Predicting Weekly Walmart Sales by Store

Harley Sorkin

May 9, 2023

Contents

1 Abstract

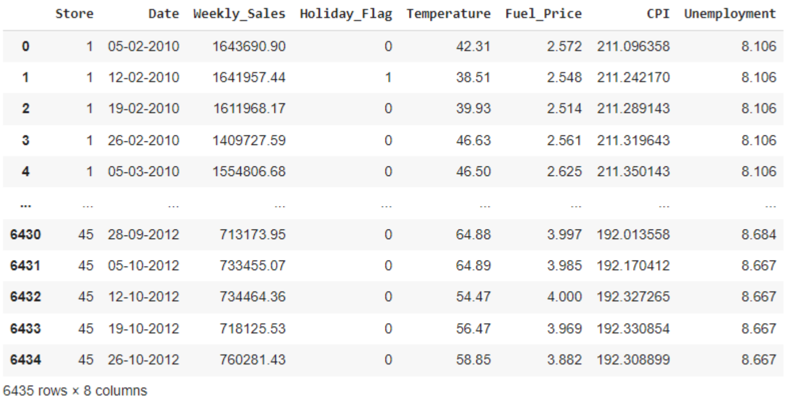
A problem many small retailers face is keeping the right amount of product in stock. Order too much, and the retailer is forced to discount products due to lack of demand. Order too little, and risk losing out on major profits due to over-demand. The goal of this project is to create a neural network model to help accurately predict the weekly sales for a retailer. By being able to accurately predict a weeks given sales, that store might be able to better order inventory and staff appropriately. The goal is to create a model that will be able to predict weekly sales, and show which external factors most affect that prediction.

This project specifically is modeled after the Walmart data set posted on Kaggle (Found here).

2 Data Analysis

2.1 Data Cleaning

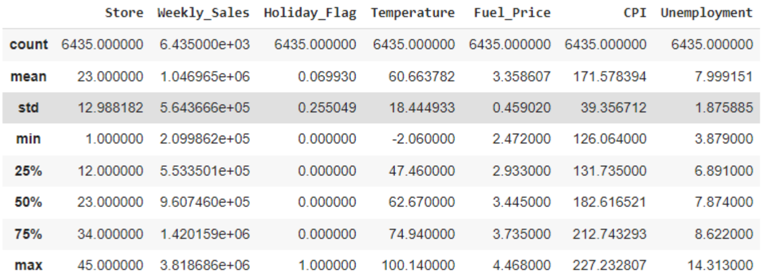
The data provided in this data set is already formatted for easy manipulation, except for the date feature. To make the date more readable the date was converted from a date-time object with syntax ”dd-mm-yy” to an integer representing the amount of weeks since the first date listed per store, i.e. the first date for a given store would be 1, the second date would be 2, 3, and so on. This column is later to be dropped to prevent creating a time-series based model, but is used for preprocessing and visual analysis. The features include store number (int64), date (object), weekly sales (float64), holiday flag (int64), temperature (float64), fuel price (float64), CPI (float64), and unemployment rate (float64).



	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
0	1	05-02-2010	1643690.90	0	42.31	2.572	211.096358	8.106
1	1	12-02-2010	1641957.44	1	38.51	2.548	211.242170	8.106
2	1	19-02-2010	1611968.17	0	39.93	2.514	211.289143	8.106
3	1	26-02-2010	1409727.59	0	46.63	2.561	211.319643	8.106
4	1	05-03-2010	1554806.68	0	46.50	2.625	211.350143	8.106
...	...	...	...	...	...	...	...	...
6430	45	28-09-2012	713173.95	0	64.88	3.997	192.013558	8.684
6431	45	05-10-2012	733455.07	0	64.89	3.985	192.170412	8.667
6432	45	12-10-2012	734464.36	0	54.47	4.000	192.327265	8.667
6433	45	19-10-2012	718125.53	0	56.47	3.969	192.330854	8.667
6434	45	26-10-2012	760281.43	0	58.85	3.882	192.308899	8.667

6435 rows × 8 columns

Figure 1: Walmart.csv dataframe



	Store	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
count	6435.000000	6.435000e+03	6435.000000	6435.000000	6435.000000	6435.000000	6435.000000
mean	23.000000	1.046965e+06	0.069930	60.663782	3.358607	171.578394	7.999151
std	12.988182	5.643666e+05	0.255049	18.444933	0.459020	39.356712	1.875885
min	1.000000	2.099862e+05	0.000000	-2.060000	2.472000	126.064000	3.879000
25%	12.000000	5.533501e+05	0.000000	47.460000	2.933000	131.735000	6.891000
50%	23.000000	9.607460e+05	0.000000	62.670000	3.445000	182.616521	7.874000
75%	34.000000	1.420159e+06	0.000000	74.940000	3.735000	212.743293	8.622000
max	45.000000	3.818686e+06	1.000000	100.140000	4.468000	227.232807	14.313000

Figure 2: Dataframe statistics

2.2 Preprocessing Analysis

The following are the histogram distributions for all 45 stores listed in the data set, before normalization. The goal is to see if there are any values that occur more or less than others, which can then be targeted to evaluate the impact those values have on predicting weekly sales.

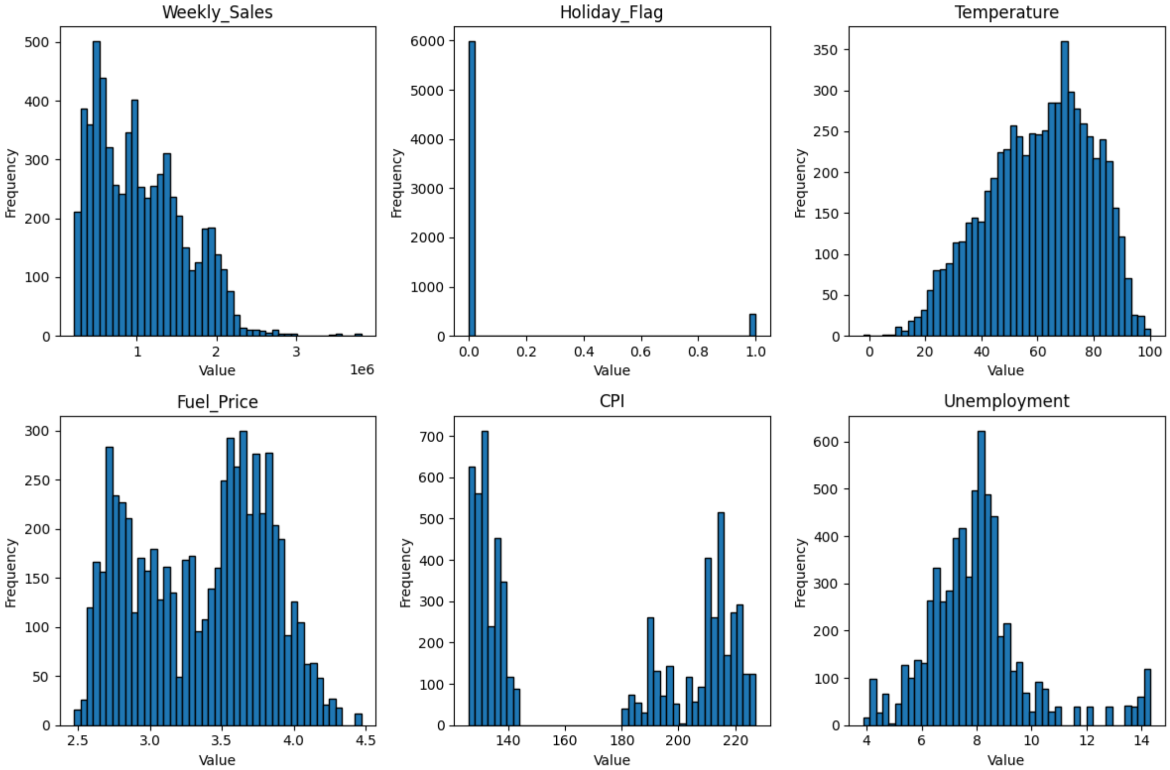


Figure 3: Attribute Distributions

The model will likely be more accurate at predicting weekly sales when a store historically makes around \$500,000/week, when the temperature is in the mid-70’s, fuel prices are somewhere between \$3.50-4.00/gallon, less than 150% CPI, and around 8% unemployment.

2.3 Normalization

I started by normalizing my data using the min-max normalization technique, using the equation:

$$normalized = \frac{value - min}{max - min}$$

The min-max normalization presented issues, however. Namely because one of the features, ’Holiday_Flag’, is represented as a binary float, only containing the values 0 and 1. Because of this, min-max normalization has no affect on these values, and results in unwanted bias towards the holiday flag due to those values being outliers.

While the min-max normalization is helpful for creating readable plots, Z-score normalization helps to prevent outlying values from having too much of an affect on the model. For the purposes of creating a more accurate and unbiased model, Z-score normalization was implemented. This was accomplished using the equation:

$$normalized = \frac{value - mean}{std}$$

2.4 Finding Identifiable Data

By graphing the min-max normalized values for weekly sales, holiday flag, temperature, fuel price, CPI, and unemployment rate against the date, it becomes easy to pick out which features have the biggest impacts on weekly sales. The following are 3 graphs that show the data for stores 1, 22, and 45:

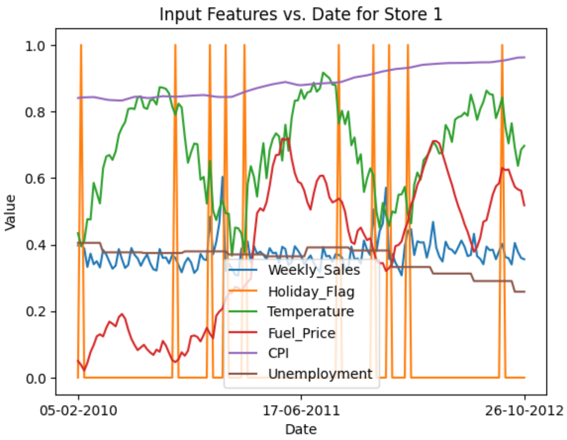


Figure 4: Store 1

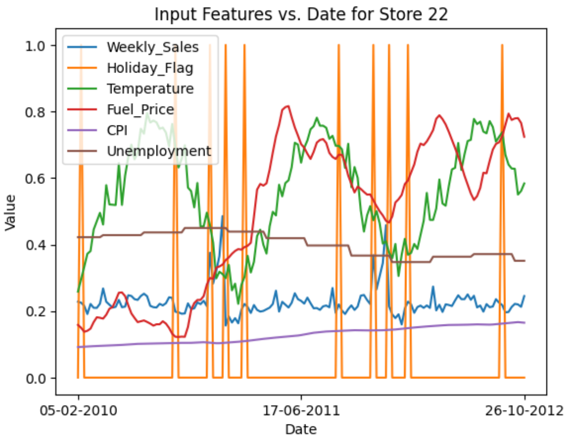


Figure 5: Store 22

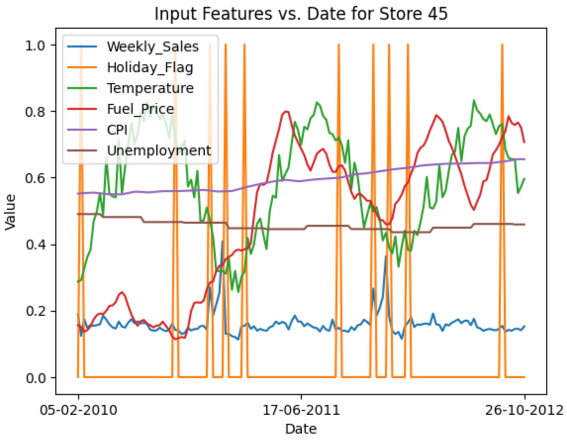


Figure 6: Store 45

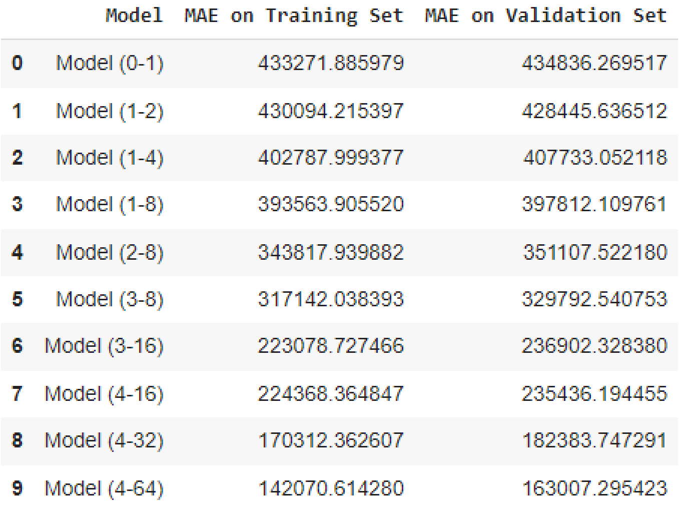
There are a few notable relationships, namely between weekly sales against the holiday flag, and weekly sales against temperature. Holiday flags typically see a marked increase in weekly sales, so it should be expected that if a holiday flag is present sales will be higher. The other important relationship to note is that just before the coldest times of the year, weekly sales are at the highest they reach. Because the date feature is not being used for the purposed of this project, tracking changes in temperature is not possible, and may cause an unwanted relationship between certain temperatures and weekly sales values.

3 Creating a Model

3.1 Model Selection and Evaluation

To gauge the effectiveness of the model, the MAE was calculated both using the normalized data and the inverse transformed predictions versus actual values. This was done so that the MAE can be read as a dollar amount correlating to weekly sales, where the value is the mean difference between predicted weekly sales and actual values.

The model was split 75% for testing, and 25% for validation, and trained for 100 epochs. A baseline linear regression model was used, and gradually increased in size. Overall, 10 models were tested. In the following table (Fig. 7), the MAE for both the training and validation sets are listed, inverse transformed, for each model. The size of the model can be read as the first number being the number of layers, and the second number is the number of nodes per layer. There are 2 additional layers, a single neuron output and an 6 neuron input. For example, Model (3-8) will have the shape 6-8-8-1.



	Model	MAE on Training Set	MAE on Validation Set
0	Model (0-1)	433271.885979	434836.269517
1	Model (1-2)	430094.215397	428445.636512
2	Model (1-4)	402787.999377	407733.052118
3	Model (1-8)	393563.905520	397812.109761
4	Model (2-8)	343817.939882	351107.522180
5	Model (3-8)	317142.038393	329792.540753
6	Model (3-16)	223078.727466	236902.328380
7	Model (4-16)	224368.364847	235436.194455
8	Model (4-32)	170312.362607	182383.747291
9	Model (4-64)	142070.614280	163007.295423

Figure 7: Table showing results for various model sizes given 75/25 split for training and validation

Based on these results alone Model (4-64) appears to be the most accurate, with a significantly lower MAE for training and validation sets than any other model. Looking at the learning curves, however, shows that Model (4-32) has a much better fit with less noise.

To further validate the choice of Model (4-32), plotting the actual versus predicted values can show variance in the accuracy. The closer the points are to the black diagonal line indicate strong accuracy in predicting 'Weekly_Sales' values. In comparison, here are the plots for Model (3-8):

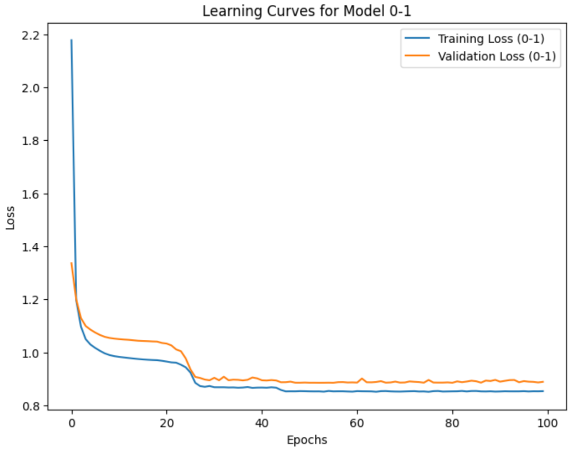


Figure 8: Model (0-1)

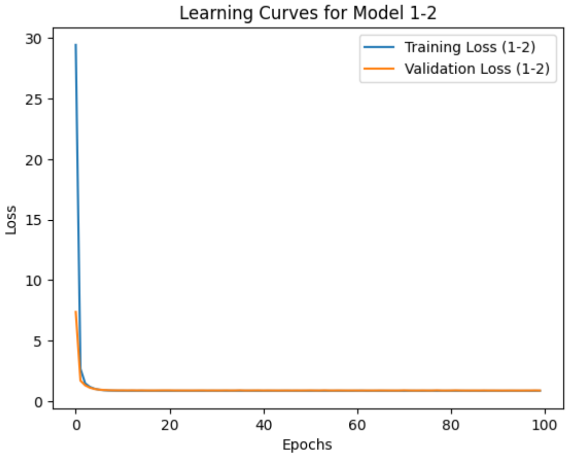


Figure 9: Model (1-2)

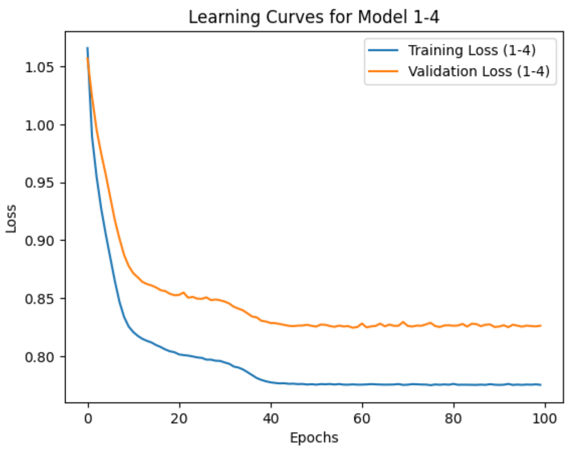


Figure 10: Model (1-4)

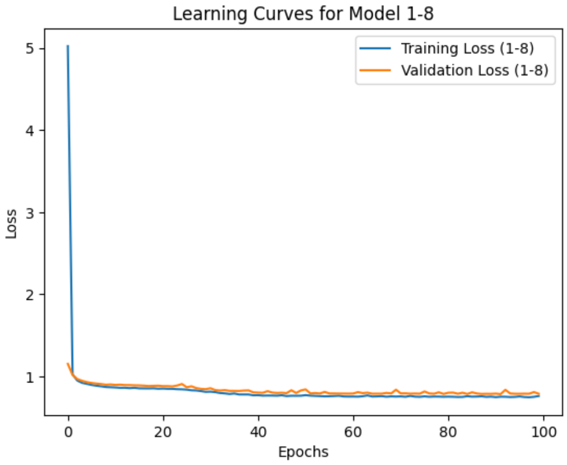


Figure 11: Model (1-8)

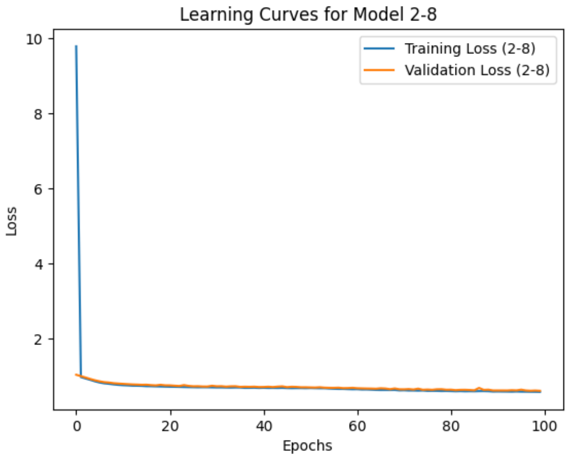


Figure 12: Model (2-8)

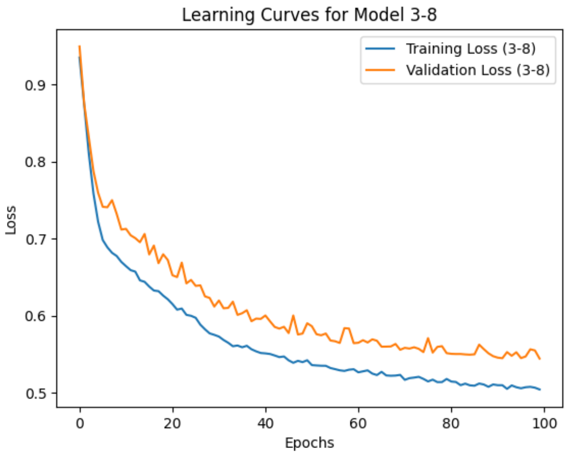


Figure 13: Model (3-8)

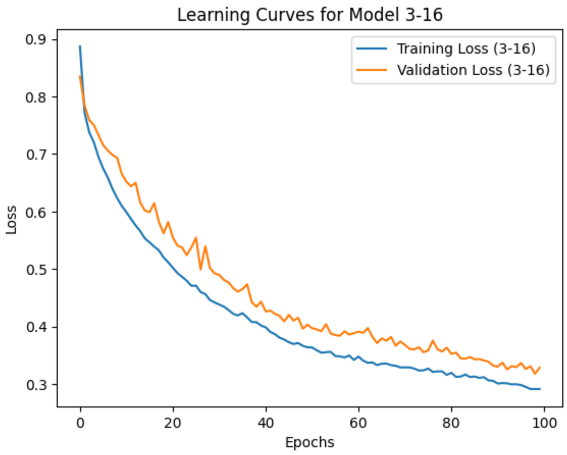


Figure 14: Model (3-16)

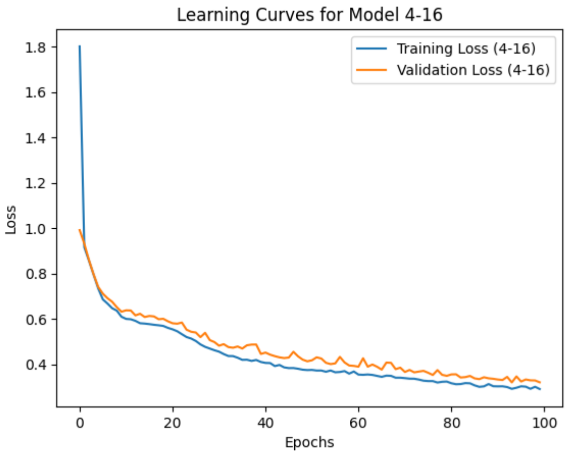


Figure 15: Model (4-16)

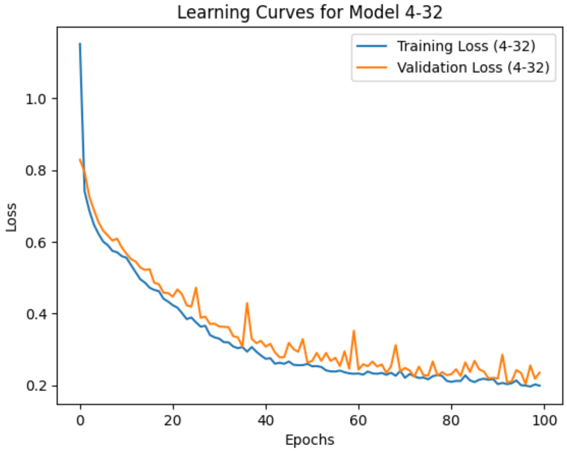


Figure 16: Model (4-32)

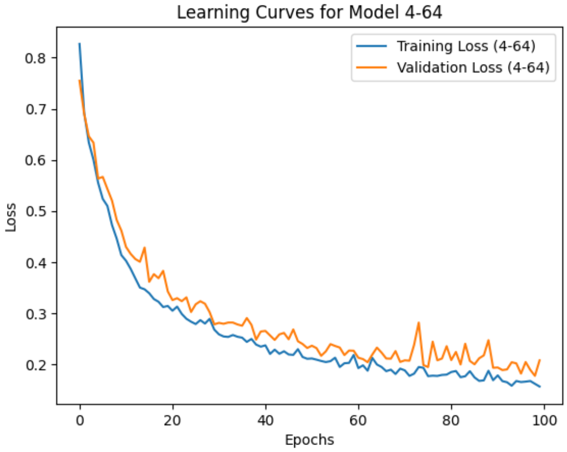


Figure 17: Model (4-64)

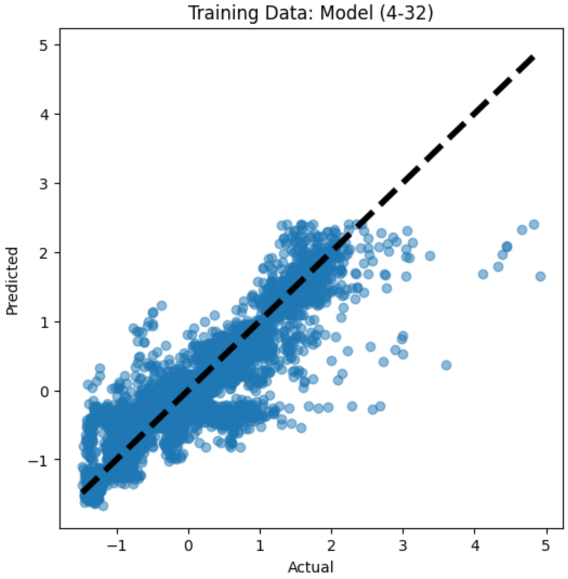


Figure 18: Model (4-32) Training Actual vs. Predicted Values

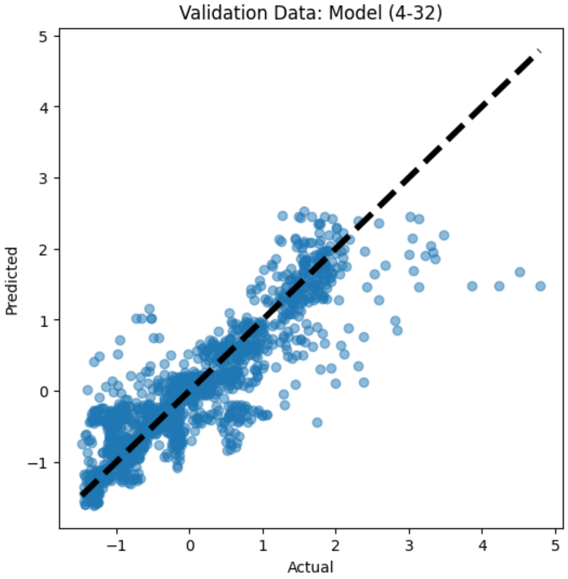


Figure 19: Model (4-32) Validation Actual vs. Predicted Values

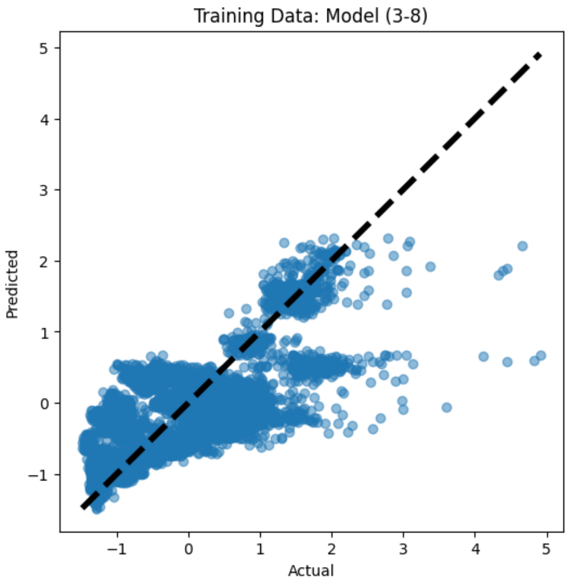


Figure 20: Model (3-8) Training Actual vs Predicted Values

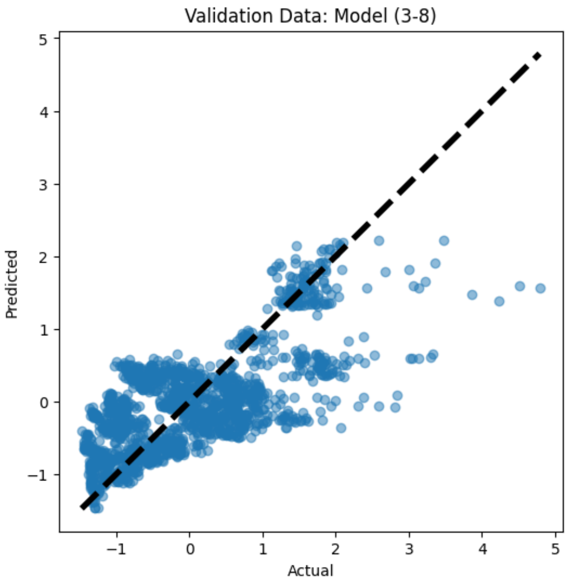


Figure 21: Model (3-8) Validation Actual vs. Predicted Values

3.2 Feature Importance

In order to determine feature importance the model was trained on the z-score normalized data set using Model (4-32) using one feature at a time. The features that produced the lower MAE values are deemed more important. The model was then tested in an iterative pattern, starting with all 6 inputs and dropping the least important feature after each fitting until the model is only trained on the 'Store' feature. The order of inputs to drop was determined to be CPI, Temperature, Holiday_Flag, Fuel_Price, Unemployment, and lastly Store.

Removing a singular feature at a time resulted in the accuracy of the model decreasing significantly. Oddly enough, however, the MAE decreased after removing both CPI and Temperature, but the change is so insignificant it can be chalked up to model variance.

When only the 'Store' feature remains, however, the MAE does not stray far beyond twice the value found when all 6 inputs are kept. This value is still high (around 300,000), but only training on unimportant features such as only based on CPI results in MAE around 480,000.

4 Conclusion

Using the Walmart data set posted by Walmart on Kaggle, this project aims to predict the 'Weekly_Sales' feature without the use of the 'Date' feature. A neural network model with the architecture 6-32-32-32-32-1 and trained for 100 epochs over the remaining 6 features is able to predict values of 'Weekly_Sales' with a Mean Absolute Error of around 170,000 after inverse transforming the predicted and actual values to their original scale. This means that for any given Walmart store the weekly sales can be predicted within around $170,000 (not bad when stores make weekly sales in the millions), given the right data points are collected.

Throughout the process of creating the model, it was found that certain features have less of an impact than others. Names the 'Store', 'Unemployment', and 'Fuel_Price' features appeared to have the greatest impact on the accuracy of the model.