Forecasting Future Walmart Sales using KNN/Decision Tree Regression

Muhammad Salman – 100995527 - Masters of Engineering in Electrical and Computer Engineering

Seyedeh Marziyeh Zamiri – 101214196 - Masters of Engineering in Electrical and Computer Engineering

Professor Mikhail Genkin, DATA 5000 Y – March 30th, 2021

Introduction

- Walmart Data set imported from Kaggle
- Comprised of three main CSV files containing correlation of different features, based on data pulled from 45 different store locations, with a number of departments in each store
- Features vary from numerical values, such as amount of sales, to boolean True/False for if a week contains a holiday or not
- Goal is to build/train a model that can predict future Walmart sales based on a specific store/department and date

Dataset

- Relevant CSVs within the dataset are listed below, along with the features within each one
- *features.csv*: Store #, Date, Local Temperature, Local Fuel Price, Markdown/Discounts, CPI index, Unemployment Index, and if the current week is a holiday
- *stores.csv*: Store#, Type of Store, and size of store by number of products
- *train.csv*: Store#, Department#, Date, Weekly Sales, and if the current week is a holiday

Methodology

Data Cleaning

- Once data was imported using Excel, all following data analysis was done using Python
- The first step was to load the data using the Pandas package, and import the CSV table as a data frame
- We proceeded to remove N/A data, as can be seen in the CSV screenshots in the Dataset section
- Once completed, all three data frames were merged into one, named main
- The main data frame was created by merging on common features between the three data frames, such as Store# and Date

```
#Load all excel files in through pandas
train  = pandas.read_csv("train.csv")
features = pandas.read_csv("features.csv")
stores  = pandas.read_csv("stores.csv")

#Some values maybe NA depending on stores
#Clean up data

#For unavaialble markdowns, fill with 0
features['Markdown1'] = features["MarkDown1"].fillna(0)
features['Markdown2'] = features['MarkDown2'].fillna(0)
features['Markdown3'] = features['MarkDown3'].fillna(0)
features['Markdown4'] = features['MarkDown4'].fillna(0)
features['Markdown5'] = features['MarkDown5'].fillna(0)
```

Figure 1: Code to Load CSV and Clean Data

Data Exploration

- The next step was to explore the data and observe strong correlation between features
- The sets that were observed were: Type of Store vs. Weekly Sales, Store# vs. Weekly Sales, Is Holiday vs. Weekly Sales, Week of Year vs. Weekly Sales and more

Observations

- Type of Store vs. Weekly Sales: Weak correlation, Type C seems to have least amount of sales, no strong difference between A and B
- Store# vs. Weekly Sales : Graphed to visualize data, observed which stores normally generate more sales
- Is Holiday vs. Weekly Sales: Some correlation, with slightly more sales when there was a holiday, however not as much as expected
- Week of Year vs. Weekly Sales: A strong rise in sales nearing the holiday season at the end of the year (Thanksgiving-Christmas)

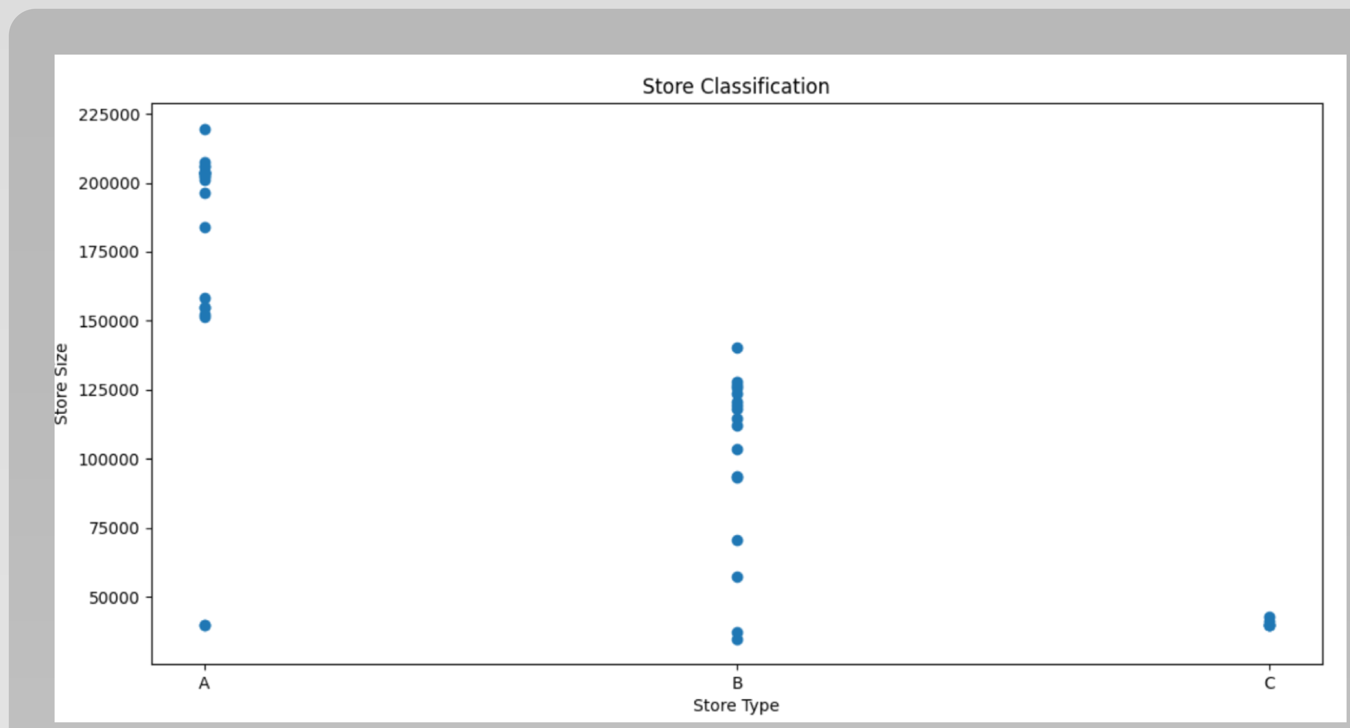

Figure 2: Type of Store vs. Size of Store

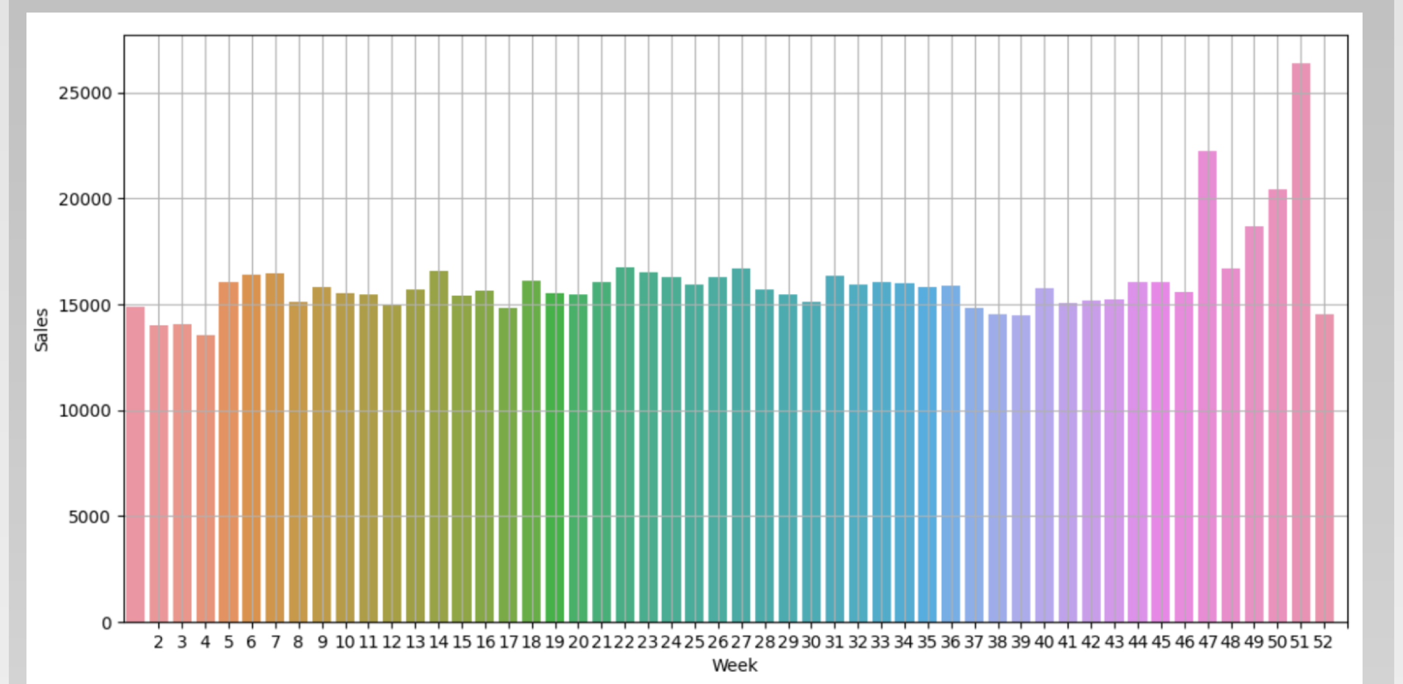

Figure 3: Week of Year vs. Weekly Sales

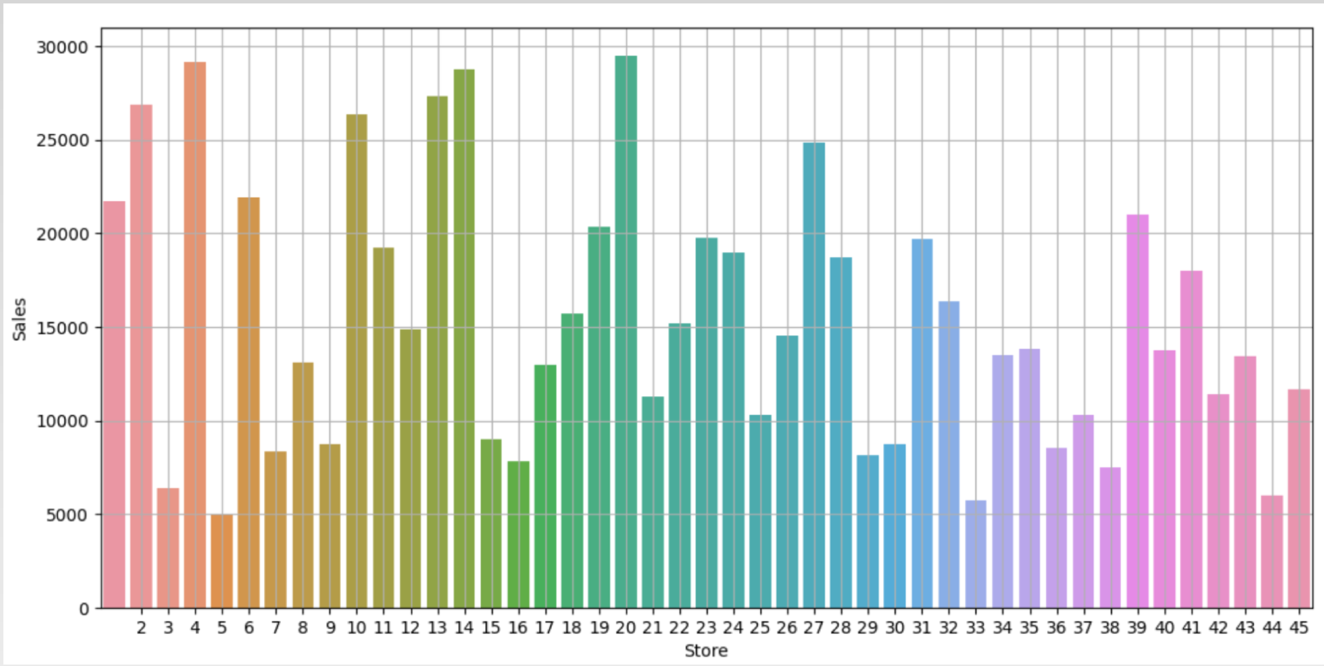

Figure 4: Store# vs. Weekly Sales

Results

Data Modelling

- In order to model the data, a machine learning model is normally selected
- We initially selected the <u>K Nearest Neighbors Regression</u> model
- Regression models allow a solid value to be predicted based on a series of dependent variables, such as we have in our case, since we are predicting future sales per store and department at certain dates
- The KNN approach will allow features to be classified based on their similarities as described with input data
- Predicted values will be classified based on how much their data resembles points in the dataset
- This modelling can be done in python using scikit-learn library in Python
- The strongest model had 15 nearest neighbors selected, with features such as date broken into Y-M-D, where day/year were ignored for better prediction
- However, a stronger model was discovered using Decision Tree regression
- A Decision Tree parses the data through smaller nodes, known as decision nodes, until it achieves the target or leaf node, which is Weekly Sales in this case

KNN Regression

- After adding/removing features, the data was split into 80% train, and 20% test, where the 20% would then be predicted by the model
- The greatest success rate with was 37.8% with 15 Neighbors
- MAE: 11035.06, MSE: 320087846.66, RMSE: 17890. 99

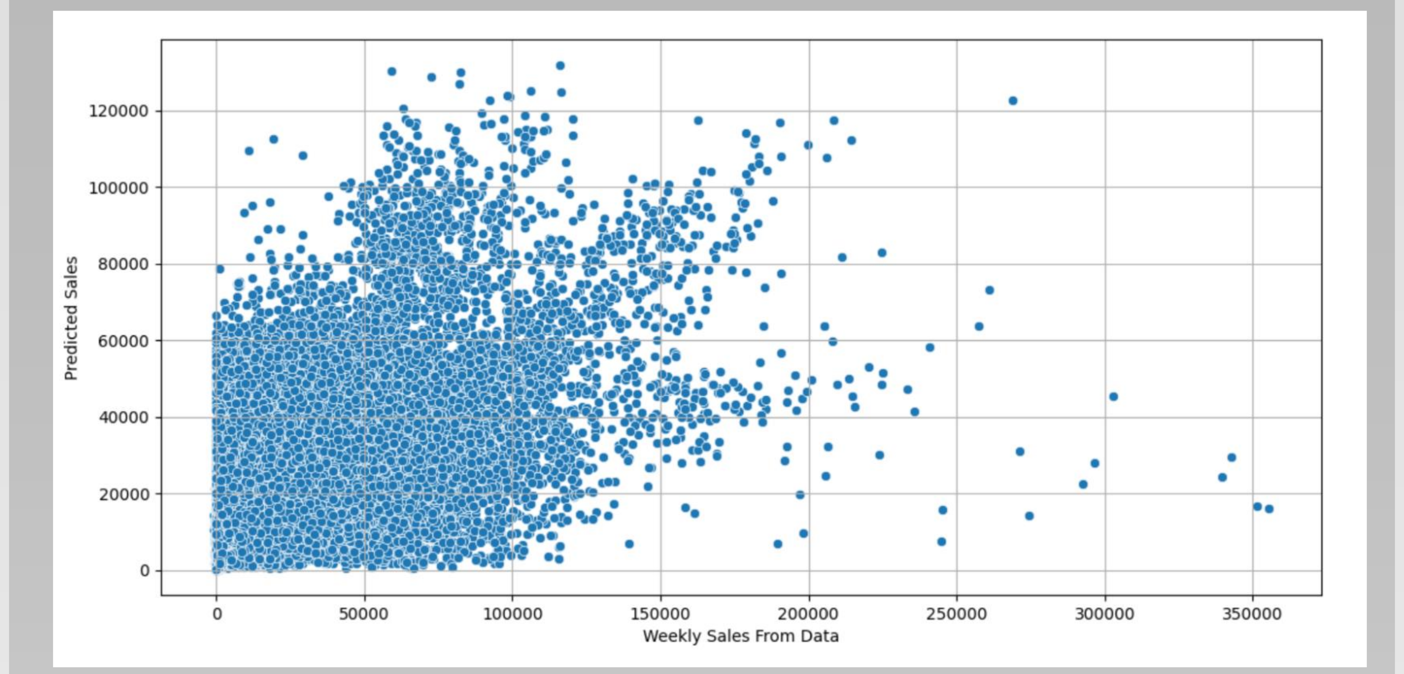

Figure 5: Trained Model with KNN Regression

Decision Tree Regression

- The greatest success rate with was 99%
- MAE: 1.48, MSE: 3457.50, RMSE: 59.56

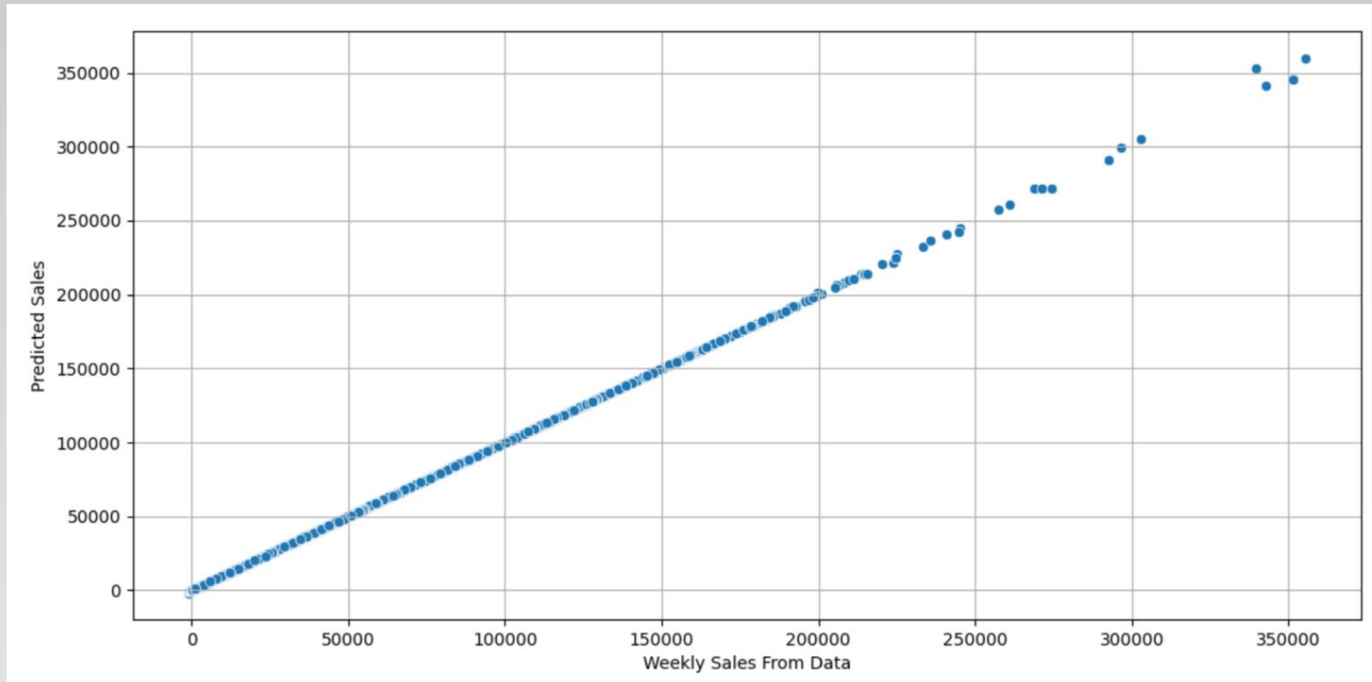

Figure 6: Trained Model with Decision Tree Regression

Conclusion

- The more linear the graph, the stronger the model
- Features such as day throw off model, week# makes it stronger
- Decision Tree stronger than KNN, with the magnitude of error being much less significant
- The MAE, for example, measures the absolute distance between predicted and actual data, which for KNN was quite large, meaning the average predicted sale was off by $11035.06 from its predicted value

References

[1] Kaggle.com. 2014. Walmart Recruiting - Store Sales Forecasting | Kaggle. [online] Available at: <u>https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting/data</u> [Accessed 31 January 2021].

Acknowledgments

We would like to thank the organizers of Data Day for allowing us to present our Walmart Forecaster Project, as a part of the DATA 5000 course.

We would also like to thank Professor Michael Genkin for his dedication and support during the semester, as well as Professor Komeili and Professor Velazquez for teaching the course.