Understanding Data and their Environment

Student ID: 201276930

May 2019

Word count: **2750**

Contents

1 Introduction

In this report the analyst will outline the analytical approach taken to provide sales forecasts to produce a predictive model that will predict sales on departmental level for a nationwide retailer in the United States. Analysts were provided with historical sales data, however, the data had several problems which were rectified (these will be addressed below). This report covers the following:

- Analysis of the available data, quality description, variables and relevance to sales prediction.
- Perform initial exploratory analysis by the creation of 5-number summary, Bi and Multivariate inspection and finally co-variance, pairwise correlation and t-test.
- Dataset linkage.
- Preprocessing the data after linkage: null checker, plotting the data, converting data types and so on.
- Identifying key factors that affect sales by producing multiple graphs and association tests.
- Aggregating the data to pre-holiday and holiday weeks to understand the effects of promotional marketing.
- Building multiple predictive models, **Deep Learning Imputation** using Datawig, **Linear regression** (pre and post imputation) and **Random Forest regression**.

2 Reproducibility

The tools used to analyse and manipulate the data are:

- Python 2.7
- Jupyter Notebook
- Git (Version Control)
- Pandas
- Numpy
- Matplotlib
- Seaborn
- Scipy
- Statsmodels

- Sklearn
- Datawig (Deep Learning imputation)

All of the work presented in this report is reproducible with any kind of sales data. It can be found in the Appendix.

3 Data review

There are four different datasets provided, these are:

- stores.csv - information about stores including ID number and size.
- features.csv - information related to stores, departments and regional activity.
- train.csv - contains historical training data covering sales from 2010 to 2012.
- test.csv - is identical to train.csv but we predict the weekly sales column.

3.1 Stores

This dataset contains three variables:

- Store
- Type
- Size (sq ft)

Immediately, the analyst spotted that **Store** is a common variable, it exists in the other three datasets. This means the analyst may be able to merge datasets on the column Store. Most of the data in this table is categorical (Store and Type) but **Size (sq ft)** is an integer and each store has a unique size. All 45 entries are filled with data meaning no imputation is required. Size may have a relevance to Sales, as the store size increases one may expect that sales also increase Yoo et al. (1998).

3.2 Features

This dataset contains twelve variables, these are:

- Store
- Date
- Temperature
- Fuel_Price

- Promotions 1 to 5
- CPI
- Unemployment
- IsHoliday

One of the problems immediately identified was the number of missing rows of data. The total size of the dataset is 8189 entries but for some columns there are less, which means the quality of the dataset as a whole is not so great, for example:

- Promotion1: 4032
- Promotion2: 2921
- Promotion3: 3613
- Promotion4: 3464
- Promotion5: 4050
- CPI: 7605
- Unemployment: 7605

There are four different data types in this dataset, **Date** is an Object and not a date-time dtype. This means series analysis cannot be performed until the date column is converted to a date-time type for all datasets.

Fuel_Price is the cost of fuel in the region, this means it can have an affect on sales. If the fuel price decreases this could then increase weekly sales as more people may be inclined to travel by car to stores and purchase more than that which they would have if they went on foot due to storage capacity. Promotions can be an incentive for people to purchase products as they may save money. Expect to see spikes in sales when promotions are on, especially during public holidays. Another problem identified with promotions is the amount of minus values present, these could mean savings when purchasing some products but not very clear. The consumer price index measures the weighted average of prices of a basket of consumer goods. Accurately measuring the rate of change and inflation is important to see how sales fluctuate during these periods Boskin et al. (2011), but at this moment it is not clear what the values indicate. The unemployment variable is highly unlikely to be relevant to sales, however, one can argue that the higher the unemployment rates, the lower the weekly sales as the majority of unemployed people will spend less as they have less. In the paper Ganong et al. (2016) 6% of monthly spending is reduced for unemployed people. The IsHoliday variable can be used to compare holidays to non-holiday weekly spending. T-test can be adopted for the holiday variable to measure the size of the difference relative to the variation in the sample data.

3.3 Train

The variables in this dataset are:

- Store
- Department
- Date
- Weekly_Sales
- IsHoliday

The Weekly_Sales variable contains sales information for the given department at each store. A problem with the data is that it contains minus values but this is very small so the impact is minuscule (also the minus values can indicate losses). We only have sales data from 2010 to 2012. The problem with department is that its anonymised, if each department was identifiable then the analyst can identify what each department sold then you can see what products sell more over time. For example, you would expect DIY departments to sell more over Christmas.

3.4 Test

The test dataset is the same as **Train.csv** the only unique column it contains is **Weekly_Sales to be predicted**. The predictive models will populate this dataset.

4 Exploratory analysis

This chapter contains discussions of the various approaches taken to explore the data Abzalov (2016) before performing data wrangling to link the data-sets together (analysis techniques are performed after the data linkage stage). Some of the methods used are:

- Null checker
- 5-number summary *count, mean, std, min, 25%, 50%, 75%* and *Max*
- Scatter plots
- Line plots
- Covariance

When the data is first imported into Jupyter Notebook, you must check how many cells are null values, to do this a short **null_checker** function was created (can be found in the Appendix), the results were tabulated:

	Stores	Features	Test	Train
Store	0	0	0	0
Type	0			
Size	0			
Date		0	0	0
Temp		0		
Fuel_P		0		
Promotion1		4158		
Promotion2		5269		
Promotion3		4577		
Promotion4		4726		
Promotion5		4140		
CPI		585		
Unemployment		585		
IsHoliday		0	0	0
Department			0	0
Weekly_Sales				0
Weekly_Salestbp			115064	

Figure 1: The amount of empty cells

In Figure 1, you can see most of the data that is missing is in the **Features** data-set. The promotions are important as they are directly associated with sales, in the later chapters the analyst will discuss the different approaches to deal with this issue.

The 5-number summary for both **Feature** and **Train** were calculated as these tables contain the most useful data. The results were as follows:

	Store	Temperature	Fuel_Price	Promotion1	Promotion2	Promotion3	Promotion4	Promotion5	CPI	Unemployment
count	8190.000000	8190.000000	8190.000000	4032.000000	2921.000000	3613.000000	3464.000000	4050.000000	7605.000000	7605.000000
mean	23.000000	59.356198	3.405992	7032.371786	3384.176594	1760.100180	3292.935886	4132.216422	172.460809	7.826821
std	12.987966	18.678607	0.431337	9262.747448	8793.583016	11276.462208	6792.329861	13086.690278	39.738346	1.877259
min	1.000000	-7.290000	2.472000	-2781.450000	-265.760000	-179.260000	0.220000	-185.170000	126.064000	3.684000
25%	12.000000	45.902500	3.041000	1577.532500	68.880000	6.600000	304.687500	1440.827500	132.364839	6.634000
50%	23.000000	60.710000	3.513000	4743.580000	364.570000	36.260000	1176.425000	2727.135000	182.764003	7.806000
75%	34.000000	73.880000	3.743000	8923.310000	2153.350000	163.150000	3310.007500	4832.555000	213.932412	8.567000
max	45.000000	101.950000	4.468000	103184.980000	104519.540000	149483.310000	67474.850000	771448.100000	228.976456	14.313000

Figure 2: Summary of Feature dataframe

In Figure 2, the fuel_price and unemployment are closer to the expected value (mean) where as Promotions 1 to 5 have a high standard deviation meaning they are spread out over a wider range of values. The reason for this is because Promotions 1 to 5 have a lot of missing data. So does CPI thus it has a high standard deviation.

In Figure 3, the Store variable has a very similar standard deviation with the Store variable in feature_df meaning they can be merged. Weekly_Sales has a high standard deviation which means spikes can be expected due to the values being spread out. There are over +421000 weekly sales records, this can help predict future sales using a Machine Learning model.

	Store	Dept	Weekly_Sales
count	421570.000000	421570.000000	421570.000000
mean	22.200546	44.260317	15981.258123
std	12.785297	30.492054	22711.183519
min	1.000000	1.000000	-4988.940000
25%	11.000000	18.000000	2079.650000
50%	22.000000	37.000000	7612.030000
75%	33.000000	74.000000	20205.852500
max	45.000000	99.000000	693099.360000

Figure 3: Summary of Train dataframe

six scatter plots were produced comparing data that is believed to contain the most analytical problems (after the data wrangling process more graphs were produced to compare a more wider range of variables).

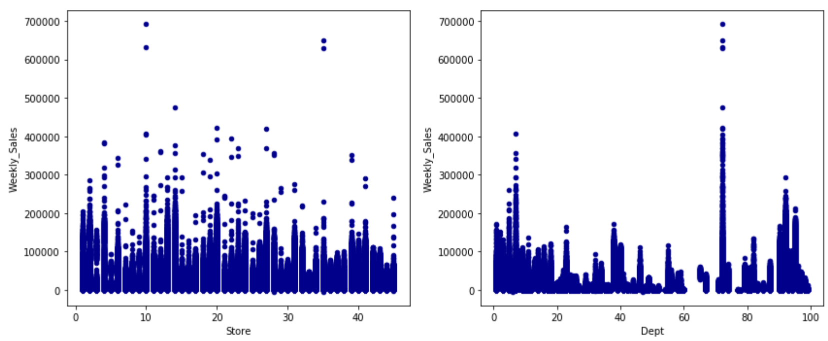

Figure 4: Scatter plots of Store and Department compared with Weekly Sales

In Figure 4, you can immediately see that there are less stores that sold over 300000 weekly sales. On a departmental level you can see that departments from 0-20 have a much higher concentration of products sold. Departments from 60-100 have also sold quite a lot of products especially department 70. It could mean that these spikes are due to holiday seasons and must be analysed further.

In Figure 5, you can see that there is not much of a difference when it comes to Promotions 1 and 2. Promotion 1 is more dispersed after 40000 than Promotion 2.

In Figure 6, a major problem was discovered in the variable CPI. CPI looks like it has values in two different ranges and outliers. You can either normalise CPI using min-max scaling or any other normalisation technique or impute the missing values and see what happens. The analyst personally doesn't believe CPI would make a big impact on the R^2 therefore you don't really need to use it in your training set.

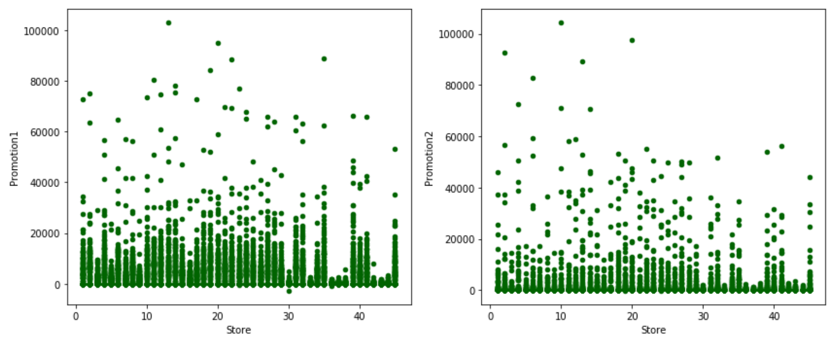

Figure 5: Scatter plots of Store compared with Promotions 1 and 2

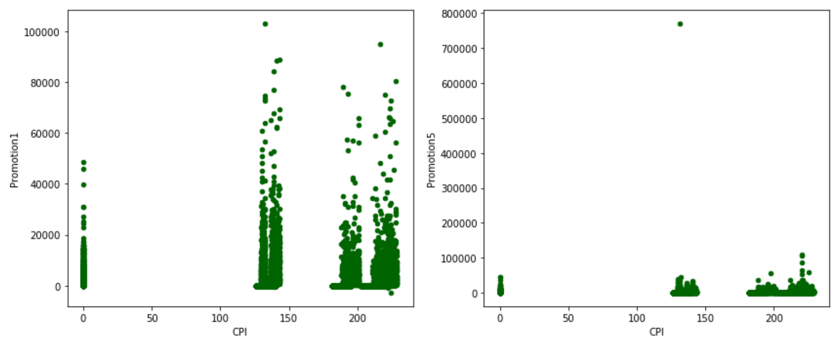

Figure 6: Scatter plots of CPI compared with Promotions 1 and 5

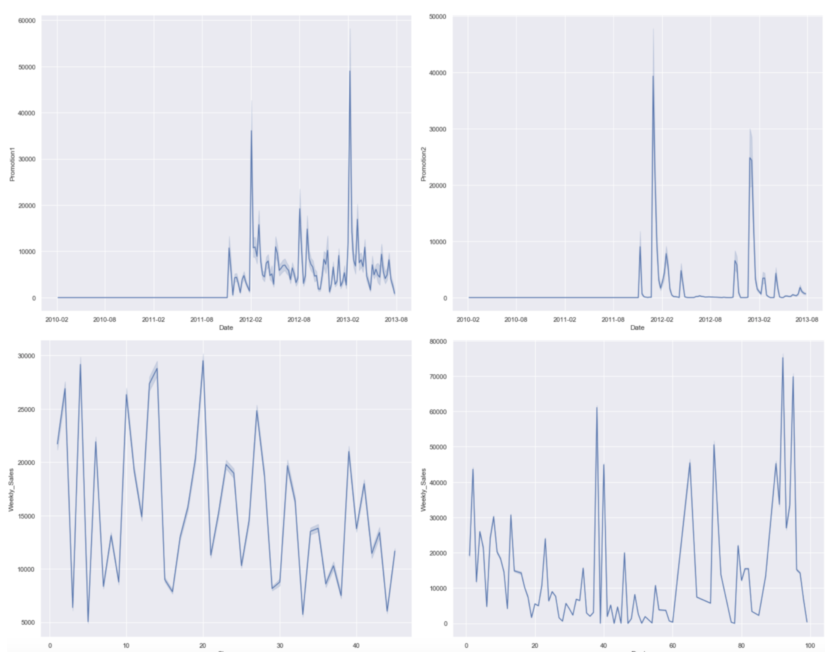

Figure 7: Line plots of promotions and weekly sales

In Figure 7, you can see that promotion data is non existent from 02-2010 to 12-2011. For Promotion1 there are spikes on new year day, super bowl, a week before thanksgiving there is another spike and a very big spike a few days before Christmas. For Promotion2 there are only two spikes and these indicate promotions targeted at Christmas shoppers. It is likely that Promotion2 was applied to departments that sold Christmas gifts and DIY. Some stores have made more weekly sales than others but the general trend is similar among all stores. Departments are more interesting as you can see spikes for departments 38 to 40 and 60 to 98. These departments probably sell products which have promotions on or products which are more popular.

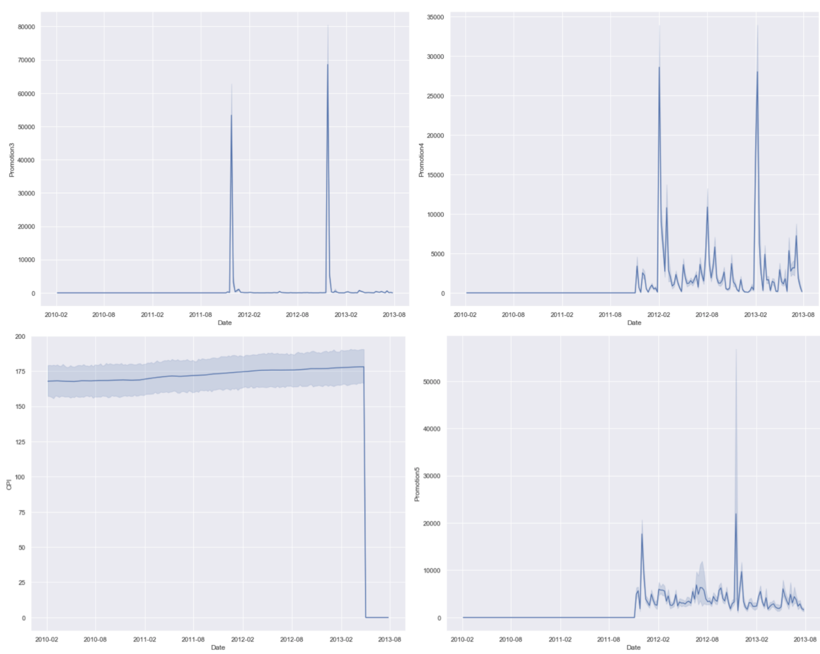

Figure 8: Line plots of promotions and CPI

In Figure 8, you can see similar trends between all four promotions. Promotion3 has only two spikes for Christmas and Promotion4 has several spikes but two big ones which indicate Christmas. It is clear that Promotions 2 and 3 are targeted at Christmas gifts only. Promotions 1 and 4 on the other hand are focused on all holidays but also include Christmas sales. CPI has a very linear growth and starts from 126 to 268, nothing informative can be deduced from CPI. Promotion 5 is very similar to Promotions 1 and 4 as there are only big spikes for Christmas.

Co-variance is a measure of the joint variability of two random variables Miller & Chapman (2001). In Figure 9, the analyst calculated the co-variance of all the numeric columns, you can see that there are:

	Store	Temperature	Fuel_Price	Promotion1	Promotion2	Promotion3	Promotion4	Promotion5	CPI	Unemployment
Store	168.687263	-4.989264	0.373565	-8.337152e+03	-3.651613e+03	-2.736468e+03	-3.427193e+03	-1.006021e+03	-100.299091	5.006917
Temperature	-4.989264	348.890354	0.816587	-1.583846e+04	-2.259900e+04	-8.842017e+03	-7.445559e+03	-3.470960e+03	-56.073624	-2.775438
Fuel_Price	0.373565	0.816587	0.186051	8.381261e+02	7.129273e+01	9.219665e+01	3.022976e+02	4.076271e+02	-5.662715	-0.144643
Promotion1	-8337.152336	-15838.463844	838.126086	5.459608e+07	6.382950e+06	-1.028792e+06	2.874332e+07	1.242641e+07	-17583.402352	-2625.933908
Promotion2	-3651.613028	-22599.003490	71.292731	6.382950e+06	3.020114e+07	-6.799934e+05	1.774877e+06	2.590959e+06	6947.946632	-318.754005
Promotion3	-2736.468246	-8842.017389	92.196654	-1.028792e+06	-6.799934e+05	5.685090e+07	-5.026811e+05	7.170161e+05	6119.991482	-135.797915
Promotion4	-3427.192765	-7445.559475	302.297581	2.874332e+07	1.774877e+06	-5.026811e+05	2.215690e+07	5.309252e+06	-14861.276798	-1258.980667
Promotion5	-1006.020707	-3470.959744	407.627060	1.242641e+07	2.590959e+06	7.170161e+05	5.309252e+06	8.894797e+07	-10020.404904	-1495.195493
CPI	-100.299091	-56.073624	-5.662715	-1.758340e+04	6.947947e+03	6.119991e+03	-1.486128e+04	-1.002040e+04	3439.299976	68.543342
Unemployment	5.006917	-2.775438	-0.144643	-2.625934e+03	-3.187540e+02	-1.357979e+02	-1.258981e+03	-1.495195e+03	68.543342	7.335949

Figure 9: Summary of Feature co-variance association

- Positive linear relationship between temperature and fuel_price.
- fuel_price on the other hand has a positive relationship with store, temperature and promotions 1 to 5.
- Promotion1 has a positive linear relationship with fuel_price, promotions 2, 4 and 5.
- Promotion2 has a co-variance with fuel_price, promotions 1, 4 and 5 and CPI.
- Promotion3 has a co-variance with: fuel_price, promotion5 and CPI.
- Promotion4 has a positive co-variance with: fuel_price, promotions 1, 2 and 5.
- Promotion5 has a positive co-variance with: fuel_price and promotions 1 to 4.
- CPI has a positive co-variance with: promotion2, promotion3 and unemployment.
- Unemployment has a positive co-variance with: store and CPI.

It would be interesting to focus on the above in the pre-processing.

5 Data wrangling

In this chapter the analyst merges the datasets together to compare the various variables and also run predictive models to populate the predicted sales column in the test dataset.

Both Feature and Store tables have a common variable **Store**, you can merge on this variable first, then merge the newly created dataframe with Train, the resulting table is:

```
final_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 421570 entries, 0 to 421569
Data columns (total 16 columns):
Store           421570 non-null int64
Dept            421570 non-null int64
Date            421570 non-null datetime64[ns]
Weekly_Sales    421570 non-null float64
IsHoliday       421570 non-null bool
Temperature     421570 non-null float64
Fuel_Price      421570 non-null float64
Promotion1      421570 non-null float64
Promotion2      421570 non-null float64
Promotion3      421570 non-null float64
Promotion4      421570 non-null float64
Promotion5      421570 non-null float64
CPI             421570 non-null float64
Unemployment    421570 non-null float64
Type            421570 non-null object
Size (sq ft)    421570 non-null int64
dtypes: bool(1), datetime64[ns](1), float64(10), int64(3), object(1)
memory usage: 51.9+ MB
```

Figure 10: Store, Train and Feature merged

Just to make sure you don't have problems with the Date column after merge, you must check the start and end dates which should be 2010 and 2012:

```
final_df['Date'].min()
```

```
Timestamp('2010-02-05 00:00:00')
```

```
final_df['Date'].max()
```

```
Timestamp('2012-10-26 00:00:00')
```

Figure 11: Merged dataframe start and end dates

Now that you have a merged dataframe you can compare promotions with weekly sales and so on. Merging allows you to align the rows from each dataframe based on common attributes or columns. The analyst will join Test with the table from Figure 10 after pre-processing for the predictive analysis stage.

6 Quantitative analysis

In this short chapter, the analyst will present a t-test and rank correlation analysis of the newly created dataframe.

	Weekly_Sales	Temperature	Fuel_Price	Promotion1	Promotion2	Promotion3	Promotion4	Promotion5	CPI	Unemployment
Weekly_Sales	1.000000	-0.002312	-0.000120	0.047172	0.020716	0.038562	0.037467	0.050465	-0.020921	-0.025864
Temperature	-0.002312	1.000000	0.143859	-0.026415	-0.179672	-0.056026	-0.050281	-0.014752	0.182112	0.096730
Fuel_Price	-0.000120	0.143859	1.000000	0.297056	0.029153	0.018615	0.166622	0.215420	-0.164210	-0.033853
Promotion1	0.047172	-0.026415	0.297056	1.000000	0.174868	-0.014411	0.838904	0.415050	0.010915	-0.105168
Promotion2	0.020716	-0.179672	0.029153	0.174868	1.000000	-0.006080	0.113250	0.131735	-0.003554	-0.041427
Promotion3	0.038562	-0.056026	0.018615	-0.014411	-0.006080	1.000000	-0.012020	0.042471	-0.005839	-0.018078
Promotion4	0.037467	-0.050281	0.166622	0.838904	0.113250	-0.012020	1.000000	0.303370	-0.002047	-0.076513
Promotion5	0.050465	-0.014752	0.215420	0.415050	0.131735	0.042471	0.303370	1.000000	0.067906	-0.120406
CPI	-0.020921	0.182112	-0.164210	0.010915	-0.003554	-0.005839	-0.002047	0.067906	1.000000	-0.299953
Unemployment	-0.025864	0.096730	-0.033853	-0.105168	-0.041427	-0.018078	-0.076513	-0.120406	-0.299953	1.000000

Figure 12: Rank correlation of the merged dataframe

You can see in Figure 12, the correlation is positive for the Weekly_Sales row compared with the Promotions. This indicates that people are more prone to purchase products and/or more products when promotions are on. Also you can see that there is a correlation between Unemployment and Temperature, this could mean workers take holidays when the temperature is sunny or people are more prone to be fired when the temperature's high.

The t-value measures the size of the difference relative to the variation in the sample data.

- T is simply the calculated difference represented in units of standard error. The greater the magnitude of T, the greater the evidence against the null hypothesis. The closer T is to 0, the more likely there is no significant difference Brown & Melamed (2012).
- The p-value is a number between 0 and 1 and interpreted in the following way: A small p-value (typically <= 0.05) indicates strong evidence against the null hypothesis, so you reject the null hypothesis.

The algorithm implemented for the t-test can be found in the Appendix. The analyst compared the weekly sales on holidays and when there are no holidays. The results were as follows: *Ttest_indResult (statistic=8.29475685393189O1)* and *pvalue=1.0912222677433l6e-16*. There is a significant difference between holiday sales and non-holiday sales given the results above.

7 Predictive model pre-imputation

The analyst decided to run a predictive regression model Olive (2017) pre-imputation making this a benchmark to compare with results post-imputation predictive analysis. The analyst used all the numeric data type variables in the training data to predict the weekly_sales test data. The results are as follows:

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.094
Model:                            OLS   Adj. R-squared:                  0.094
Method:                 Least Squares   F-statistic:                     4646.
Date:                Wed, 15 May 2019   Prob (F-statistic):               0.00
Time:                        11:54:10   Log-Likelihood:             8.5648e+05
No. Observations:              402475   AIC:                        -1.713e+06
Df Residuals:                  402465   BIC:                        -1.713e+06
Df Model:                           9
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1             0.0030      0.000      9.709      0.000       0.002       0.004
x2             0.0156      0.000     41.985      0.000       0.015       0.016
x3             0.0056      0.001      3.715      0.000       0.003       0.009
x4             0.0113      0.001     10.244      0.000       0.009       0.013
x5             0.0315      0.001     24.146      0.000       0.029       0.034
x6             0.0073      0.002      4.388      0.000       0.004       0.011
x7             0.0279      0.001     18.962      0.000       0.025       0.031
x8             0.0048      0.000     18.638      0.000       0.004       0.005
x9             0.0065      0.000     18.285      0.000       0.006       0.007
const          0.0080   9.68e-05     82.930      0.000       0.008       0.008
==============================================================================
Omnibus:                   327809.602   Durbin-Watson:                   2.000
Prob(Omnibus):                  0.000   Jarque-Bera (JB):         13650592.088
Skew:                           3.657   Prob(JB):                         0.00
Kurtosis:                      30.577   Cond. No.                         72.2
==============================================================================
```

Figure 13: OLS results pre-imputation

The R^2 value is extremely poor Nagelkerke (1991), this could be an indication that the lack of promotion data does have an affect on the quality of the predictive model.

The algorithm is presented in the Appendix.

8 Public holiday sales analysis

This chapter contains analysis of weekly sales data a week before public holidays and a week after.

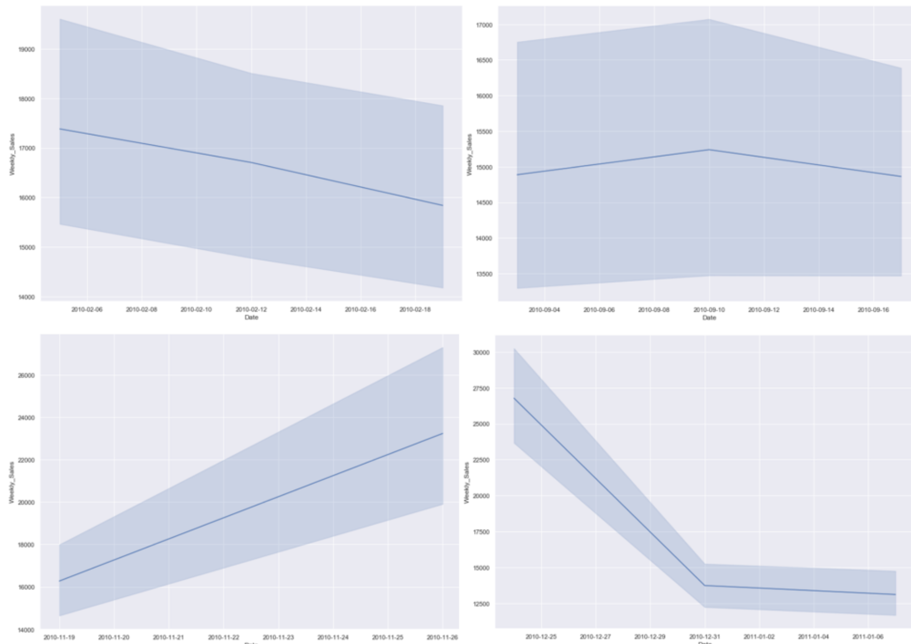

Figure 14: Holiday sales 2010 and 2012

In Figure 14, on the top left Super Bowl for 2010 weekly sales can be seen. Sales start to decrease and continue to decrease after the event. Top right 2010 Labor Day we can see a slight increase in sales and a decrease after the holiday. A very interesting graph is the Christmas 2010 holidays. We can see sales skyrocket a week before and gradually decrease till Christmas day and after Christmas gradually decreases. The holiday sales graphs for other years not included in Figure 14 can be found in the Appendix as these all showed similar trends.

9 Deep Learning Imputation

The analyst adopted an advanced predictive technique to impute missing data, imputation can fill in the missing data to later help us train more accurate regression/NN models on the dataset Little & Rubin (2015). One of the most advanced methods is deep learning imputation using **Datawig** python library.

The variables that are closely correlated to Promotions and CPI are weekly sales, temperature and fuel price. The analyst also calculated the R^2 score to see how accurate the imputed values are from the training data.

The final_df with the newly populated columns:

Promotion1_imputed	Promotion2_imputed	Promotion3_imputed	Promotion4_imputed	Promotion5_imputed	CPI_imputed	Weekly_Sales_imputed
-23.376474	-1607.985551	-353.826430	-301.255218	-80.106949	154.367388	5862.466556
2095.467932	1228.118108	688.213762	123.822650	554.509723	219.549703	19622.972464
278.288496	-1401.248725	317.925636	-157.032943	519.280497	139.631219	13679.830948
160.640149	-131.224040	-700.324773	36.995891	-18.198700	159.889929	12142.983804
249.207386	-3010.555324	26.838928	-10.333324	-2.116164	149.428467	59436.916811

Figure 15: five rows of imputed columns

The R^2 score for the new columns are:

- Promotion1: 0.8796632334818133
- Promotion2: 0.6603054641256494
- Promotion3: 0.4918742528293464
- Promotion4: 0.8915634612104615
- Promotion5: 0.5287041047093568
- CPI: 0.35543474289083
- Weekly_Sales: 0.5619225329783258

The results above are quite promising for the promotions and weekly sales columns but CPI will not be used in any predictive modelling.

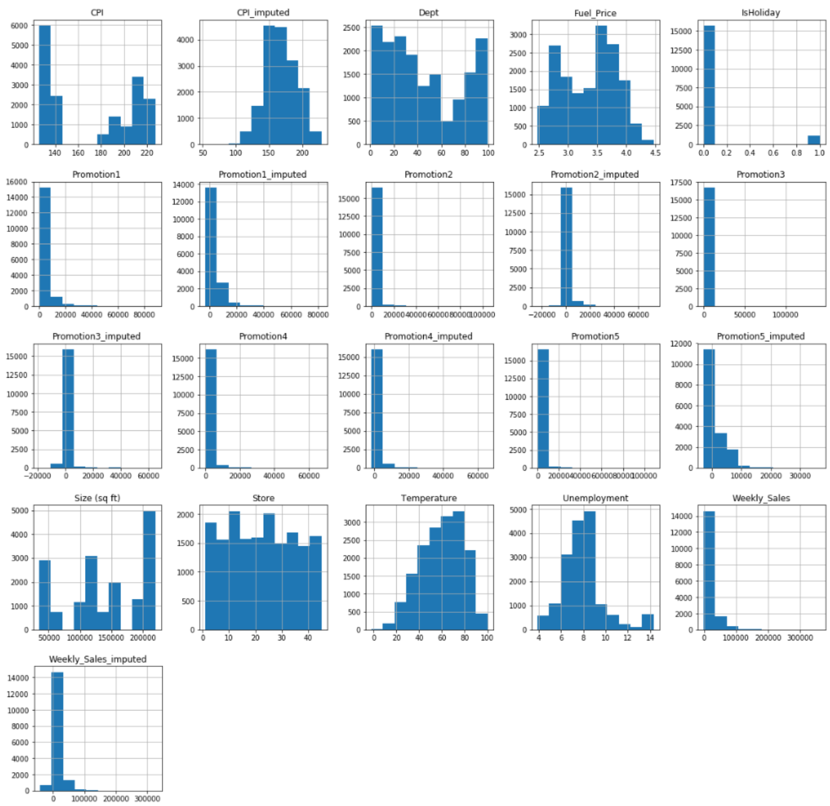

Figure 16: Histogram comparing old columns with imputed ones

In Figure 16, we can see imputed columns are quite similar to the pre-imputation columns. This could mean that the imputation process was a success. The algorithm used can be found in the Appendix.

10 Predictive model post-imputation

Now that missing values have been imputed, the analyst joins the test dataframe to predict the values for the **Weekly sales to be predicted** column using the imputed columns as training data (excluding CPI).

10.1 Linear regression

The analyst first performs linear regression (algorithm found in Appendix), the results are as follows:

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.958
Model:                            OLS   Adj. R-squared:                  0.958
Method:                 Least Squares   F-statistic:                 1.173e+05
Date:                Mon, 13 May 2019   Prob (F-statistic):               0.00
Time:                        20:02:12   Log-Likelihood:             3.0625e+05
No. Observations:               98944   AIC:                        -6.125e+05
Df Residuals:                   98924   BIC:                        -6.123e+05
Df Model:                          19
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1            -0.0002      0.000     -1.582      0.114      -0.000    4.55e-05
x2             0.0002      0.000      1.731      0.084   -2.56e-05       0.000
x3             0.5161      0.002    317.162      0.000       0.513       0.519
x4             0.0420      0.001     67.174      0.000       0.041       0.043
x5             0.0104      0.001     20.605      0.000       0.009       0.011
x6            -0.0200      0.005     -3.900      0.000      -0.030      -0.010
x7            -0.1047      0.003    -33.058      0.000      -0.111      -0.098
x8            -0.0371      0.003    -11.390      0.000      -0.043      -0.031
x9             0.0386      0.006      5.957      0.000       0.026       0.051
x10            0.0371      0.003     10.682      0.000       0.030       0.044
x11           -0.0050      0.000    -15.503      0.000      -0.006      -0.004
x12           -0.0691      0.007    -10.591      0.000      -0.082      -0.056
x13            0.2285      0.003     70.270      0.000       0.222       0.235
x14            0.0824      0.003     27.296      0.000       0.076       0.088
x15            0.0337      0.007      4.875      0.000       0.020       0.047
x16           -0.0049      0.002     -2.216      0.027      -0.009      -0.001
x17            0.0146      0.001     24.944      0.000       0.013       0.016
x18            0.0101      0.000     32.677      0.000       0.010       0.011
x19            0.0230      0.001     23.102      0.000       0.021       0.025
const               0          0        nan        nan           0           0
x20         7.551e-05    8.4e-05      0.899      0.369   -8.92e-05       0.000
==============================================================================
Omnibus:                    73499.657   Durbin-Watson:                   1.989
Prob(Omnibus):                  0.000   Jarque-Bera (JB):         49693832.712
Skew:                           2.315   Prob(JB):                         0.00
Kurtosis:                     112.692   Cond. No.                          inf
==============================================================================
```

Figure 17: OLS results post-imputation

In Figure 17, we can see that the coefficient of determination (R^2) is very high **0.95** meaning the populated dataframe with promotions is a very good training set to predict future values like 2013 sales forecasts.

10.2 Random Forest regression

As the analyst now has a training set, they decided to run a Random Forest regressor Cutler et al. (2012) as well just to see how well it performs and if it is better than linear regression. The metrics used are **Mean Absolute Error**, **Mean Squared Error** and **Root Mean Squared Error**. The results are as follows:

- Mean Absolute Error: 0.00554390305146
- Mean Squared Error: 0.000322561653259
- Root Mean Squared Error: 0.0179600014827

Given these values, Random Forest is definitely not a good model to use for this dataset.

The algorithm used can be found in the Appendix.

11 Weekly Sales on Department level

In this section the analyst will convey some of the newly graphed data after prediction. More specifically the weekly sales on departmental level will be presented:

Figure 18: Store and Department weekly sales

In Figure 18, the analyst presents the weekly sales predicted on departmental and store level.

12 Conclusion

One of the biggest problems the analyst came across was the lack of data. It was clear from the predictive model that promotions were valuable to the prediction of weekly sales. If the analyst had just dropped them or filled the missing cells with NaN's then this may have produced an ill-performing regression model. It would be highly desired to investigate time series analysis techniques for future forecasting, and to compare these techniques with imputation and regression.

References

Abzalov, M. (2016), Exploratory data analysis, *in* 'Modern Approaches in Solid Earth Sciences'.

Boskin, M. J., Dulberger, E. R., Gordon, R. J., Griliches, Z. & Jorgenson, D. W. (2011), 'Consumer Prices, the Consumer Price Index, and the Cost of Living', *Journal of Economic Perspectives* .

Brown, S. & Melamed, L. (2012), T Test, *in* 'Experimental Design and Analysis'.

Cutler, A., Cutler, D. R. & Stevens, J. R. (2012), Random forests, *in* 'Ensemble Machine Learning: Methods and Applications'.

Ganong, P., Noel, P., Campbell, J., Chetty, R., Chodorow-reich, G., Cutler, D., Glaeser, E., Hendren, N., Kekre, R., Levenson, A., Pallais, M., Parker, J., Plagborg-møller, M., Scharfstein, D., Shleifer, A., Shoag, D. & Stancheva, S. (2016), How Does Unemployment Affect Consumer Spending?

Little, R. J. & Rubin, D. B. (2015), Missing Data, *in* 'International Encyclopedia of the Social & Behavioral Sciences: Second Edition'.

Miller, G. A. & Chapman, J. P. (2001), 'Misunderstanding analysis of covariance', *Journal of Abnormal Psychology* .

Nagelkerke, N. J. D. (1991), 'A note on a general definition of the coefficient of determination'.

Olive, D. J. (2017), *Linear regression*.

Yoo, C., Park, J. & MacInnis, D. J. (1998), 'Effects of Store Characteristics and In-Store Emotional Experiences on Store Attitude', *Journal of Business Research* .

A Appendix

```
# Check if Null values exist
def null_checker(dataframe_column):
    return dataframe_column.isnull().sum().sum()
```

Figure 19: null_checker function

```
# Difference test: t-test for holidays and non holidays sales:
holiday_true = final_df[final_df['IsHoliday'] == True]
holiday_false = final_df[final_df['IsHoliday'] == False]

ttest_ind(holiday_true['Weekly_Sales'], holiday_false['Weekly_Sales'])
```

Figure 20: Difference t-test

```python
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import label
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

```python
regression_before_imputation = pd.concat([final_df, test_df])
```

```python
regression_before_imputation = regression_before_imputation.fillna(0)
```

```python
# Use min max scaler
scaler = MinMaxScaler()
DF_Scaled_1 = scaler.fit_transform(regression_before_imputation[columns])
DF_Scaled_1 = pd.DataFrame(data=DF_Scaled_1, columns=regression_before_imputation[columns].columns)
```

```python
# Make X and y
y = DF_Scaled_1['Weekly_Sales']
DF_Scaled_1 = DF_Scaled_1.drop('Weekly_Sales', axis = 1)
X = DF_Scaled_1
```

```python
#Reference Variable
DF_Scaled_1['_intercept'] = 1
```

```python
# split the dataset into the training set and test set
X_train, X_test, y_train, y_test = train_test_split(X,y, random_state=0)

logit = sm.OLS(np.array(y_train), np.array(X_train))

    # Fit the model
result = logit.fit()
```

```python
print(result.summary())
```

Figure 21: Regression pre-imputation algorithm

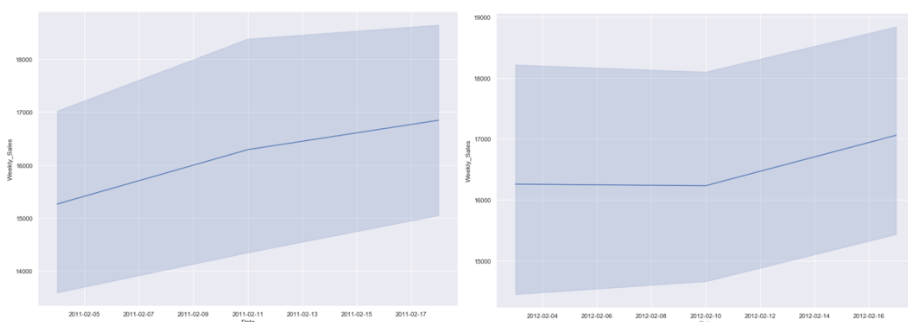

Figure 22: Holiday sales 2011 and 2012

```
# Libraries used
import datawig
import pandas as pd
from sklearn.metrics import r2_score as score

# After imputing final_df.csv we clean the data by rounding to 1 decimal place and filling the
 null values for
# the f2_score metric to not throw an error regarding float size or null values.
df = pd.read_csv('final_df.csv')
df = df.round(1)
df = df.fillna(0)

# Split the dataset into training and test data, 80% to %20
df_train, df_test = datawig.utils.random_split(df, split_ratios=[0.8, 0.2])

# Creating the SimpleImputer object, the input columns are the columns we believe are relevant
 in calculating the
# Promotions. Output_column is the column we are imputing the values for. The output_path is th
e output log.
imputer = datawig.SimpleImputer(
    input_columns = ['Weekly_Sales', 'Temperature', 'Fuel_Price',
    'Promotion5', 'Promotion4', 'CPI', 'Promotion2', 'Promotion3'],
    output_column = 'Promotion1',
    output_path = 'imputer_model'
)

# We fit the training data and state the number of epochs
imputer.fit(train_df=df_train, num_epochs = 50)

# The imputed dataframe is produced.
imputed = imputer.predict(df_test)

# We calculate the f2_score for the Promotion1 and the Imputed Promotion1
f1 = score(imputed['Promotion1'], imputed['Promotion1_imputed'])
print('Promotion1 f2_score: ', f1)
```

Figure 23: Deep Learning Imputation algorithm

```
from sklearn.ensemble import RandomForestRegressor

regressor = RandomForestRegressor(n_estimators=20, random_state=0)
regressor.fit(X_train, y_train)
y_pred = regressor.predict(X_test)
```

```
from sklearn import metrics

print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

Figure 24: Random Forest regression algorithm