
Month

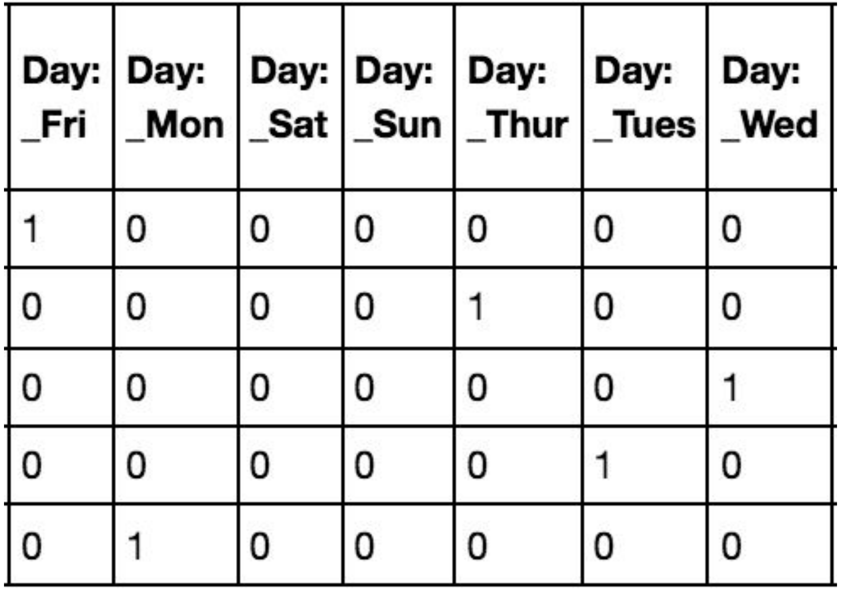
Feature Correlations

The darker the color of the tile, the more correlated the intersecting features are. The important part here is the third column of tiles, showing relations between sales and the other numeric variables.



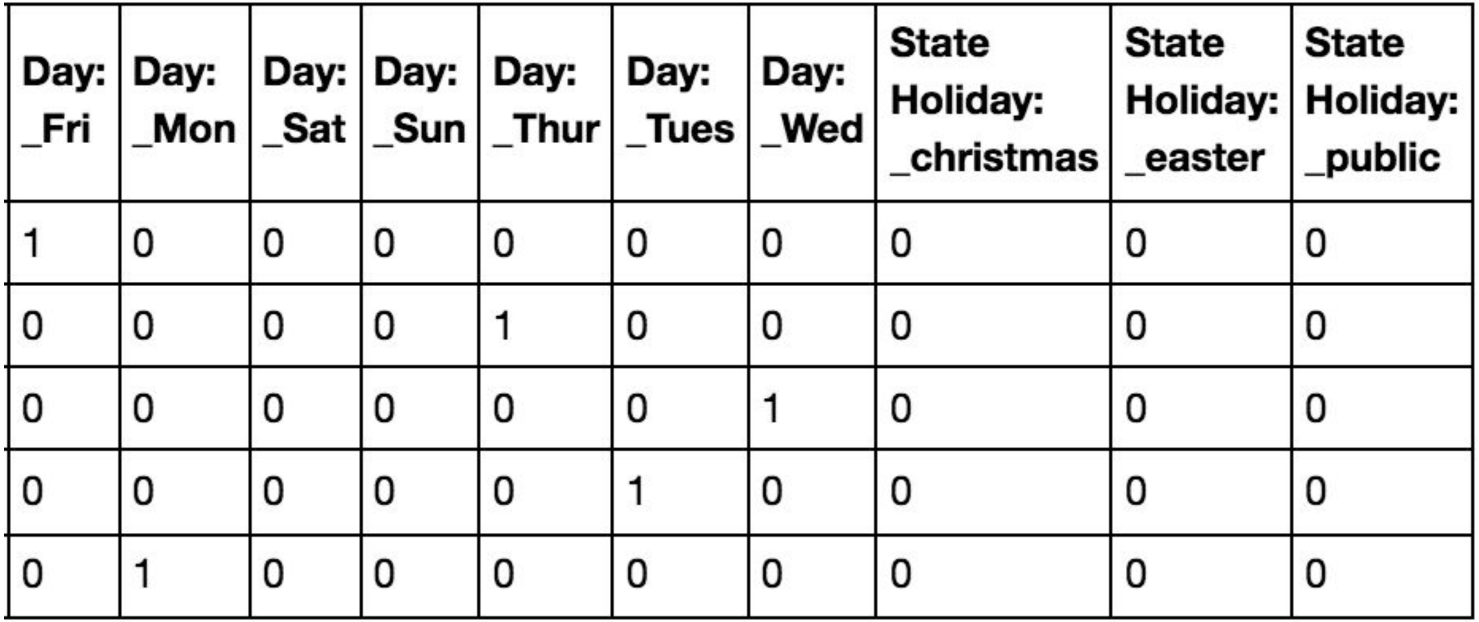
Feature Engineering

- Remove any leftover columns that would not be useful for modeling (for example, store id).
- Create dummy variables via one-hot encoding for the remaining categorical variables (example shown).
- Ended this process with 37 features, and further reduced the dimensionality during preprocessing.



Day: _Fri	Day: _Mon	Day: _Sat	Day: _Sun	Day: _Thur	Day: _Tues	Day: _Wed
1	0	0	0	0	0	0
0	0	0	0	1	0	0
0	0	0	0	0	0	1
0	0	0	0	0	1	0
0	1	0	0	0	0	0

Feature Engineering



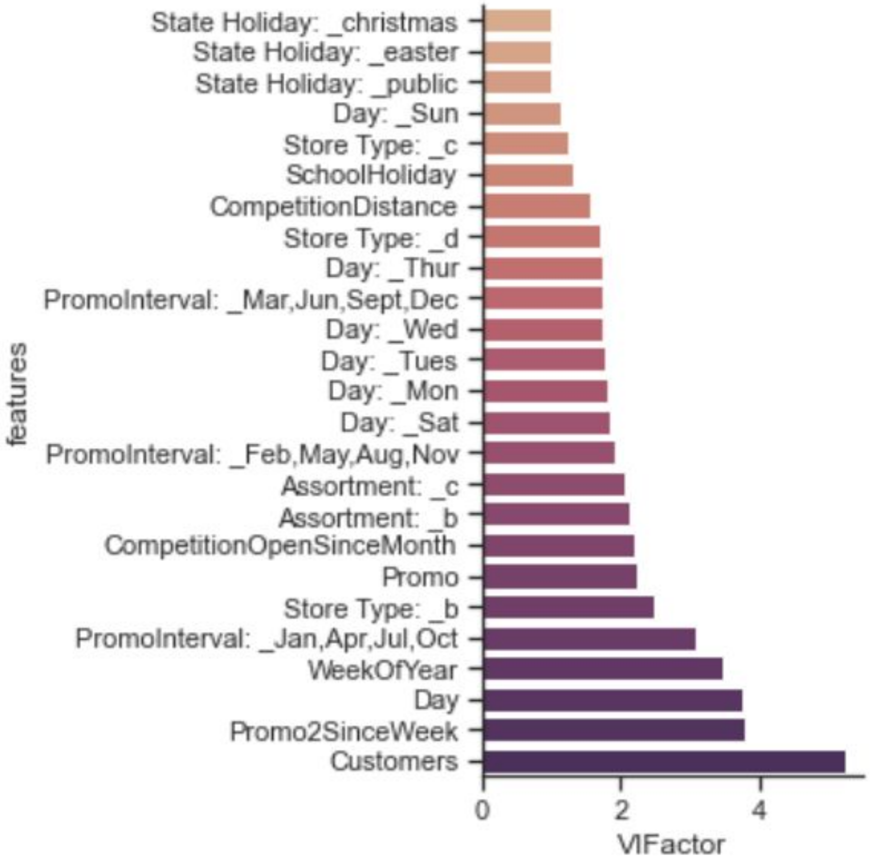
Day: _Fri	Day: _Mon	Day: _Sat	Day: _Sun	Day: _Thur	Day: _Tues	Day: _Wed	State Holiday: _christmas	State Holiday: _easter	State Holiday: _public
1	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0	0	0
0	1	0	0	0	0	0	0	0	0

Preprocessing

To process my data for modeling, I filtered the features using a variance inflation factor function, which accepts the dataframe as input, and compares the VIF thresholds of the features in order to determine which features have high correlation with one another. In a regression model, this is referred to as multicolinearity, and hinders the predictability of the model.

Final Features

I was left with 25 features. Using these 25, I subset the dataframe to only include those features. With the final dataframe, I split my data into the train/test split I would be using for modeling, opting for a test size of 25%.



Modeling

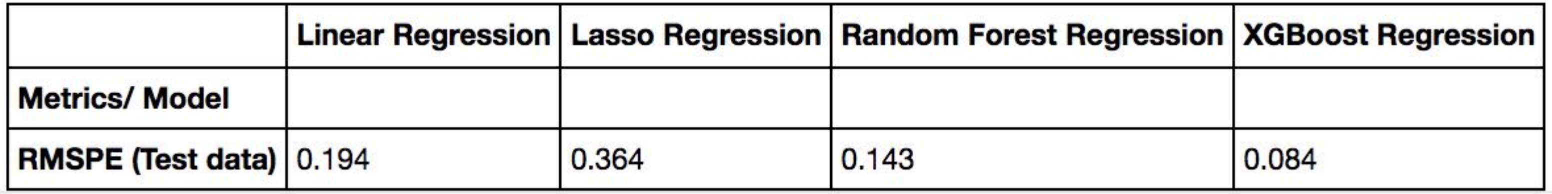
Models:

- Linear Regression
- Lasso Regression
- Random Forest Regression
- XG Boost Regression

Evaluation:

$$\mathrm{RMSPE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{y_i}\right)^2}$$

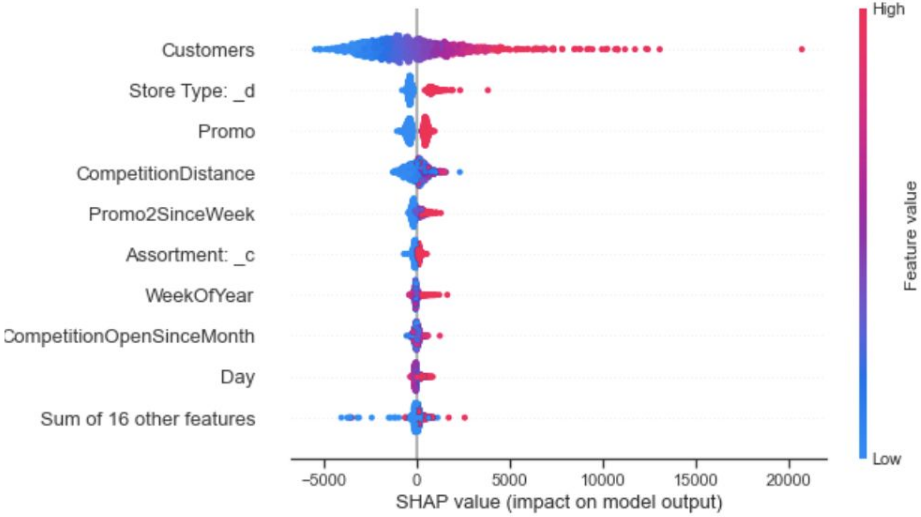
Model Results



	Linear Regression	Lasso Regression	Random Forest Regression	XGBoost Regression
Metrics/ Model				
RMSPE (Test data)	0.194	0.364	0.143	0.084

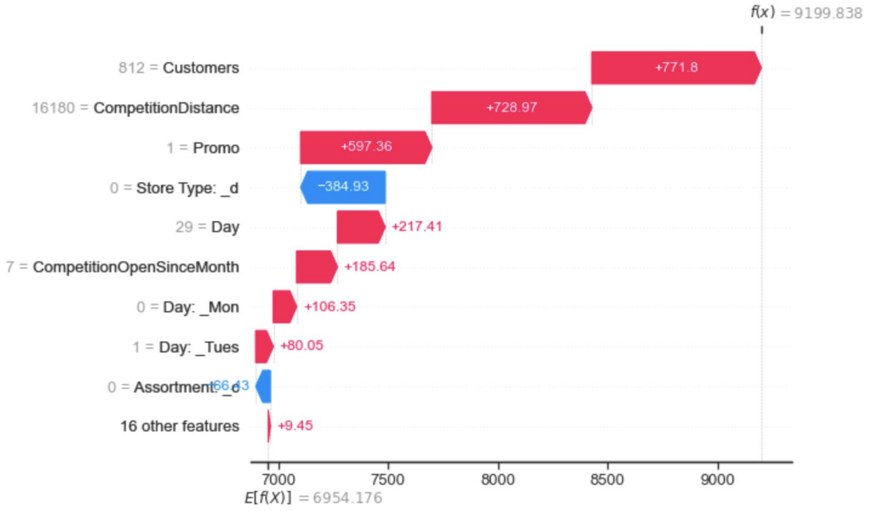
The lower the RMSPE, the lower the percentage error, and the better the model performed. The XG Boost regressor was the victor, almost twice as good as the next best model.

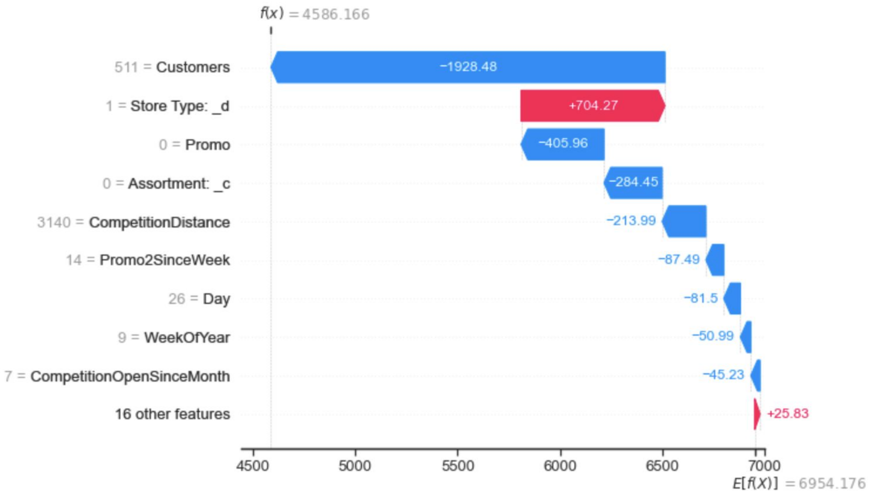
SHAP Analysis



SHAP (SHapley Additive exPlanations) is used to determine a feature’s role in the predictability of a model. From top to bottom, the features are ranked in importance. Their color indicates how strong their impact is. Their horizontal position indicates whether this impact was positive or negative.

SHAP Analysis





Conclusions

- The top 4 features driving sales are **Customers**, **StoreType: _d**, **Promo**, and **CompetitionDistance**. More customers means more sales. Store type d has the highest average sale per customer. Promotions attract more customers, which means more sales. Close competition can hinder a store's sales.
- The XG Boost regressor did a great job of predicting sales, with a very low RMSPE score. It would be in the best of interest of Rossmann to consider implementing such a model for their predictions.
- For a more accurate prediction, it would be useful to have more information on the top features, that being how the store types vary, how competition operates, etc.
- For future scope, I would like to perform a time series analysis on the data to further understand how sales have evolved over time and will continue to evolve.