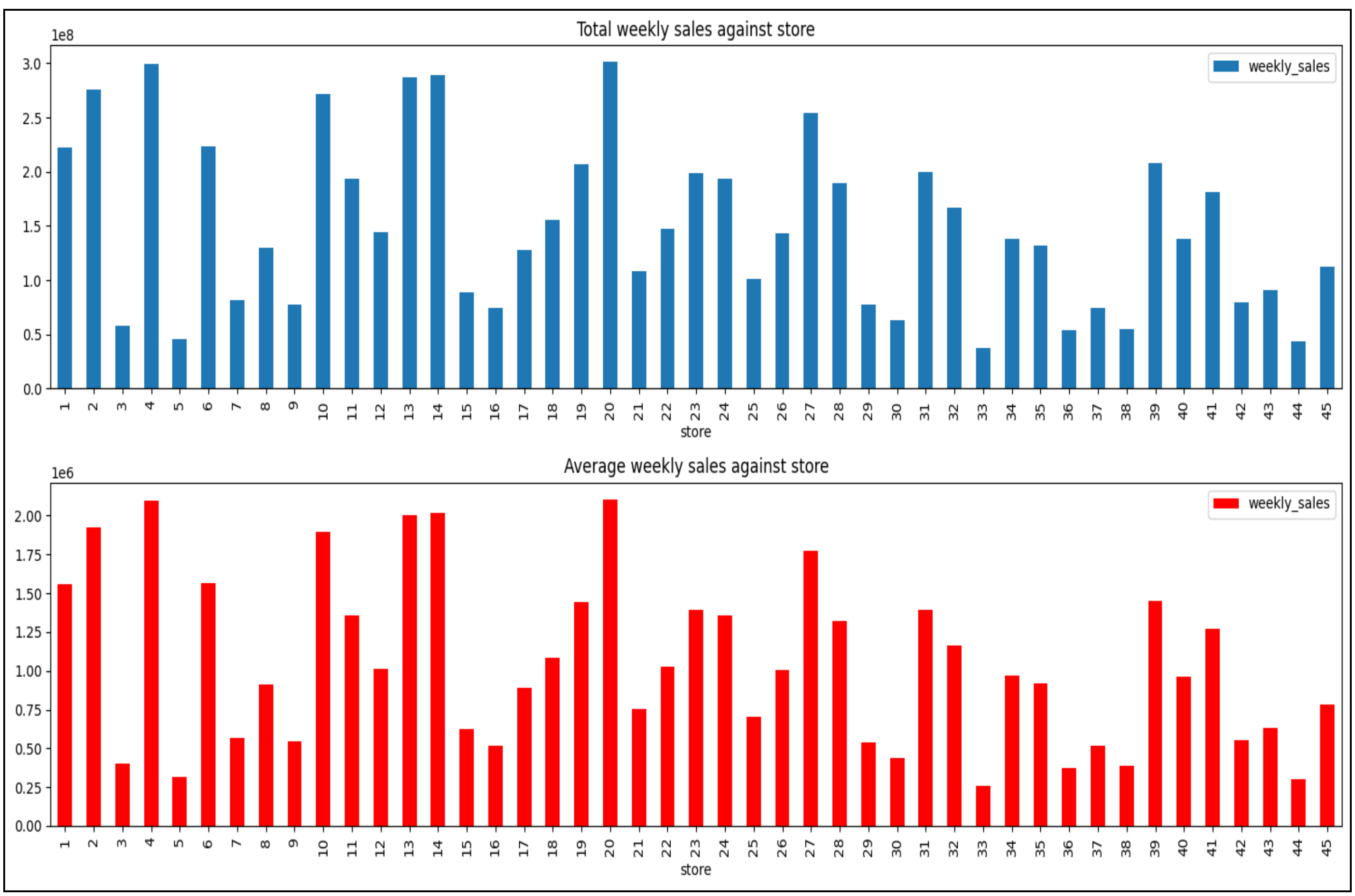
Predicting Weekly Sales for Walmart Retail Stores
Capstone 2 Project
Damilola T. Olaiya

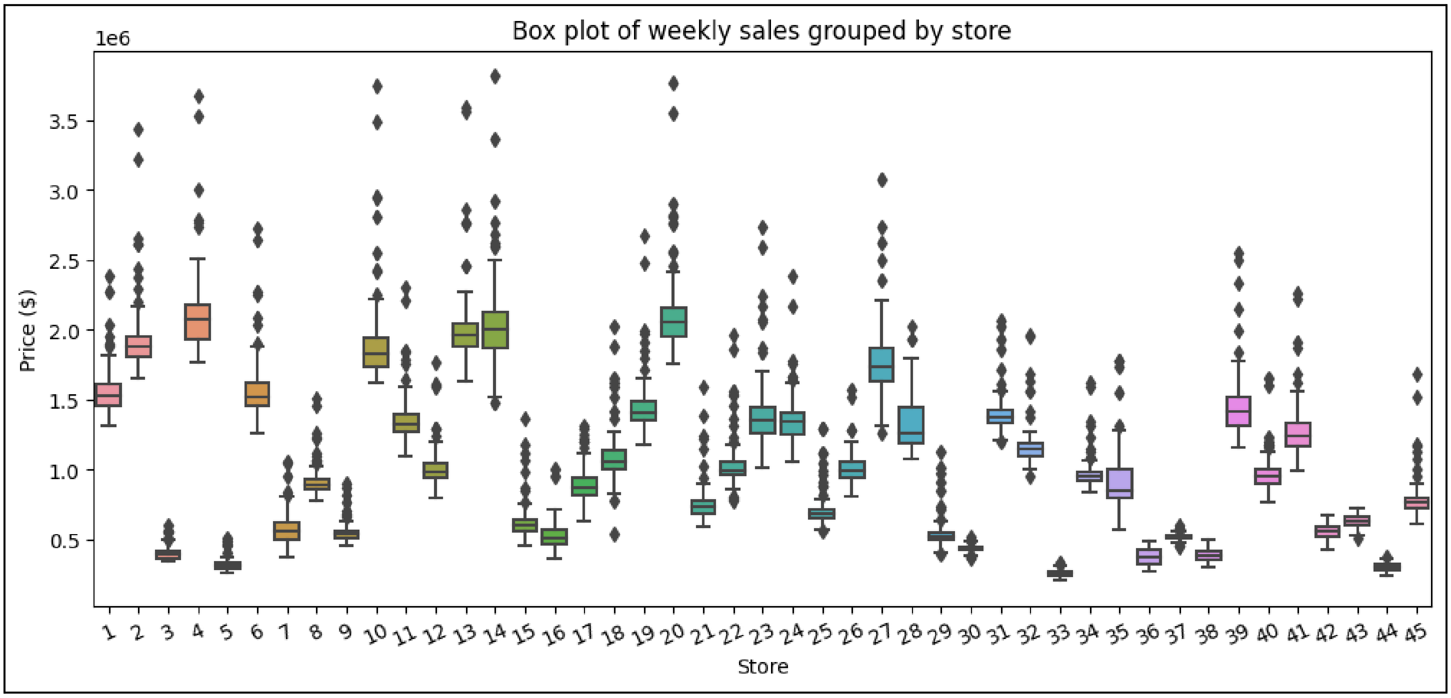
Audience
The audience includes:
1. A combination of executive + technical professionals
2. Springboard
3. Potential future employers

Proposal
1. Hypothesis → How can Walmart use its reported sales data to (i) predict and take advantage of future sales/demand and (ii) potentially improve inventory allocation/scheduling?
2. Criteria for success → Creating a model that can accurately predict the sales with regards to single and multiple features

Data Wrangling
1. The dataset contains sales information from 45 walmart stores and was obtained from here.
2. There were 7 features/columns were store, weekly_sales, holiday_flag, temperature, fuel_price, cpi and unemployment.
    a. The target feature is weekly_sales.
3. There were 6435 entries/samples in the data. This corresponded to 143 entries each for 45 stores.
4. There were no missing values. The data was clean.
5. Store-to_store analysis yielded the following:
    a. Total and Average weekly sales per store

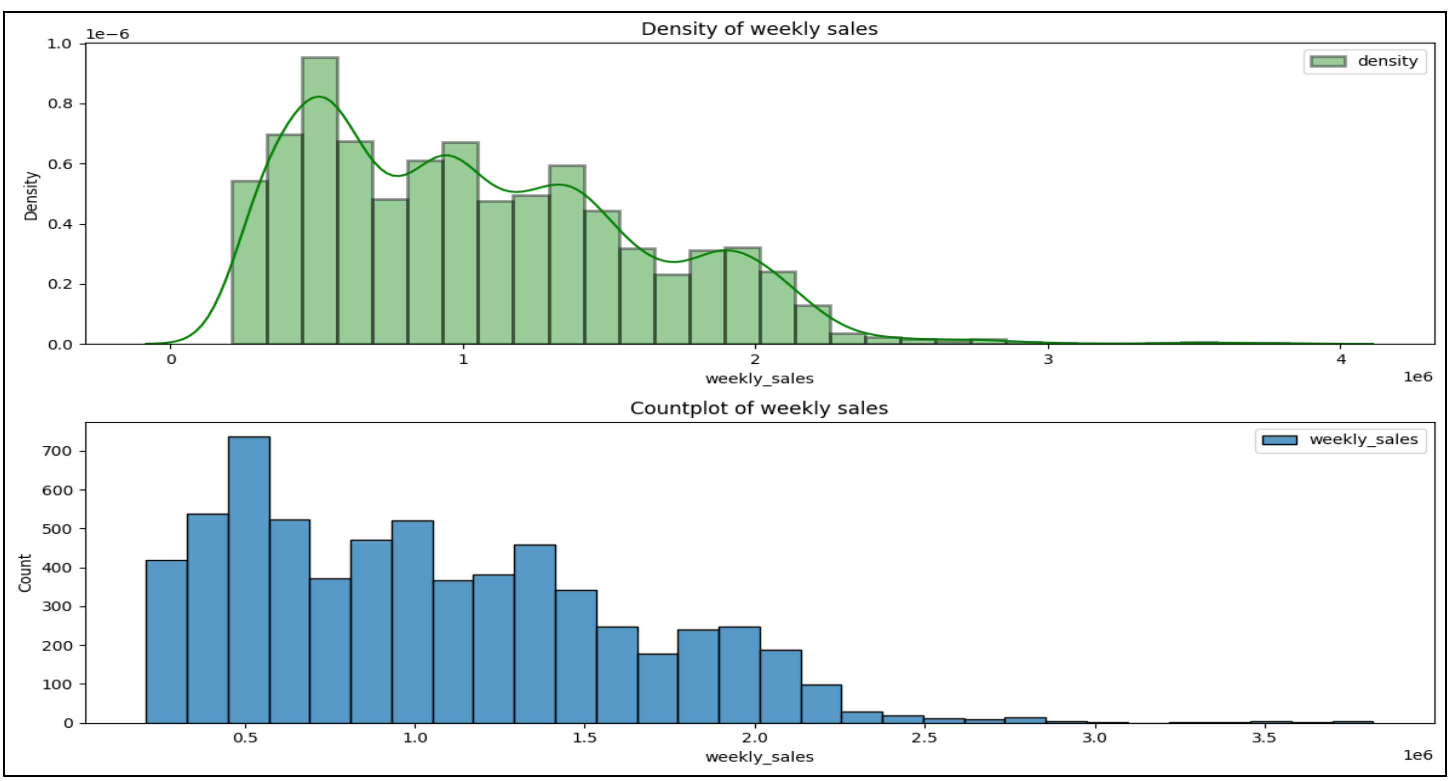


b. Boxplot of weekly sales grouped by store

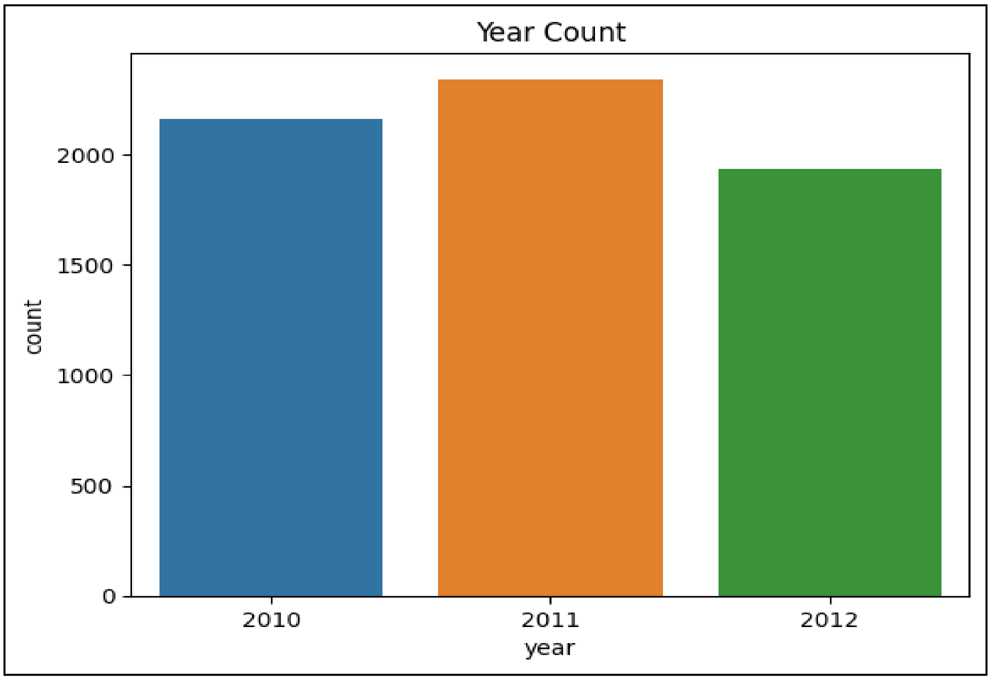


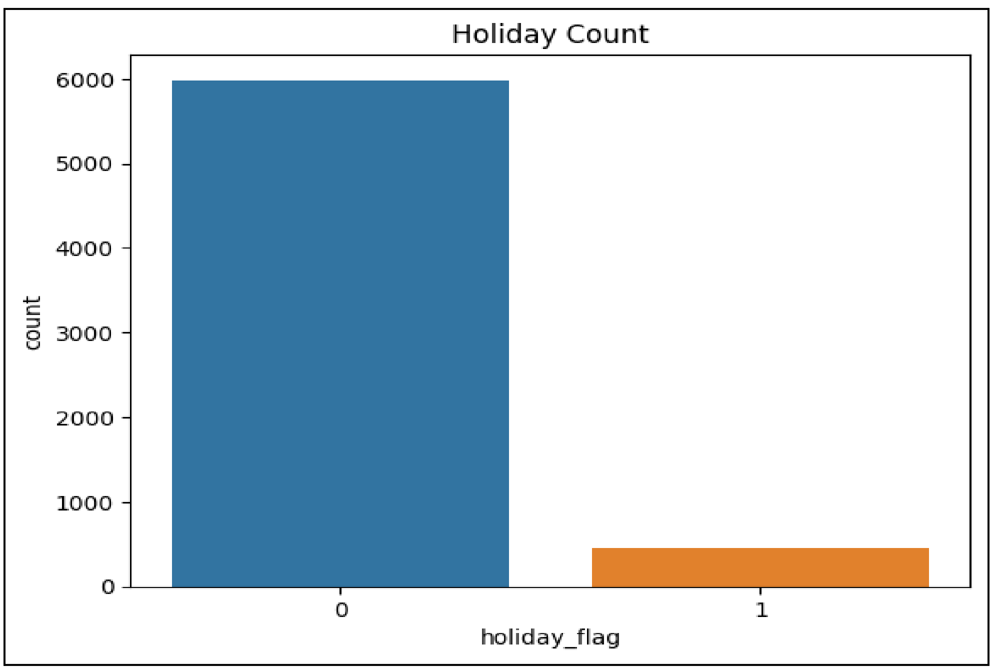
Exploratory Data Analysis (EDA)

1. The date feature was removed and expanded into its own separate dataframe. This included extracting weekday, month and year from each sample.
2. Categorical and numeric features were parsed and listed out. There were 2 categorical (holiday_flag, store) and 4 numerical features (unemployment, fuel_price, cpi, temperature, weekly_sales).
3. As expected, there are far more non-holidays than holidays.
4. EDA produced several infographics including:
    a. Distribution and Countplot of weekly sales



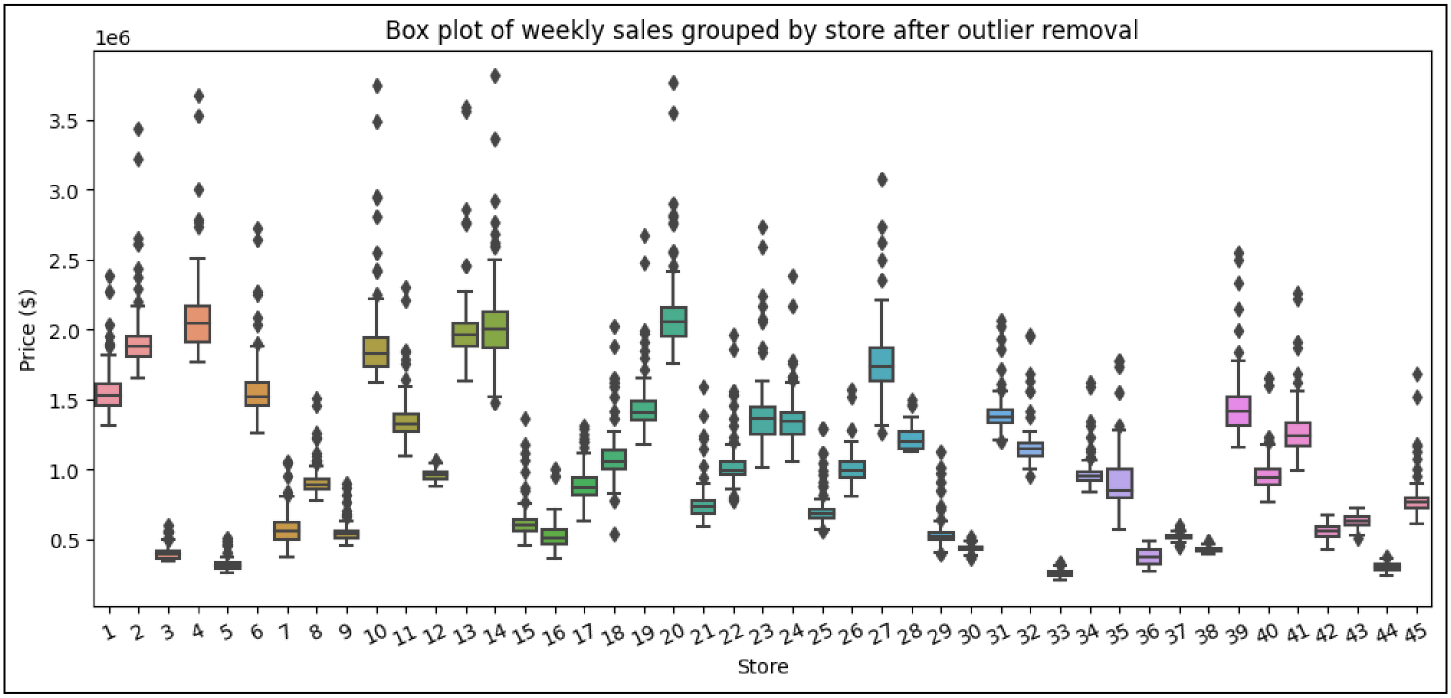
b. Year and Holiday Count plots





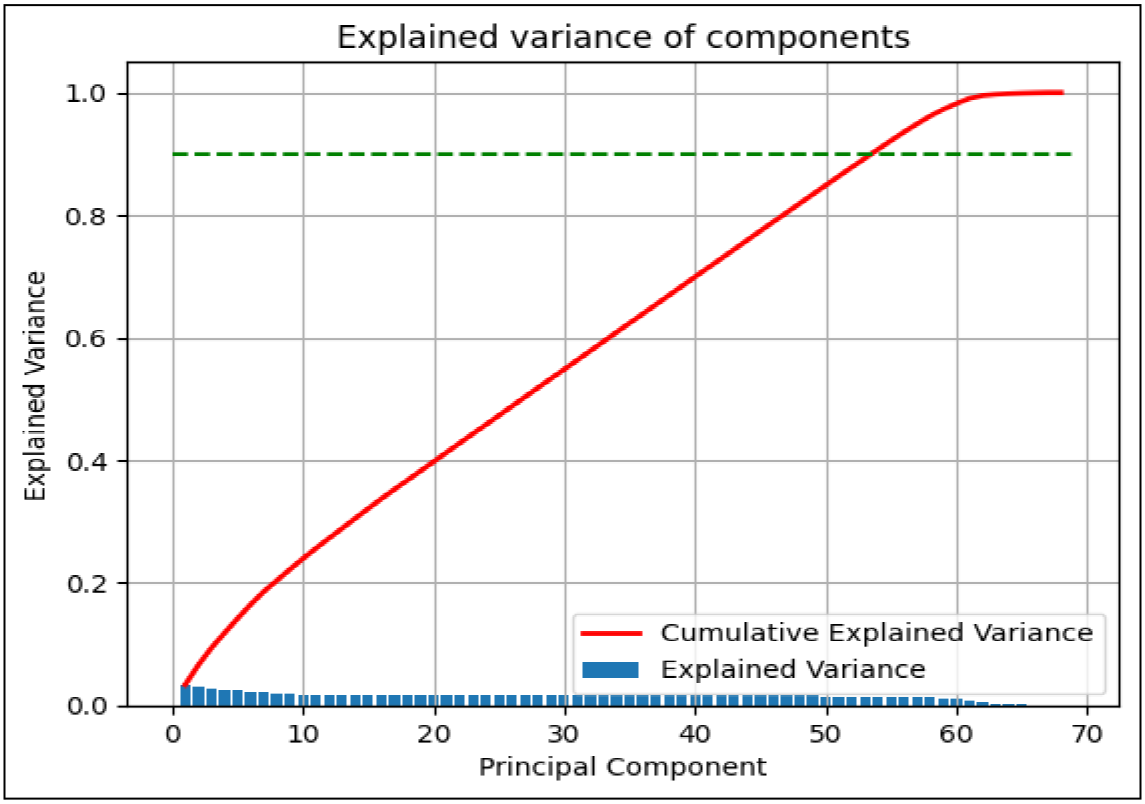
Pre-processing and Training

1. The data was checked for uniqueness and no duplicate rows were found.
2. Outliers were removed using the IQR (inter quartile range). This involved dropping all samples/rows from numerical features whose values were greater than (75th percentile + 1.5 * IQR) or less than (25th percentile + 1.5 * IQR).
    a. This resulted in the number of samples going from 6435 to 5951 (7.52% drop) and yielded the following boxplot



3. The holiday_flag feature was one-hot encoded (the values were 1 for holiday and 0 for non-holiday) and dummy variables were created for store, weekday, month and year features.
    a. The first column was dropped in each case to prevent issues of multicollinearity.
    b. The end result was a dataframe with 69 features/columns.
4. The data was split into X_train, X_test, y_train and y_test.
    a. It was then scaled/standardized to have a mean of 0 and a standard deviation of 1.
    b. Note that the standard scaler was fit and transformed using X_train only then used to transform X_test. This was done so that the model would be completely unaffected by the testing data.

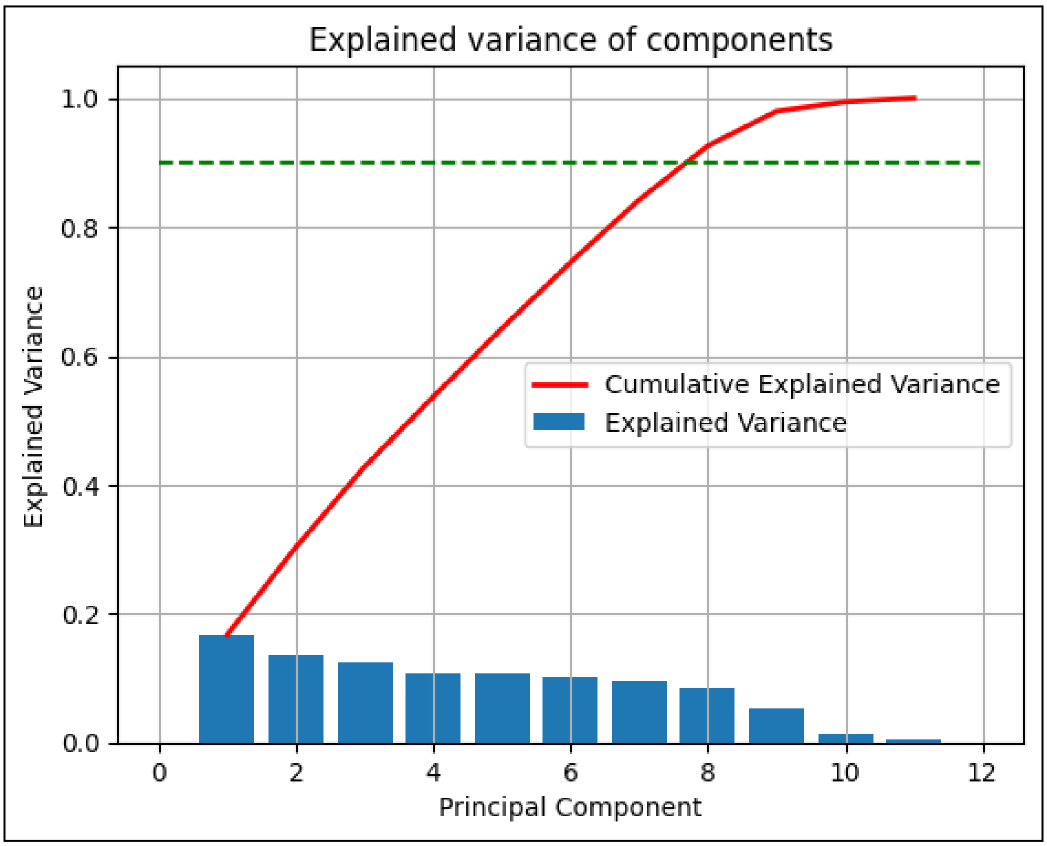
5. Principal component analysis (PCA) was performed on the data.
   a. Using all features (no reduction), the variance was explained as such:



   b. Using PCA and linear regression, RMSE was calculated on the train and test datasets for features ranging in number from 1 to 69. Naturally, the error reduced with more features as more variance was explained.

Modeling

1. Using dummy variables for all categorical features makes the data too granular and convoluted (69 features/columns) as evidenced by the PCA decomposition result from pre-processing.
    a. Going forward, I assumed that all stores (store 1 to store 45) are within the same market segment and ignored store to store differences.
    b. Additionally, the holiday_flag feature did not need to be standardized as the values were already within scale for analysis.
    c. To that effect, I mapped the holiday_flag feature back to 1s and 0s and eliminated the store, month and year features from the data.
    d. This left 11 features remaining in the dataset (holiday_flag, temperature, fuel_price, cpi, unemployment, weekday_1, weekday_2, weekday_3, weekday_4, weekday_5, weekday_6).
2. I was able to infer that feature reduction may be unnecessary as, although 90% of the variance is explained cumulatively by 8/11 principal components, only one of the components had a variance that was significantly lower than the others.
    a. That, combined with the relatively small number of features, may allow us to ignore feature reduction.



3. There were **four** different models used: multiple linear regression, lasso regression, ridge regression and random forest regression.
4. Cross validation was performed for ridge, lasso and random forest regressions using:
    a. alphas of 0.1, 1, 10, 100,1000 and 10000 for lasso and ridge regression
    b. parameter {n_estimators: [300,400,500], max_depth:[4,6,8], min_samples_leaf :[0.1,0.2], max_features:['log2','sqrt']} for random forest regression
    c. The best alpha for both lasso and ridge was found to be **100**.
    d. The best parameters for the random forest regression were found to be **{'max_depth': 6, 'max_features': 'log2', 'min_samples_leaf': 0.1, 'n_estimators': 400}**.
5. During modeling, the Model Evaluation Comparison Matrix (MECM) was created and populated
    a. When sorted by **increasing Test RMSE**, I obtained:

	Train-R2	Test-R2	Train-RSS	Test-RSS	Train-MSE	Test-MSE	Train-RMSE	Test-RMSE
Random Forest Regression Model (RF)	0.036973	0.040372	1.490568e+15	3.847395e+14	3.131446e+11	3.230391e+11	559593.245026	568365.259504
Lasso Linear Regression (LLR)	0.019928	0.018191	1.516950e+15	3.936325e+14	3.186869e+11	3.305059e+11	564523.605053	574896.441420
Ridge Linear Regression (RLR)	0.019971	0.017923	1.516884e+15	3.937400e+14	3.186731e+11	3.305961e+11	564511.408328	574974.888760
Multiple Linear Regression (MLR)	0.020051	0.017895	1.516760e+15	3.937511e+14	3.186470e+11	3.306055e+11	564488.278884	574983.013229

Inference

1. Lower RMSE implies a better the model. That said, a significant disparity between training and testing scores would suggest overfitting.
2. All regression models were fairly similar in terms of training and test R2 and RMSE.
3. However, Multiple linear regression performed best in training metrics but worst in test metrics suggesting that it was slightly overfitting.
    a. This is in line with what we would expect from lasso and ridge regression which work to combat overfitting.
4. Random forest regression performed best for all metrics and gave the best overall results.

Conclusions

1. The dataset was quite small with just 6435 samples initially, which dropped 7.5% after cleaning.
2. Cross validating the Lasso and Ridge regressions allowed us to select the best alpha.
3. We will proceed with the **Random forest regression model** as it performed best.

Further Steps to consider

1. Using pca for feature reduction
2. Using more of the generated features in the regression
3. Testing more parameters in the grid search cv at the cost of time
4. Using random forest with bagging, boosting etc
5. Using a polynomial regression model