# CS 5540 Machine Learning Project Proposal Report (Final)

Danny Satterthwaite and Drew Jepsen

## 1. Introduction

As graduate students entering the workforce this coming summer, finding relevant and interesting data and projects has been top of our minds. As we are both interested in business, customer, and predictive analytics we were excited to find a dataset from a Walmart technical interview geared toward sales forecasting. Ultimately, the more accurately a business can forecast its revenue and identify customer and sales groupings, the more effectively the business can manage its operations and profits. Thus, sales forecasting is incredibly important to every business, large and small. Our goal is to use various machine learning algorithms to predict sales totals and performance metrics given the sales information, discounting information, and consumer information. The details of these types of variables are discussed more in the dataset section. Furthermore, we plan to go the other way, using the core sales data to predict various consumer metrics to see if a potential correlation between consumer measures and Walmart sales is reciprocal. For example, can we predict CPI or fuel prices given Walmart's weekly sales data? With the goal of sales forecasting in mind, core to our analysis will be various implementations of regression and hopefully neural networks. We also plan to use various classification algorithms (discussed below) to identify sales and customer groupings.

Given that this dataset comes from a Walmart interview, there are likely thousands of people who have analyzed the data however we could not find Walmart's official results or winning analysis. That being said, there are countless analyses of the data online. These analyses tend to focus more on exploratory data analysis and solely on predicting sales. While sales forecasting is our primary goal, our analysis is differentiated through our deeper analysis and forecasting (reverse) of the customer metrics (CPI, fuel price, etc).

## 2. Problem Definition and Algorithm

### 2.1 Task Definition

See section 2.2 for a description (on an input and output basis) of the algorithms we plan to use and are in the process of coding up. There are various problems we attempt to answer but there are two main categories: predicting sales and identifying sales and customer categories. Details are included on a per-algorithm basis below.

### 2.3. Dataset

The dataset[5] is comprised of historical sales data from 45 Walmart locations around the country. Sales are tagged by their location, week, and department as well as several other factors. The data is comprised of both the core sales data (store, date, department, weekly sales), data that could affect sales discounts (whether the week is a holiday week and promotional markdowns), as well as variables that correspond to the consumer (region temperature, fuel price, consumer price index, unemployment rate, holidays). The data is labeled, does not require additional hardware, and is split across 4 files which are broken down as follows:

1) "stores.csv"
    a. Contains store information
2) "train.csv"
    a. Contains historical sales data by store by week by department
3) "test.csv"
    a. The same as "train.csv" without a sales column as that's what we are trying to predict
    b. In our project, we will likely do the splitting ourselves but this is nice to have
4) "features.csv"

    Contains additional information on discount and consumer information (CPI, fuel price, etc). This will be core to our analysis.

```
RangeIndex: 421570 entries, 0 to 421569
Data columns (total 20 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   Store         421570 non-null  int64
 1   Dept          421570 non-null  int64
 2   Date          421570 non-null  datetime64[ns]
 3   Weekly_Sales  421570 non-null  float64
 4   IsHoliday     421570 non-null  int64
 5   Temperature   421570 non-null  float64
 6   Fuel_Price    421570 non-null  float64
 7   MarkDown1     421570 non-null  float64
 8   MarkDown2     421570 non-null  float64
 9   MarkDown3     421570 non-null  float64
 10  MarkDown4     421570 non-null  float64
 11  MarkDown5     421570 non-null  float64
 12  CPI           421570 non-null  float64
 13  Unemployment  421570 non-null  float64
 14  Size          421570 non-null  int64
 15  Year          421570 non-null  int32
 16  Month         421570 non-null  int32
 17  Week          421570 non-null  UInt32
 18  Type_B        421570 non-null  bool
 19  Type_C        421570 non-null  bool
dtypes: UInt32(1), bool(2), datetime64[ns](1), float64(10), int32(2), int64(4)
```

Along with this data, we will include new data, collected from online reviews of products. The data will be collected using the Python library, Beautiful Soup, to crawl the Walmart reviews page, collecting the date of a post, the user who posted it, the text message, and the review rating. This text-based data will be used to perform sentiment analysis and further bolster our forecasting models. We will implement the NLP tools from Python libraries NLTK and PyTorch. Following the paper written by Zhao[4], we will categorize our text data and utilize it as a feature alongside those previously stated.

## 2.2 Algorithm Definition

A core part of our analysis revolves around comparing and contrasting different sets of predictors (feature sets) used for the same task. Therefore, we created the following sets (subsets) of features and respective response variables to use in our analysis (note the notations that we use from here on out and we often refer to them generally as "feature sets").

Feature and response variable sets:

- X, y: The full set of data.
    - [Store, Dept, IsHoliday, Temperature, Fuel_Price, Markdown1, Markdown2, Markdown3, Markdown4, Markdown5, CPI, Unemployment, Size, Year, Month, Week, Type_B, Type_C] → (to predict) Weekly Sales
- X_2, y_2: A hypothesis minimal subset of features.
    - [Store, Dept, IsHolday] → (to predict) Weekly Sales
- X_3, y_3: A mix of the minimal set with some consumer pricing indicators.
    - [Store, Dept, IsHolday, Temperature, Fuel_Price, CPI, Unemployment]
- X_4, y_4: An extension of set 3 now including markdown information
    - [Store, Dept, IsHolday, Temperature, Fuel_Price, CPI, Unemployment, Markdown1, Markdown2, Markdown3, Markdown4, Markdown5] → (to predict) Weekly Sales
- X_5, y_5: Predict CPI Using everything else
    - [Store, Dept, Weekly_Sales, IsHoliday, Temperature, Fuel_Price, Markdown1, Markdown2, Markdown3, Markdown4, Markdown5, Unemployment, Size, Year, Month, Week, Type_B, Type_C] → (to predict) CPI
- X_6, y_6 Predict whether a given week is a holiday week
    - [Store, Dept, Weekly_Sales, Temperature, Fuel_Price, Unemployment, Size, Year, Type_B, Type_C] → (to predict) Is_Holiday

From there, we use the following models:

- Polynomial regression and Random Forest, SVM using sets: (regression task to predict weekly sales)
    - X, y
    - X_2, y_2
    - X_3, y_3
    - X_4, y_4
    - X_5, y_5
- Logistic regression, SVM, and Random Forest using (classifying holiday weeks)
    - X_6, y_6
- K- Means:
    - Use sales data to predict clusters of consumers (CPI and or gas price).
- PCA:
    - To attempt to distill discounting and holiday periods
    - To attempt to distill gas prices and CPI
- ANN: (to predict weekly sales given all features)
    - Using sets X, y

See the discussion of the results below. While we are sticking with the main algorithms in class, as described above and below, we are using them with different feature sets and using different models for the same goal so that we can compare differences between the models. Furthermore, as we will further explain in the methodology section, for robustness, reproducibility, and to improve the strength of our results, all of the models (described above) are implemented using 3-fold cross-validation with random search used to iterate over and determine the best parameters. From there the model is trained on the full training set and evaluated on the test set.

Recapping the milestones, the first milestone involves applying linear and polynomial regression to predict weekly sales using different sets of features. The goal is to predict weekly sales based on store, department, week, and holiday week status and use region temperature, fuel price, markdown, consumer price index (CPI), and unemployment rate. Additionally, a combined model will incorporate all these features to evaluate if including more variables improves prediction accuracy. In our second milestone, we implement logistic regression to predict whether a given week is a holiday based on sales performance. The input here is sales data, excluding obvious indicators like discounting, and the output is a binary classification (holiday or not). The interesting aspect of this problem lies in its potential to detect patterns in consumer behavior during holidays, which can inform promotional strategies and stock allocation. This will also be done using SVM and random forests. The third milestone uses K-means clustering to identify clusters of consumers based on sales data, CPI, and gas prices. The inputs are the sales and economic variables, with the output being clusters of consumers exhibiting similar behaviors. This clustering problem is interesting because identifying consumer

segments helps tailor marketing efforts, optimize pricing strategies, and improve product recommendations. Principal Component Analysis is used to reduce the dimensionality of the data and extract important features related to discounting periods, holidays, CPI, and gas prices. The input consists of high-dimensional sales and economic data, while the output is a reduced set of components that distill key information.

## 3. Experimental Evaluation

### 3.1 Methodology

Our approach to forecasting weekly sales in Walmart stores begins with data preprocessing, which was crucial for ensuring the model's accuracy and robustness. Given the dataset's structure, we began by merging the main training data with additional features provided. Missing values in the promotional markdown data (MarkDown1-5), common for dates before 2011, were handled using median imputation to avoid introducing bias while ensuring that imputed values did not distort holiday season trends. Normalization was another key preprocessing step, especially important for algorithms sensitive to feature scale, like KMeans clustering and PCA, which we used to reduce dimensionality and noise. We chose this data set as it is all real data that can be used to gain truly beneficial insight into a market.

Our first experiments involved testing Linear Regression and Random Forest models on this preprocessed dataset, with and without PCA for dimensionality reduction, allowing us to compare computational efficiency and predictive accuracy. The training and test datasets span two years of historical sales data, providing a realistic scope that reflects annual economic and seasonal shifts, making it ideal for forecasting. To visualize and analyze clustering in the data, we applied KMeans clustering and explored the Elbow Method to determine optimal cluster counts, revealing natural groupings within the data and enabling insights into patterns across stores and departments.

Continuing to the models, to be both as diligent and thorough as possible, for all of our models (discussed above) we implemented 3-fold cross-validation and random search to find (attempt) the best hyperparamates. Given our large dataset, we believe 3-fold cross-validation to be sufficient. Furthermore given we suspect there is potential overfitting and a need to do feature selection, we chose to use elastic net regularization in all models. Furthermore, as discussed in class, a random search approach for finding optimal hyperparameters is the most efficient. For all hyperparamers, we provide a wide range of possible values to test. We use 15  iterations (combinations) of hyperparameters to train the model across the folds. From there we pick the best model using MSE for regression models and precision for classification, train the full dataset, and evaluate the model on the test set (more on this in the results section). Key to our analysis will be that for the prediction of weekly sales we trained various polynomial linear regression models

random forests, and SVMs with different data containing different subsets of features. More specifically, we will compare and test both how the results differ given the model type (RF vs polynomial regression vs SVM ) and feature subset. We visualize our results using PCA to help limit our degree as well as by comparing performance across both model and performance metric as well as to the analogous ANN to predict weekly sales. We will wrap up by comparing our results to the other work. See the discussion of results to follow.

## 3.2 Results

Our results consist of two broad tasks - regression and classification - across three modeling goals.

Further details on the methodology of regression tasks:

- All regression tasks used polynomial regression, Random Forest, and SVM models. We used 3-fold cross-validation to examine 15 random combinations of hyperparameters.
    - More specifically, for the polynomial regression we used elastic net regularization and we considered an alpha value from 1 to 100,000, an L1 ratio from 0.001 to 1, and a polynomial degree from 1 to 3.
    - For the random forest, we considered 10 to 200 estimators (number of trees) and a maximum tree depth of either 5, 15, or 30 (to try to minimize overfitting).
    - For the SVM we had trouble running our code with it never converging. In doing some research, we found this is likely to be due to the n^2 to n^3 complexity of the SVM given it must check every point as a potential defining vector. So, while not ideal, we chose to take a random sample of 50k entries (from our over 400k entries) to build the SVM models. We then use an RBF kernel and we consider a C = 1/alpha between -100 and 100 and a gamma between -1,000 and 1000.
    - We recognize this is an expansive range of hyperparameters and in future work, we hope to refine this and consider more combinations. The models were chosen and evaluated using MSE given it is the go-to in machine learning modeling. However, in our results, we also report root mean squared error and mean absolute error for context and to further our discussion.
- Furthermore, we also created a neural network to predict weekly sales using all other features. These results are compared to the analogous polynomial regression, SVM, and Random forests below. For our neural network, we

experimented with a plethora of layers and several nodes with the best results arising from a model using leaky Relu activation functions and with 4 hidden layers. The layers had the following sizes: input_size, 300, 200, 100, 50, 1. Furthermore, with an intense fear of overfitting, we applied several regularization techniques including batch normalization, dropout, and early stopping. We once again used MSE as the error metric but also reported MAE and RMSE.

- Finally, as will be later discussed, the original idea of attaching test data to the dataset was morphed into its own sentiment analysis. This used a separate dataset, tokenizing and vectorizing the text data to perform a simple classification of the 1 out of 5 rankings of products on the Walmart website.

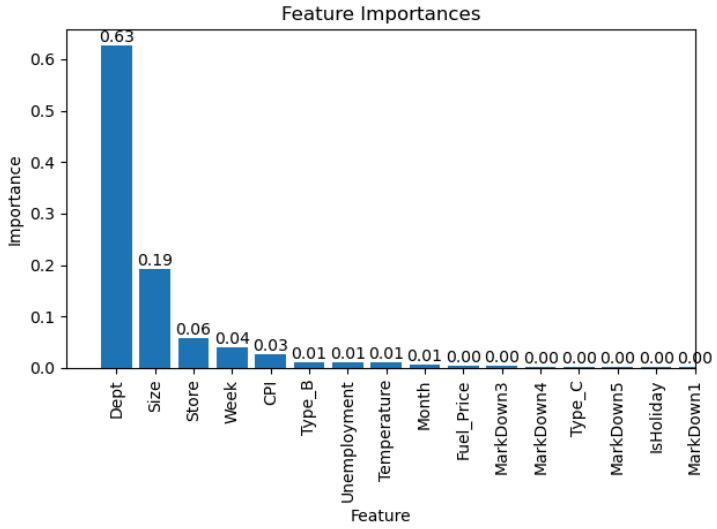
<u>Further details on classification tasks:</u>

- All classification tasks used polynomial logistic regression, Random Forest, and SVM models. We used 3-fold cross-validation to examine 15 random combinations of hyperparameters.
    - More specifically, for the polynomial logistic regression we used elastic net regularization and considered an alpha value from -1,000 to 1, an L1 ratio from 0.001 to 1, and a polynomial degree from 1 to 3.
    - For the random forest, we considered 10 to 200 estimators (number of trees) and a maximum tree depth of either 5, 15, or 30 (to try to minimize overfitting).
    - For the SVM we had trouble running our code with it never completing. In doing some research, we found this is likely to be due to the n^2 to n^3 complexity of the SVM given it must check every point as a potential defining vector. So, while not ideal, we chose to take a random sample of 50k entries (from our over 400k entries) to build the SVM models (the same subset we used in regression tasks). We then use a polynomial kernel and consider a C = 1/alpha between -100 and 100 and a degree between 1 and 3. We recognize this is an expansive range of hyperparameters and in future work, we hope to refine this and consider more combinations.
    - The models were chosen and evaluated using precision (sk learns average_precsion metric to be exact) as we have an unbalanced dataset. More specifically, only 7.03% of our entries (rows) correspond to a holiday week. Furthermore, we realized that, in context, Walmart would rather say a holiday week is not a holiday week, (so they would't feel the need to lower pricing or run sales and sell things for a discount when they don't need them), which is why we chose to use precision instead of recall. Thus, while the optimal model is chosen by looking at precision and that is the lens we interpret the results, we also report accuracy, recall, and F1 scores.

We next discuss our results broken down by goal.

Goal 1: Predict weekly sales given various sets of features.

- We first consider feature set 1 as described above (denoted X and y respectively). This is the most complete set of predictors. We proceed by breaking the results down by model and then discussing them in context.
    - Polynomial regression: The best model of our random search had a degree of 3, an elastic net ratio of 0.112, and an alpha of 1.0. This results in a training MSE of 419907757, testing MSE of 423549443, training RMSE of 20491, testing RMSE of 20580, training MAE of 13337, and testing MAE of 13315.6.
    - For the random forest, the best model of our random search used 180 trees with a max depth of 30. This results in a training MSE of 1777440, testing MSE of 13381859, training RMSE of 1333, testing RMSE of 3658, training MAE of 531, and testing MAE of 1440. Furthermore, per Professor Wshah's feedback, from this random forest model, we also extracted feature importance (see figure below). We see, interestingly that the department and size of the store alone account for over 80% of the impurity reduction due to department and size. This is interesting and we will compare it to the other random forest we generate in our final discussion.



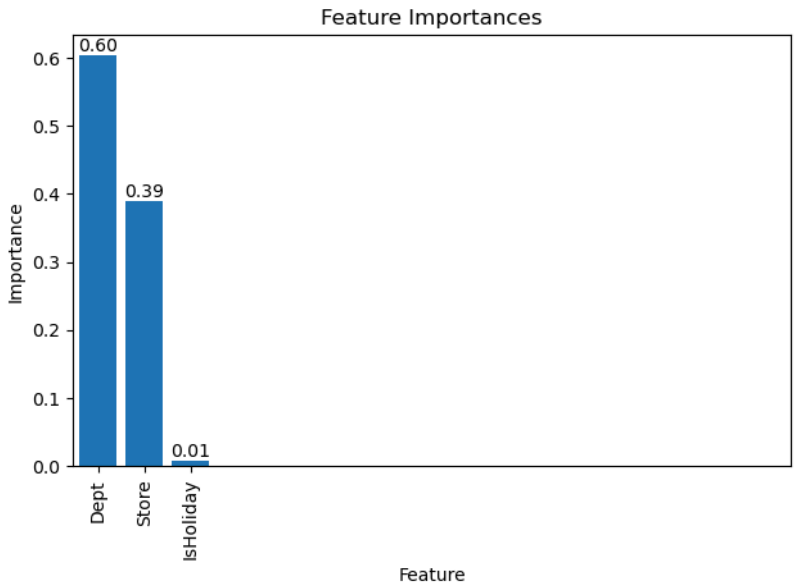

-

- For the SVM, our best hyperparameters were a gamma of 0.1 and a C value of 35.94. This results in a training MSE of 540913405.1, testing

MSE of 544178936.1, training RMSE of 23257.54, testing RMSE of 544178936.2, training MAE of 12515.37, and testing MAE of 12594.92

- Discussion: While the model was chosen by minimizing MSE as this is the industry default, MSE is relatively hard to interpret given the square root. Thus, we turn to examining MAE as it has the most practical interpretation. MAE represents the average absolute difference between actual and predicted weekly sales. In looking at our 3 models, we see that the random forest model outperforms the polynomial regression by nearly a factor of 10 with the MAE for the random forest being 1440 with the polynomial regression and SVM having MAE of 13316 and 12595 respectively. So, given that the range of our full dataset is $426,230.25 with a mean of $15,989.4, our best model, the random forest, having a MAE 1440 means its average absolute error of 0.337% of the range and 9% of the mean which is to say the random forest does a very good job predicting weekly sales while the other models.

- We next consider feature set 2 (denoted X_2 and y_2 respectively). This is what we believe to be the minimal subset of predictors.
    - Polynomial regression: The best model of our random search had a degree of 3, an elastic net ratio of 0.112, and an alpha of 1.0. These are the same hyperparameters that were optimal for feature set 1. This results in a training MSE of 483707189.65, testing MSE of 490149906.79, training RMSE of 21993.34, testing RMSE of 22139.33, training MAE of 14708.78, and testing MAE of 14724.06.
    - For the random forest, the best model of our random search used 60 trees with a max depth of 30. This is a simpler optimal model than the optimal model for feature set 1. This results in a training MSE of 45348117.52, testing MSE of 51764304.42, training RMSE of 6734.101, testing RMSE of 7194.741, training MAE of 2605.05, and testing MAE of 2699.11. In looking at feature importance, we again see that the department accounts for 60% of the importance (reduction in impurity), as it did with feature set 1, while the importance of store has more than doubled to 39% compared to the 19% with feature set 1.




-

- For the SVM, our best hyperparameters were a gamma of 0.1 and a C value of 35.94. These are the same hyperparameters that were optimal for feature set 1 This results in a training MSE of 541834221.80, testing MSE of 541349690.40, training RMSE of 23277.332, testing RMSE of 23266.92, training MAE of 12537.70, and testing MAE of 12519.240.
- Discussion: Per the above discussion about the interpretability benefits of MAE, we once again focus on MAE (see above for all other scoring metrics). As we saw in the forest feature set, the random forest had the smallest (and thus best) MAE of 2699 compared to the polynomial regression and SVM having MAE of 14708 and 12519 respectively. That being said, feature set 2's (what we are discussing now) MAE of 2699 is nearly double (twice as bad) as the random forest using the full feature set (1440). 2699 is now 0.63% of the range and 16.8% of the mean - again showing the decrease in performance by a factor of 2 - but still quite good performance overall. However, feature set 2's MAE for the polynomial regression of 14708 is quite close to feature set 1's MAE of 13316. The same is also true for the SVM with feature set 2 having an MAE Of 12519 which is again quite close to feature set 1's MAE of 12595. Examining feature importance, we once again see that the department accounts for 60% of the importance (reduction in impurity), as it did with feature set 1, while the importance of the store has more than doubled to 39% compared to the 19% with feature set 1. That is to say, that given we are using far few predictors in feature set 2, the store seems to be taking on that load as a predictor.

- Next using feature set 3: (going forward, results will be displayed in tabular form and then discussed)

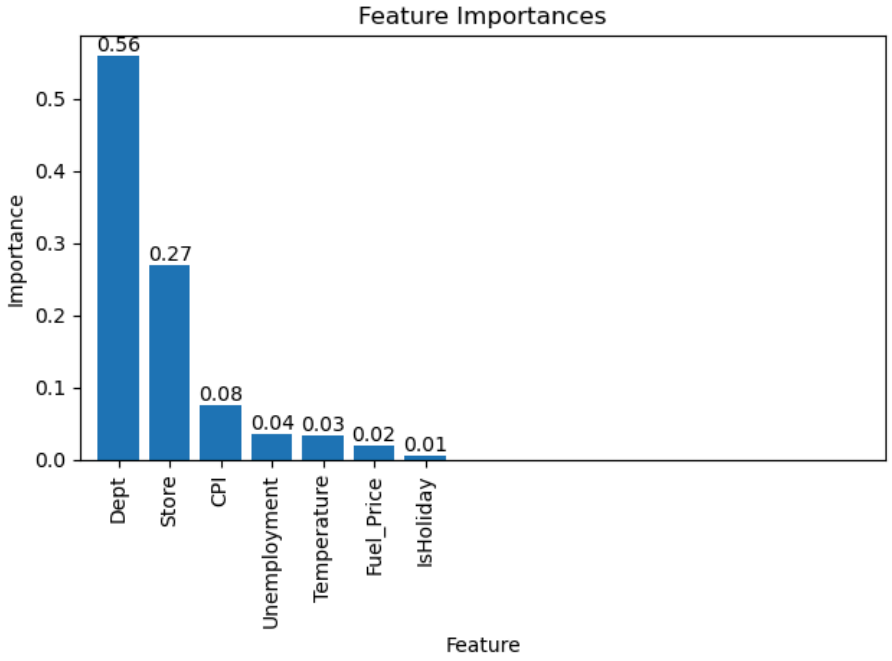
- Polynomial regression: The best model of our random search had a degree of 3, an elastic net ratio of 0.112, and an alpha of 1.0. These are the same hyperparameters that were optimal for feature sets 1 and 2. This results in:

| root_mse_train | root_mse_test | mse_train | mse_test | mae_train | mae_test |
|---|---|---|---|---|---|
| 21778.5 | 21926.6 | 474302011.5 | 480777086.0 | 14481.9 | 14500.2 |

- For the random forest, the best model of our random search used 180 trees with a max depth of 30. This is the same as the optimal model for feature set 1 but has more trees than the optimal model for feature set 2. See results below:

| root_mse_train | root_mse_test | mse_train | mse_test | mae_train | mae_test |
|---|---|---|---|---|---|
| 2015.9 | 5607.2 | 4063731.2 | 31440721.3 | 705.3 | 1919.2 |

- In looking at feature importance, we once again see that the department accounts for 57% of the importance (reduction in impurity) which is very similar to what it did with feature sets 1 and 2. Interestingly, the importance of the store being 0.27 is between its respective importance using feature sets 1 and 2. CPI is also comparatively over twice as important (0.8) as it was using feature set 1.




-

- For the SVM, our best hyperparameters were a gamma of 0.1 and a C value of 35.94. These are the same hyperparameters that were optimal for feature sets 1 and 2. This results in:

| root_mse_train | root_mse_test | mse_train | mse_test | mae_train | mae_test |
|---|---|---|---|---|---|
| 23468.0 | 23483.7 | 550746565.0 | 551485304.0 | 12668.5 | 12664.9 |

- Discussion: We focus our discussion on the test MAE of the models due to its ease of interpretation. Once again the random forest model outperforms the polynomial regression and SVM having (the random forest) a test MAE of 1919.2 compared to the polynomial regression' of 14481.9 and SVM's 12668.5 the latter two of which are between 6 and 8 times larger (worse). Furthermore, feature set 3 (what we are discussing now) MAE for its random forest model splits the difference between the test MAE using feature set 1 (1439.8) and feature set 2 (2699.1). The same (being between) holds in the added context of the range and mean as well. Regarding feature importance, department is once again the most important feature (just under 60%) with store once again being the second most important feature - both of which were the case using feature sets 1 and 2.

- Lastly looking at feature set 4 (more features than set 3, but not quite all the features)
  - Polynomial regression: The best model of our random search had a degree of 3, an elastic net ratio of 0.112, and an alpha of 1.0. These are the same hyperparameters that were optimal for feature sets 1,2 and 3. This results in:

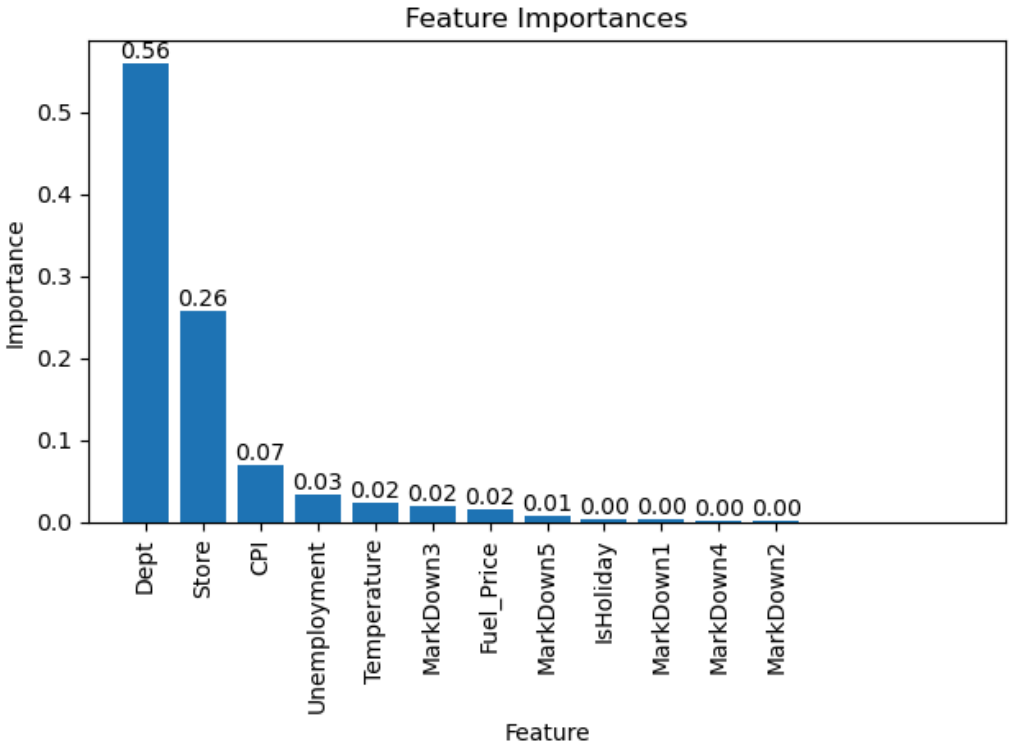
| root_mse_train | root_mse_test | mse_train | mse_test | mae_train | mae_test |
|---|---|---|---|---|---|
| 21680.7 | 21825.1 | 470053077.8 | 476336386.3 | 14401.6 | 14413.9 |

  - For the random forest, the best model of our random search used 80 trees with a max depth of 30. While depth has remained the same across all 4 random forests, this random forest (feature set 4) used 80 trees while feature sets 1 and 3 used 180 and feature set 2 used 60.
  - See results below:

| root_mse_train | root_mse_test | mse_train | mse_test | mae_train | mae_test |
|---|---|---|---|---|---|
| 1963.4 | 5205.7 | 3855080.2 | 27099581.6 | 710.8 | 1897.4 |

  - In looking at feature importance, we once again see that the department accounts for nearly 60% of the reduction of impurity and thus importance (56%) which is very similar to what it did with feature sets 1,2 and 3. Interestingly, the importance of store being 0.26 is nearly identical to its importance using feature set 3 (27%) - both of which are between its

respective importance using feature sets 1 and 2.



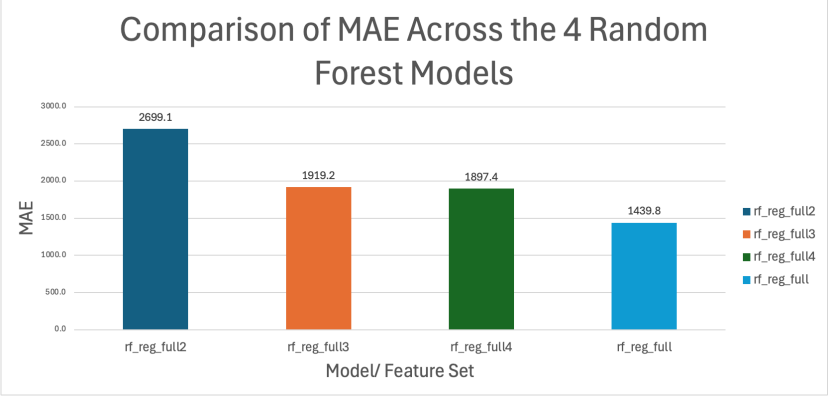

- For the SVM, our best hyperparameters were a gamma of 0.1 and a C value of 35.94. These are the same hyperparameters that were optimal for feature sets 1,2, and 3. This results in:

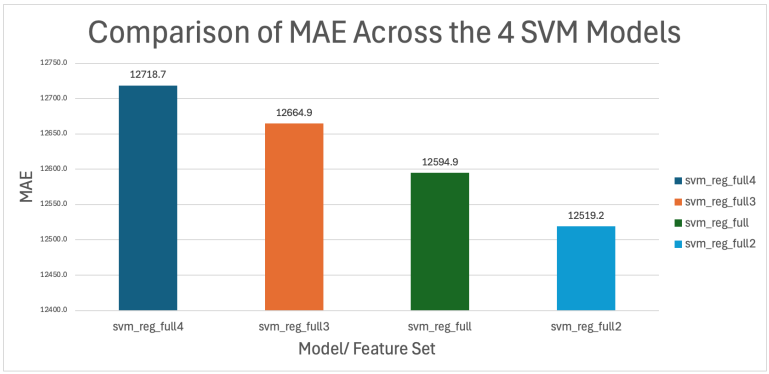
| root_mse_train | root_mse_test | mse_train | mse_test | mae_train | mae_test |
|---|---|---|---|---|---|
| 23530.8 | 23533.2 | 553698543.9 | 553813539.5 | 12718.1 | 12718.7 |

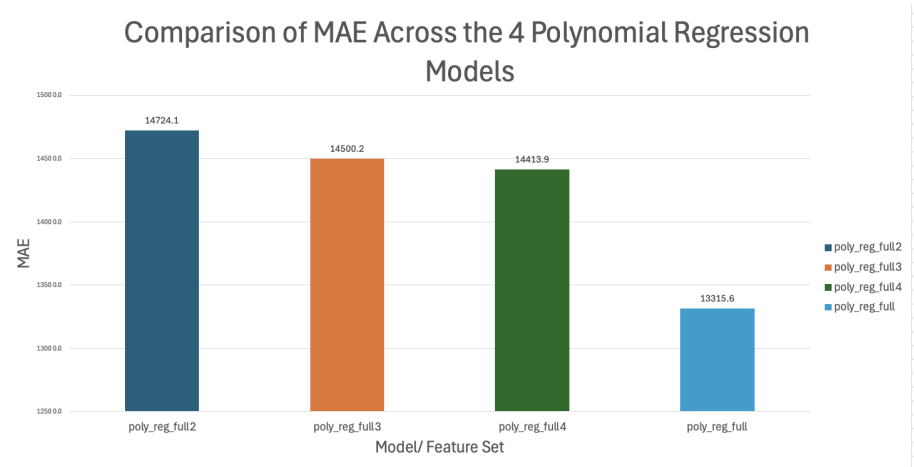
- Discussion: We continue our focus on testing the MAE of the models due to its ease of interpretation. Once again the random forest model outperforms the polynomial regression and SVM with the random forest having a test MAE of 1897.4 while the polynomial regression and SVM have test MAEs of 14413.9 and 12718.7 respectively. Focusing on the random forest (the best model), given it used more features than feature set 3's random forest but less than feature set 1's random forest, we hypothesize its test MAE to be in between the two; and this is exactly what we see. Feature set 4's (what we are talking about) test MAE of 1897.4 is slightly less than feature set 3's of 1919.2 but larger than feature set 1's of 1439.8. The same (being between) holds in the added context of the range and mean as well and thus feature set 4's random forest test MAE of 1897.4 is 0.445% of the range and 11.86 of the mean. Regarding feature importance, the department is once again the most important feature (just under 60%... 56% to be exact) with the store once again being

the second most important feature - both of which were the case using feature sets 1,2, and 3 as well.

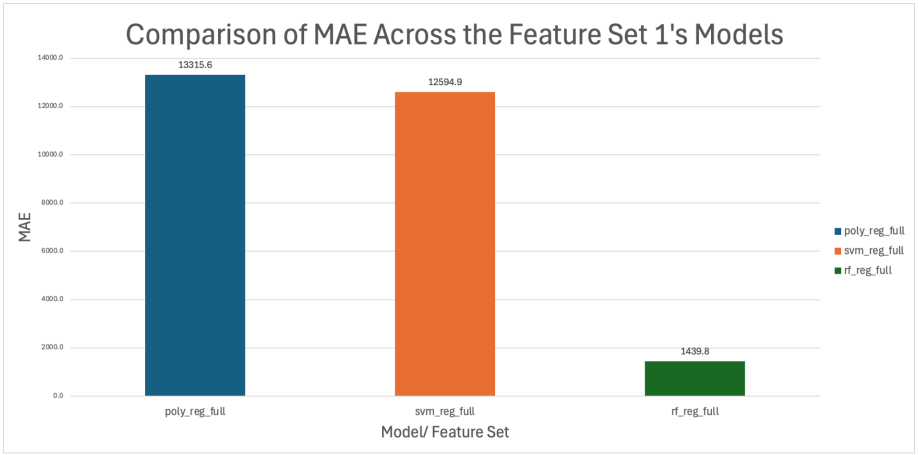
- Overall Discussion of Goal 1: In parsing through the above results, we see that the random forest model is the best modeling approach compared to SVM and polynomial regression. Across all 4 random forest models, test MAE ranged between just over 1400 and just under 2700. This means that the average absolute error using our best models (the random forests) was only between 0.328% and 7% of the range; and between 8.7% and 18.7% of the mean. This is quite strong, especially our best model using all features. We also saw that the more features we included in the model, the lower the error was. When looking at the hyperparameters, both all 4 polynomial regression models and all 4 SVM models came to use the same hyperparameters. Contrastingly, All random forest models grew to the maximum depth allowed however there was variation in the number of decision trees used. We note that given that we had to take the sample of 50k rows to get the SVM to run, we do not want to firmly say whether the SVM outperforms the polynomial regression. However, both most definitely are worse at predicting weekly sales than using a random forest model. Furthermore, the more features used in the model, the resulting error. This being said, in looking at feature importance department and store (two features) accounted for over 80% of the reduction of impurity (importance) in all models.
- See below for a visual comparison of test MAE across all feature sets grouped by type (SVM, RF vs polynomial regression)
-



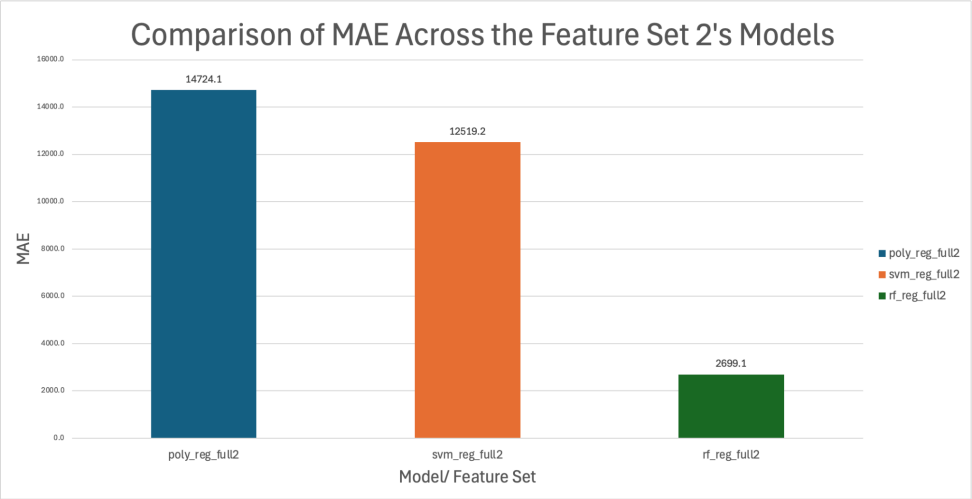
- 

- 


- We also compare test MAE across feature set 1 (most amount predictors) and feature set 2 (least amount of predictors)

- 

-




**Experimentations with Neural Networks**

The neural network we developed implemented the idea that deeper is better as more layers tend to reveal linearly separable data. However, in our experiments, it was shown that the neural network always produced worse results than the random forest. Despite significant efforts to fine-tune and optimize the architecture, the results were not as successful as anticipated. Our objective was to leverage the power of deep learning to capture complex patterns and relationships within the dataset, potentially outperforming more traditional models such as decision trees and ensemble methods. The NN failed to demonstrate a marked improvement over simpler algorithms and struggled with overfitting despite our best efforts to mitigate it. Here, we present the metrics and detail the measures we applied to combat overfitting. Given the persistent overfitting we encountered, we applied a range of strategies aimed at improving the generalizability of our model:

1. **Early Stopping:** We used early stopping with a patience parameter to monitor validation loss and halt training when no improvement was detected for a specified number of epochs. This approach was intended to prevent the model from continuing to learn noise in the training data.
2. **Dropout Layers:** Dropout was incorporated into the model architecture to randomly deactivate a fraction of neurons during training, which helps prevent overfitting by ensuring that the model does not become overly reliant on specific neurons.
3. **Batch Normalization:** To stabilize and accelerate training, we added batch normalization layers. This technique normalizes the input of each layer, which can lead to faster convergence and improved model performance.

Our extensive experimentation with the neural network involved trying out various architectures, modifying hyperparameters, and testing different combinations of regularization techniques. We even explored alternative approaches like adjusting the learning rate schedule and optimizing the initialization strategy for weights. Ultimately, after dedicating a full week (just for the NN)  of iterative testing and optimization—running models, assessing them, and fine-tuning hyperparameters—the NN only reached the performance metrics outlined above. See the results and further discussion below.


The results from the Neural Network are as follows:
Training Metrics:

- MSE: 176391696.0000
- MAE: 8131.9478
- RMSE: 13281.2539

Test Metrics:

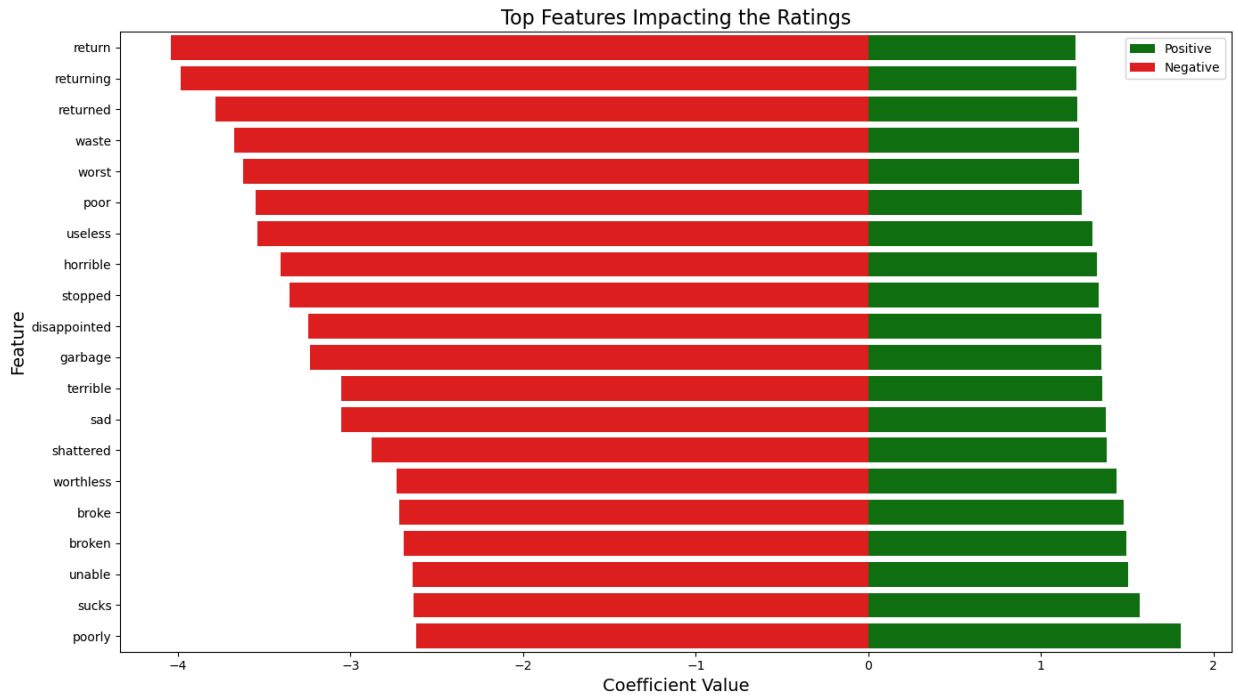
- MSE: 183896512.0000
- MAE: 8212.8057
- RMSE: 13560.8447

**Discussion of NN Results:** Despite these efforts, none of these strategies resulted in significant performance gains. The model's performance on the test set continued to lag

behind expectations, and we were unable to achieve a generalizable solution that surpassed the results of simpler models like random forests. We see that the MAE of the test set of 8,212.8 means it is 1.9% of the range and 51.3% of the mean. Comparing this to the random forest models (specifically those using the full feature set which resulted in the best performance) we see the NN's percent of range while solid, is over five times larger than the random forest percent of range (0.34%). Similarly, the NN's % of the mean is also over five timers larger than the random forest's percentage of the mean (9%). The NN performs in between, but closer to, the performance of the SVM (again) which has an MAE of 2.9% of range and 79% of mean - which as we have already discussed is less than ideal. Furthermore, we see looking at the above graph of training vs testing error (MSE) that the validation loss is lower than the training loss which is perhaps a little bit surprising. However, we believe this is likely due to the dropout normalization making a slighter weaker training model to then combine for the validation and tests set effectively creating an ensemble and thus improving the results. When looking at the final test and train results where we see test error is higher than the train error as we would expect because despite the robustness that dropout provides unseen data is still unseen data. Thus we expect the results to be slightly worse for the test set. Furthermore, from the graph, as the error falls and plateaus and there is no gigantic gap between the two (nor is there an overly large gap for the final test and train error metrics) we suspect overfitting is not a problem. That being said, as our results are not nearly as good as our random forest models we suspect that there is underfitting.

**Sentiment Analysis**

Aiming to extend beyond the dataset, we had aspired to gain additional insight by scraping reviews from Walmarts in the dataset and adding them to the models. However, our attempts at this fell apart at multiple steps requiring us to pivot to gain any useful information. Our initial plan used the information from the original data to identify which Walmart stores were used in the study and to find online store information about them. In our attempts, we learned that Walmart's online marketplace is not separated by store when it comes to reviews and ratings. So this left us with no real way to connect rating data with the sales data. So to preserve energy we chose to utilize an already existing dataset on Walmart product reviews [6]. The features we used for this analysis were product titles and text reviews to predict the rating of that product. This was accomplished using TfidfVectorizer to convert the raw text data and a ridge regressor model to predict the rating. The results of this ridge model are as follows: A Mean Squared Error of 1.0627, a Normilized Mean Absolute Error of 0.7663, and an $R^2$ Score of 0.4357. These error values are very small compared to the results of other models, but they may be incomparable as they are predicting a very different dataset.




The table above shows the feature importance of the model. It shows that there are words that are more impactful on the model. Words like “return” and “waste” are clearly correlated with negative ratings and reviews that contain those works are more likely to be negative.

Goal 2: Predict CPI using all other features.

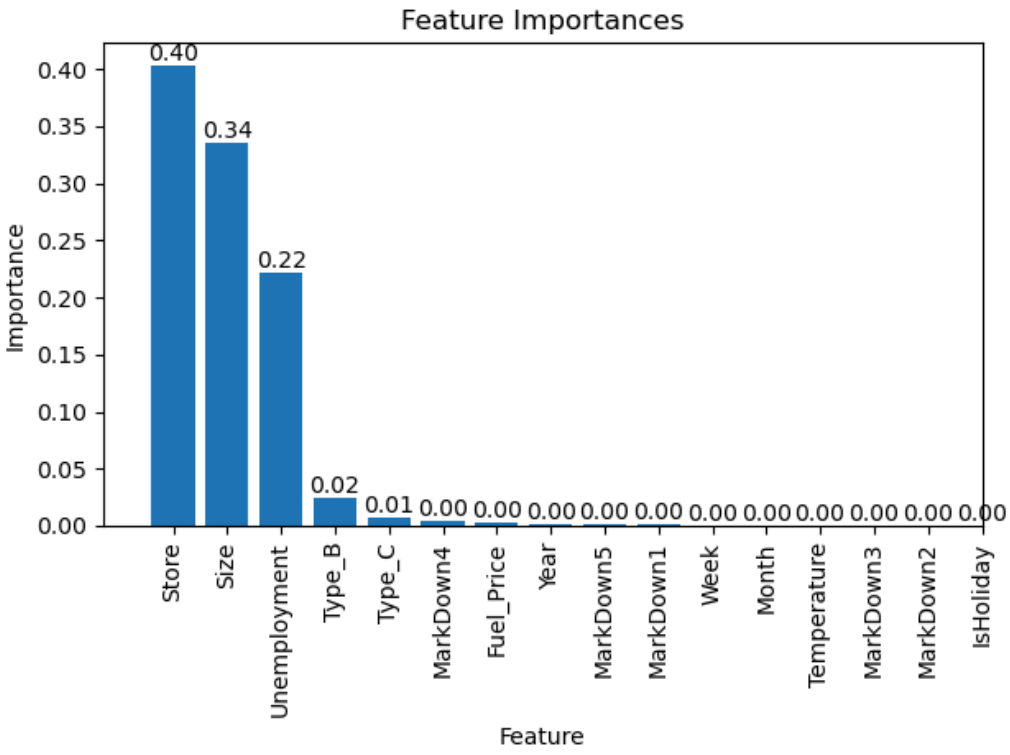
- As described above, we also wanted to go the other way. That is to say, what can all of these Walmart sales metrics tell us about overall consumer pricing levels. Thus, here we predict CPI using all other features (feature set 5).
- We use the same hyperparameter space and random search as used in goal 2, Polynomial Regression: The best model of our random search had a degree of 3, an elastic net ratio of 0.112, and an alpha of 1.0. This results in:

| root_mse_train | root_mse_test | mse_train | mse_test | mae_train | mae_test |
|---|---|---|---|---|---|
| 31.8 | 31.9 | 1010.8 | 1019.1 | 28.6 | 28.7 |

- For the random forest, the best model of our random search used 80 trees with a max depth of 30. This resulted in:

| root_mse_train | root_mse_test | mse_train | mse_test | mae_train | mae_test |
|---|---|---|---|---|---|
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

- We also calculated feature importance (see below with discussion to follow)




-

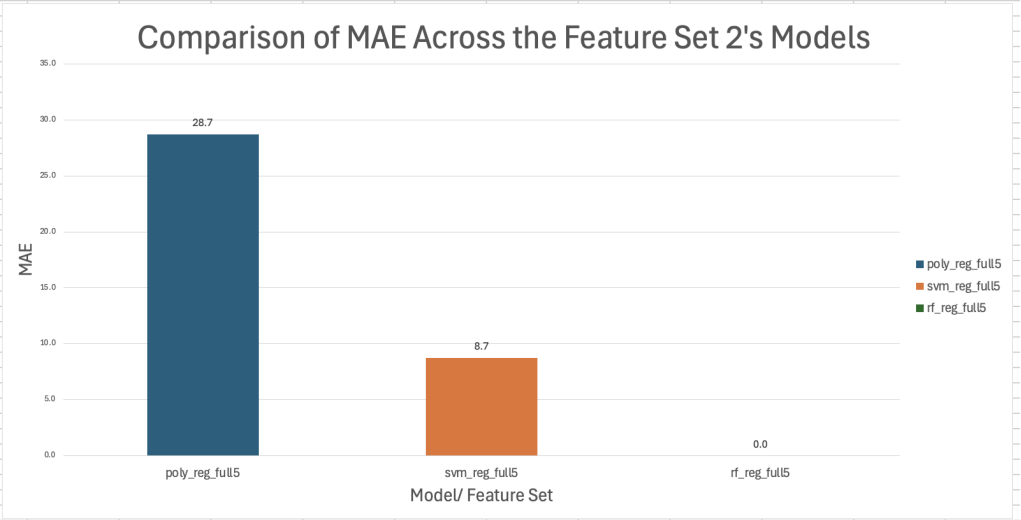
- For the SVM, our best hyperparameters were a gamma of 0.1 and a C value of 35.94. This results in:

| root_mse_train | root_mse_test | mse_train | mse_test | mae_train | mae_test |
|---|---|---|---|---|---|
| 14.6 | 15.3 | 214.0 | 235.1 | 8.1 | 8.7 |

- Discussion of Goal 2: In examining these results, we once again see that the random forest is the best model. Across all error metrics, the random forest outperforms the polynomial regression and SVM. Like in goal 1, we note that due to runtime inefficiencies associated with SVM (as further described above), we used the same random sample of 50 thousand samples as we did in all SVM calculations for goal 1. That being said across all error metrics, the SVM outperforms the polynomial regression and is clearly the second-best model. CPI for the test set has a mean of 171.23 and a range of 101.16. Thus, as we did in goal 1, given the interpretability, we analyze through the lens of MAE. Interestingly we see that the random forest has an error of 0 across all metrics across both training and test sets. The fact that the training set also results in an error of zero is interesting as it hints at overfitting not being a problem. So, while surprising, our random forest gives us a perfect performance. Note that these results are rounded, technically there is a non-zero error out to tenth (or more) digit for the random forest. We also look at feature importance and see that store

is the most important feature with size and unemployment the second and third most important features respectively. These three, combined, reduce purity by 96%. This makes sense as CPI is tracked locally in the data, thus the model can learn, given the location (store), what the CPI is. Size is likely correlated with location and demand and is thus helpful in narrowing down CPI. Unemployment is interesting as well as high CPI is often associated with inflation which can be associated with high unemployment. While not exact, unemployment and CPI are clearly related as many economists will explain.

- See below for a graph comparing MAE across models:
-




Goal 3: Classifying a given week as a holiday week using all other features (except markdowns). See algorithm definition for a more detailed explanation of the feature sets.

- While the goal of our analysis is ultimately sales forecasting, we did not want to be just one trick ponies and thus we wanted to see if we could predict whether a given week was a holiday week given the other features (see data and method sections for more details). Using our aforementioned logistic regression, Random Forest, and SVM models we achieved the following results.
- Logistic Regression: The best hyperparameters were a degree 3, an L1 ratio of 0.223, and a C value of 1.29. This led to the following results:

| f1_training | f1_testing | accuracy_training | accuracy_testing | precision_training | precision_testing | recall_training | recall_testing |
|---|---|---|---|---|---|---|---|
| 0.00228 | 0.00267 | 0.92984 | 0.92903 | 0.65854 | 0.57143 | 0.00114 | 0.00134 |

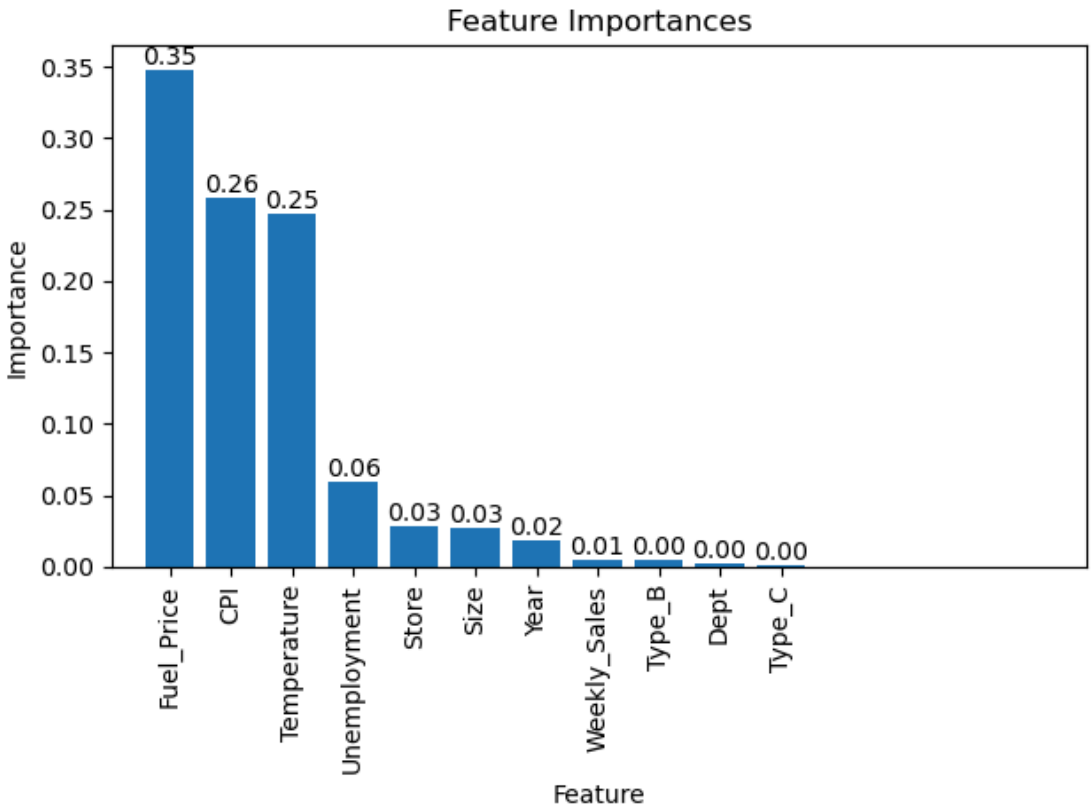
- SVM: The best hyperparameters were a degree of 3 and a C value of 0.028. This led to the following results:

| f1_training | f1_testing | accuracy_training | accuracy_testing | precision_training | precision_testing | recall_training | recall_testing |
|---|---|---|---|---|---|---|---|
| 0.00000 | 0.00000 | 0.93118 | 0.92470 | 0.00000 | 0.00000 | 0.00000 | 0.00000 |

- Random Forest: The best hyperparameters were 60 trees and a max depth of 30. This led to the following results:

| f1_training | f1_testing | accuracy_training | accuracy_testing | precision_training | precision_testing | recall_training | recall_testing |
|---|---|---|---|---|---|---|---|
| 1.00000 | 1.00000 | 1.00000 | 1.00000 | 1.00000 | 1.00000 | 1.00000 | 1.00000 |

- From the random forest, we also extracted feature importance (see graph below). We see that fuel price and CPI are the two most important features and reduce impurity by over 50%. Both fuel price and CPI are what we have been referring to as consumer metrics and it is interesting to see them correlated with holiday weeks. With our limited experience with economics, we suspect that fuel prices and CPI improve during the holidays when people are spending and traveling. Furthermore, temperature makes sense as most holiday shopping periods are in the colder months (trivially) so we are actually surprised that temperature is not of more importance.

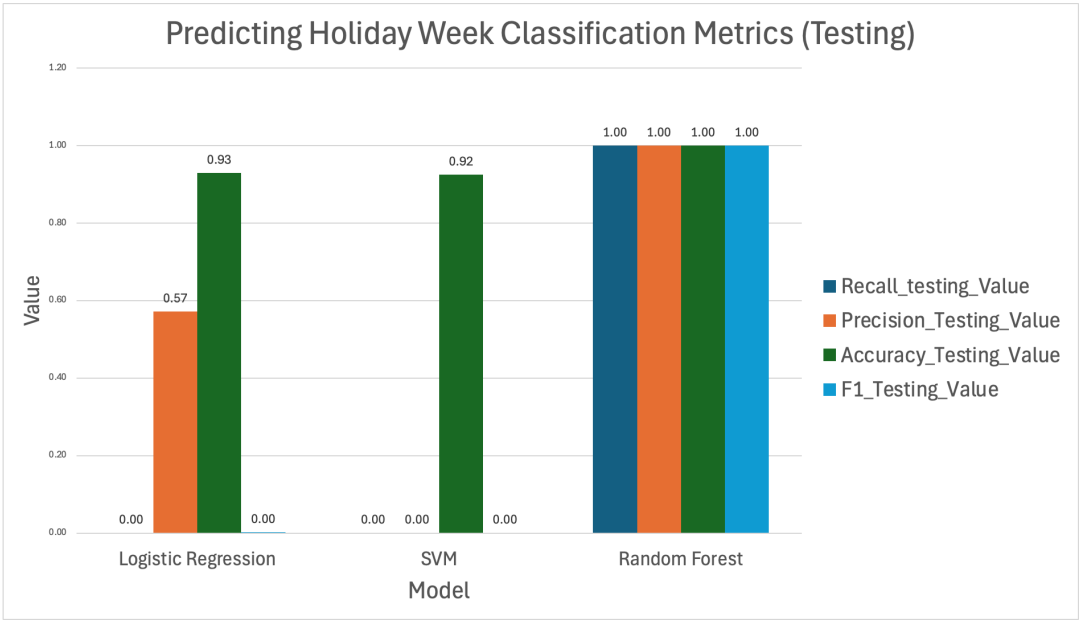



-

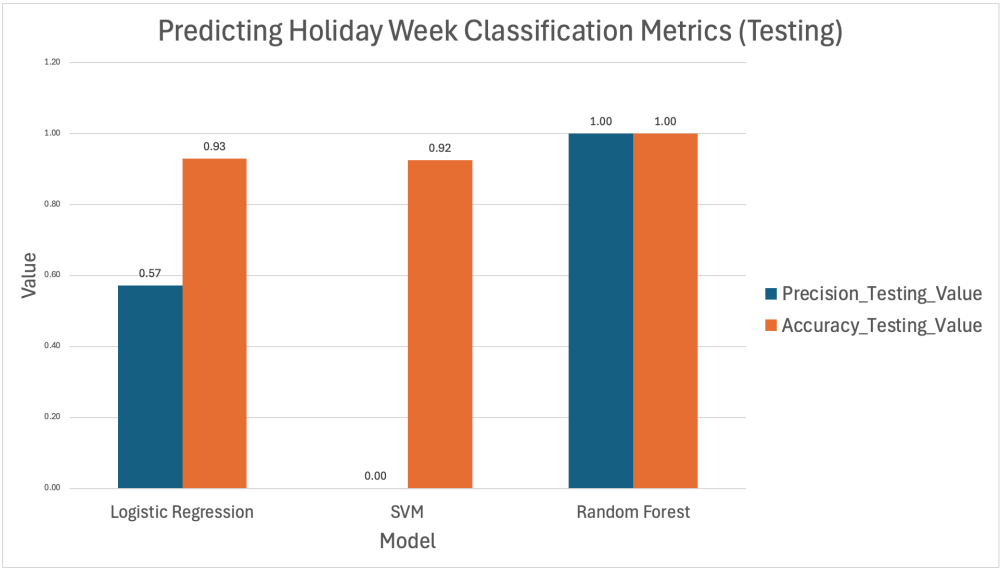
- Discussion: As only 7.03% of our entries (rows) correspond to a holiday week we have an unbalanced dataset and thus accuracy and recall are not preferred. Furthermore, as Walmart would rather minimize running sales (to the extent they still sell things) we prefer to say a holiday week is in fact not a holiday week, and thus we focus our analysis through the lens of precision. On the topic of unbalanced datasets, as only 7% of of weeks are holiday weeks, (1-0.7)% =93% of the time we are correct we see a week is a holiday week and thus we would expect accuracy to be right around 93%. This is exactly what we see for the logistic regression and SVM models. Interestingly, the random forest had perfect accuracy on both the testing and training sets. On the other hand, the logistic regression and SVM struggle with the proportion of true holiday weeks (precision) with the SVM interestingly having a precision of 0. Additionally, both the SVM and logistic regression have a recall of 0 meaning they can't predict any of the holiday weeks (this is of course problematic). Furthermore, the SVM both do incredibly poorly (0) for the F1 score. This makes sense given the severe class imbalance and thus high accuracy and recall. Thus overall, the logistic regression and SVM models are great at telling us when a week is not a holiday week but very very poor at telling us when a week is in fact a holiday week. On the completely other hand, across all measures the random forest model is perfect. This is again quite surprising but the fact that the testing scores were also perfect makes us less concerned. As we have seen through this entire project, random forests are incredibly powerful and work well with our mixture of categorical

data (easy and clear splits) and geographical spread, narrowing down the answer very quickly and accurately. This is all to say, that in our context of not selling items for less than we need to, the random forest model is clearly the best option. And, its ability to easily generate feature importance is another big advantage. In doing so (discussed above) we can see that our three most important features (fuel price, CPI, and temperature), reduce impurity and lead us to the answer by 86%.

- See the chart below looking at scoring metrics by model:




- Same as above with just precision and accuracy

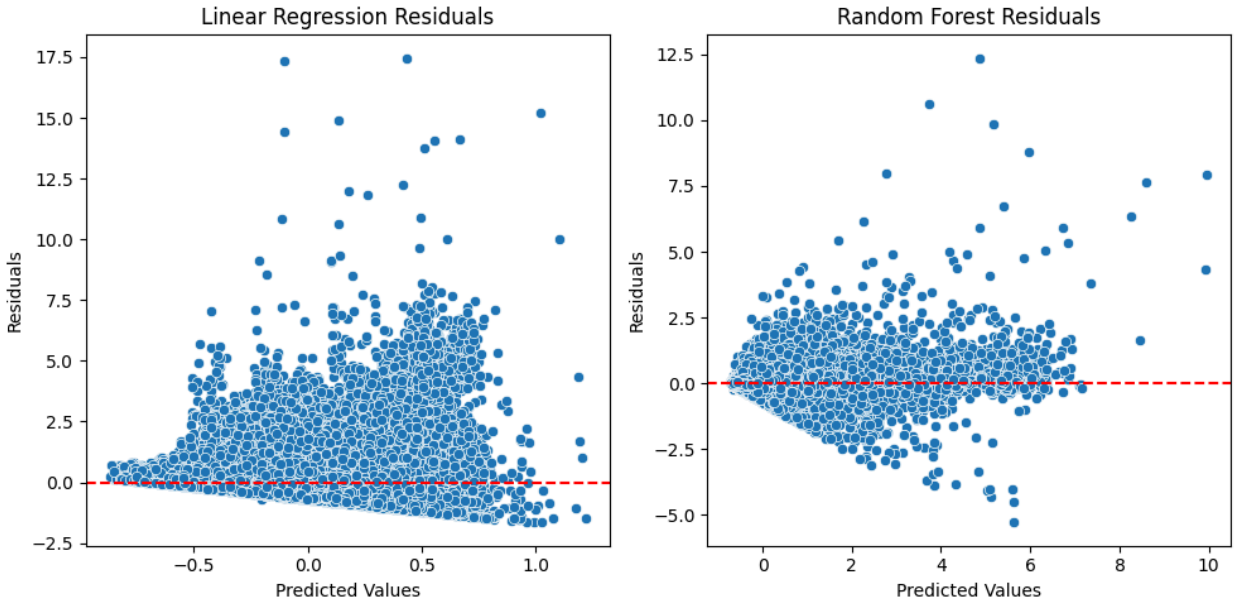



-

Additional Results (aimed at Replicating other works)

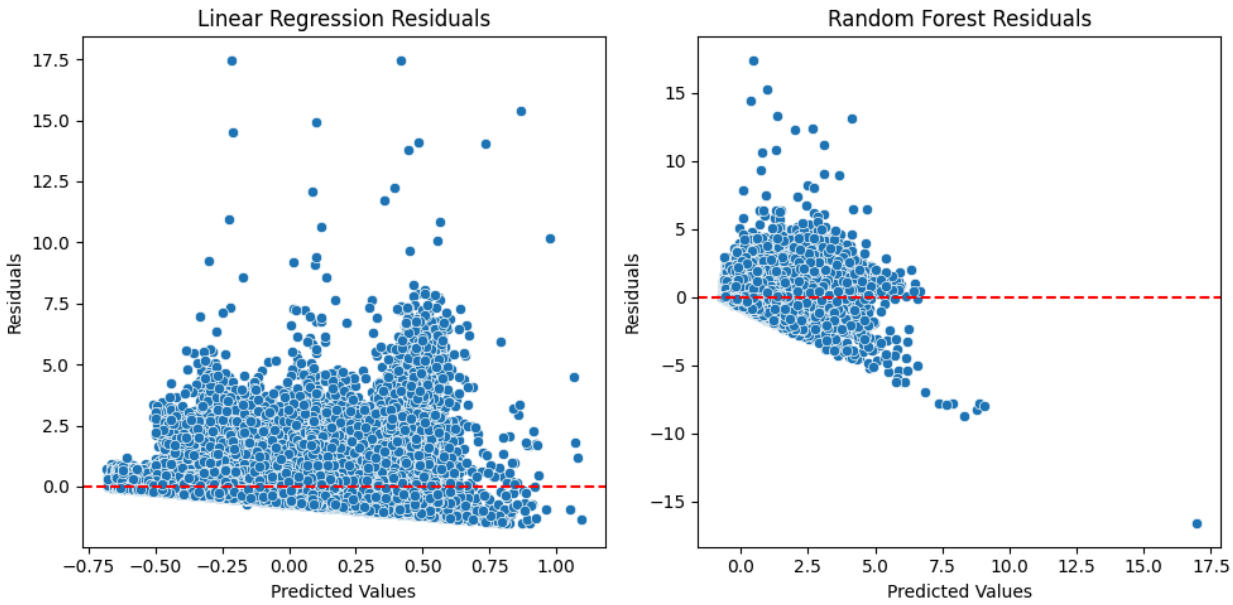
For this analysis, we used Store, Dept, Markdowns, CPI, Type, Unemployment, Weekly_Sales, Temperature, Fuel_Price, Year, and IsHoliday to forecast the department-wide weekly sales. We tested two main models: Linear Regression and Random Forest Regressor, with and without dimensionality reduction via PCA.

- For our initial run of the Linear Regression model without PCA, we evaluated its performance using Mean Absolute Error (MAE) and Weighted Mean Absolute Error (WMAE), where holiday weeks were given a weight of five as is used in the related works. Results (see next section for discussion):
  - Training MAE = 0.1751
  - Training WMAE = 0.1958

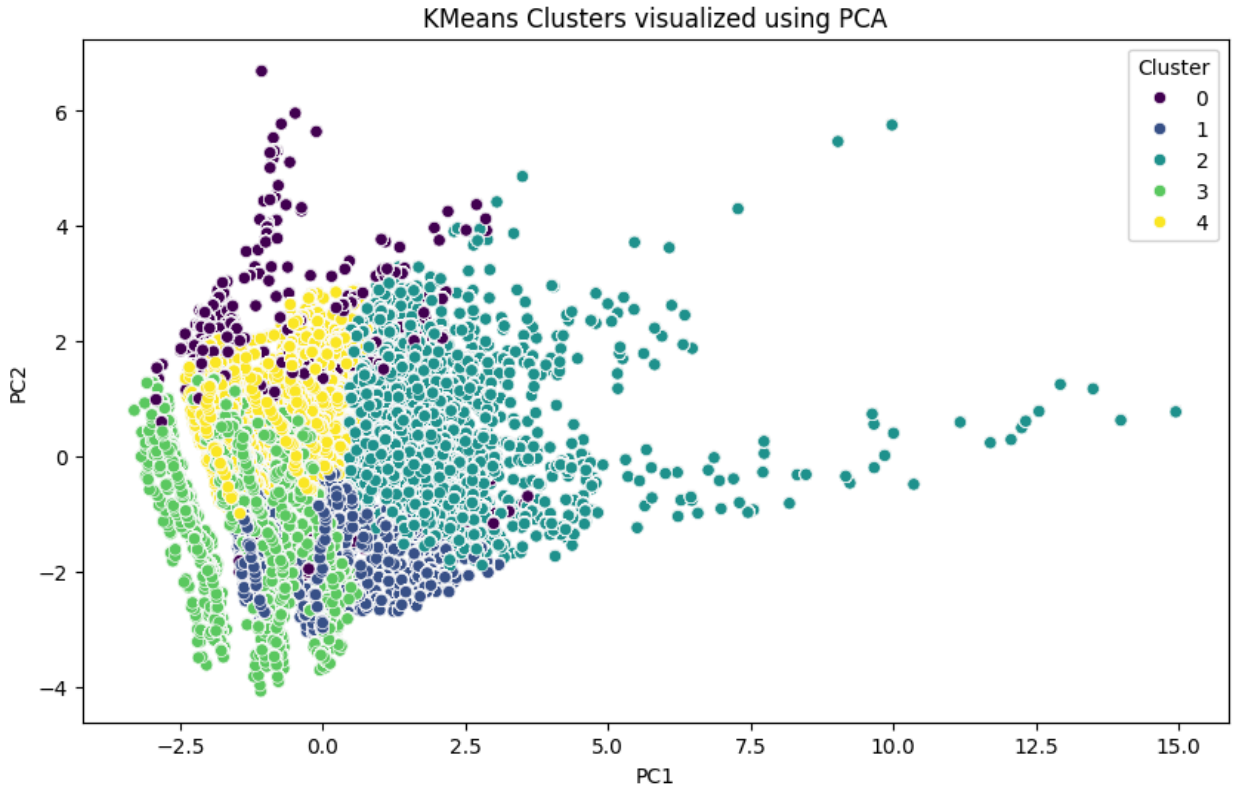



- For our initial run of the Linear Regression model with PCA, we again used MAE and WMAE. However, the model's performance declined due to information loss. Results (see next section for discussion):
  - Training MAE = 0.6438
  - Training WMAE = 0.6438

- For our initial run of the Random Forest Regressor with PCA, the model maintained better accuracy, demonstrating resilience to feature reduction. Results (see next section for discussion):
  - Training MAE = 0.2740




- Additionally, using the PCA, we performed KMeans clustering to investigate the separability of the data. The following image is the 5 clusters found with the two most influential components found from PCA:




Within many of these visualizations, a sort of trend can be seen. There seem to be "stems" that exist, seen most evidently in the random forest visualization. This can suggest the existence of correlation in the dataset as the residuals are not obviously

random. This is corrected in our models through various means such as manual and automatic feature selection and PCA, transforming the data.

### 3.3 Complete Results Discussion

Our investigations into the sales of Walmart stores have yielded great insight into the factors that result in more spending by US citizens living around the locations within the dataset. Our many-scenario approach was all aimed at our primary goal of forecasting weekly sales. Through this task, we explored the performance of multiple models, including polynomial regression, random forests, support vector machines, and neural networks of which the random forest models emerged as the strongest performers. This indicates its ability to capture complex relationships in the data effectively and to be able to behave well in smaller datasets by creating rules for itself. While polynomial regression also showed decent performance, it was clear that the model's simplicity limited its capacity to fully represent the underlying sales dynamics. Another critical area of our analysis focused on identifying holiday weeks, a binary classification task aimed at understanding shopping behaviors during special occasions. These results, while modest, revealed potential challenges such as class imbalance, which likely influenced model performance. The neural network (NN) models presented the greatest challenge in our analysis. Despite extensive tuning and the application of advanced regularization techniques like dropout, early stopping, and batch normalization, the NN's performance plateaued. While slightly outperforming polynomial regression, it fell short of random forests, underscoring the importance of dataset richness and feature diversity in training deep learning models. This finding suggests that while neural networks hold potential, they require a robust and well-preprocessed dataset to achieve their full capabilities. Overall, the project has underscored the importance of selecting both appropriate metrics for measuring performance and appropriate models based on the dataset and the specific tasks at hand. More complex models like neural networks highlight the need for feature engineering and extensive data to unlock their advantages. Beyond model performance, this project has provided actionable insights into consumer behavior. The variability in weekly sales and the challenges in classifying holiday weeks point to nuanced shopping habits influenced by a variety of factors such as seasonality, regional differences, and promotional events.

Lastly, with respect to other, similar, work on our dataset, we tried to replicate some other teams's work. Reproducing Jeswani [1]'s work has yielded insightful results on the effectiveness of different models and preprocessing strategies for the Walmart sales forecasting task. First, we evaluated the performance of models without dimensionality reduction. Using Mean Absolute Error and Weighted Mean Absolute Error as our metrics, we found that the baseline model without PCA achieved an MAE of 0.1751 and a WMAE of 0.1958, indicating a reasonably high performance in capturing weekly sales trends, including the holiday weeks. Applying PCA revealed limitations with Linear Regression, which saw a significant decrease in accuracy,

yielding both an MAE and WMAE of 0.6438. This increase in error suggests that Linear Regression, in particular, was sensitive to information loss from dimensionality reduction, reducing its ability to accurately capture the variability in sales patterns across holidays and markdowns.

In contrast, Random Forest demonstrated greater resilience to PCA. Although there was a drop in performance compared to the non-PCA baseline, Random Forest with PCA still performed relatively well, with an MAE and WMAE of 0.2740. This indicates that Random Forest was better able to manage the reduced feature set, maintaining a closer alignment to the baseline performance without PCA. These results suggest that, for this dataset, incorporating all available features without PCA yields superior accuracy, particularly for the linear models that benefit from more granular data. In contrast, Random Forest showed a more flexible response to feature reduction, performing acceptably with PCA. Ultimately, our baseline Random Forest model without PCA delivered the lowest error and was the most effective at capturing the holiday-weighted sales fluctuations, making it the preferred model in our forecasting task.

## 4. Related Work

The main focus of Rashmi Jeswani's work[1] is to predict Walmart's sales based on the available historical data and identify whether factors like temperature, unemployment, fuel prices, etc affect the weekly sales of particular stores under study. The study also aims to understand whether sales are relatively higher during holidays like Christmas and Thanksgiving than normal days so that stores can work on creating promotional offers that increase sales and generate higher. Their work also identified the correlation between many details of the physical stores and locations to sales revenue. We plan to reproduce this paper's implementation and then go beyond with our inclusion of further numerical data and textual data pulled from reviews left on various product groups bought from Walmarts.

Aayush Gupta's paper[2] discusses using time series data to identify the impact that holidays have on Walmart sales. The models implemented were expressly designed to not be black boxes so parameters and weights can be finely tuned. This paper utilizes the same data set that Jeswani's work uses with different goals and outcomes. Our method differs from Gupta's work in looking at more things that could impact sales than just holidays. The inclusion of sentiment analysis will also further diversify our work.

Gumusbas's work[3] looks to use the Walmart dataset to forecast sales based on seasons. They took averages of each day in a season to make all the data season-based. This work is similar to Jewani's paper, taking all variables into account for the forecasting model. An avenue that we will potentially pursue is the granularity of the forecasting model. In which I mean, we would look into forecasting weekly sales and monthly and yearly.

Zhao and their team's work[4] is an incredibly detailed implementation and exploration of NLPs and sentiment analysis to forecast crude oil sales. We will be sourcing their insight when using sentiment analysis to forecast Walmart product-type sales. All text was categorized

as either negative, positive, neutral, or compound. These categories alongside oil price time series were run through 5 different models using leave one out cross validation.

## 5. Next Steps

Moving beyond the work presented today, we have many ideas for what can be implemented but were not due to various reasons. There are many more models that we could have tried to implement but due to our time constraints were unable to. It is shocking to see the low performance of our neural network and is one of our greatest places of improvement for the future. Incorporating many of the new skills acquired throughout the semester, like CNNs and better handling over and underfitting, would most likely lead to great improvements in performance. Additionally, we believe that the incorporation of more current and widespread data could assist in boosting the performance across the table. Our data was limited to a strict subset of Walmart locations and broadening this dataset to more locations would probably assist in finding more distinct features of each physical location that contribute to that store's sales numbers. Following this idea of extending the dataset, working in textual data as originally planned would be very helpful in bringing meaning to the data. As reflected in this paper, attaching text review data failed due to how Walmart's online marketplace is a conglomerate and cannot be separated into where each review is coming from. Overall, from our project, there can be many leaping-off points that would lead to greeted insight about the dataset.

## 6. Code and Dataset

See the attached link for the code and data. Our analysis is organized in a jupyter notebook. To recreate the results, make sure the data locations are changed and the data is read incorrectly. From there, the rest of the analysis is automated by the code in the juppyer notebook so running all the code will reproduce the results. The key to the reproducibility of our results is that despite many instances of randomly selecting features, data, or hyperparameters, we used a constant random state (42). This should make the results replicable.

Code: We split our code into two files. See both below

First File:

- https://colab.research.google.com/drive/1EdU2SawMYtQwqRb3WnUfC3rv4KzbULUF?usp=sharing
-

Second File

- https://drive.google.com/file/d/1_OfDqMUOM3tO25gnCefm8KKhqwJewd1J/view?usp=sharing

Data Set:

https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting/data

Slides:

- https://docs.google.com/presentation/d/1HX0kZL0hWUoOSNpqtgezbxsJ_ZqGjiC5qMIwQQ6oxME/edit?usp=sharing

## 7. Conclusion

Through our analysis, we have shown that, regardless of the task or feature set, random forest models perform the best on our data given our scaling and hyperparameter turning. While the NN did not outperform Random Forest in regression or resolve the classification challenges, it provides a baseline for future exploration. Its performance may improve with hyperparameter tuning or additional data preprocessing. If time had allowed it, we could use this baseline in incorporating more complex neural networks that would better explore linear relationships in our data like RNNs, or CNNs. The results of our Neural Network underscore the challenges of applying deep learning to real-world business datasets, where more complex models do not always translate to better performance. Our exploration reaffirmed that while deep learning has significant potential for uncovering complex patterns, it may not be the most suitable approach in this case given the nature and characteristics of the dataset. On the other hand, the SVM and polynomial regression models perform significantly worse than the random forest. We suspect this is due to the relatively limited degree space we explored as well as the relatively small (only 15 combinations) number of hyperparameters combinations we tested. We also suspect our results could improve if we gave, and had the computation power to run the SVM with the full set of data. Thus, overall, we have seen through this entire project, that random forests are incredibly powerful and work well with our mixture of categorical data (easy and clear splits) and geographical spread, narrowing down the answer very quickly and accurately and, to our surprise, outperforming our NN.

## Bibliography

[1]Jeswani, R. (2021, December). *Predicting Walmart sales, exploratory data analysis, and ...* Predicting Walmart Sales, Exploratory Data Analysis, and Walmart Sales Dashboard. https://www.rit.edu/ischoolprojects/sites/rit.edu.ischoolprojects/files/document_library/Rashmi_Jeswani_Capstone.pdf

[2] Gupta, A. (2021, July 28). *Walmart recruiting-store sales forecasting*. Medium. https://medium.com/geekculture/walmart-recruiting-store-sales-forecasting-b8b2f4cf19b1

[3] Gumusbas, Ezgi. (2020). *Project4_Store_Sales_Forecasting*. GitHub. https://github.com/ezgigm/Project4_Store_Sales_Forecasting

[4] Zhao, L.-T., Zeng, G.-R., Wang, W.-J., & Zhang, Z.-G. (2019). Forecasting oil price using web-based sentiment analysis. *Energies, 12*(22), 4291. https://doi.org/10.3390/en12224291

[5] https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting/data

[6] https://www.kaggle.com/datasets/promptcloud/walmart-product-reviews-dataset