Rossmann Store Sales Forecasting
A Machine Learning Algorithm Comparison

Georgios Panagiotakopoulos
Postgraduate Student
Department of Digital Systems, University of Piraeus
Information Systems and Services
Big Data and Analytics

Email: georgepanlos@gmail.com

1 Abstracrt

Accurate sales forecasting is considered an important tool, as it helps CEOs estimate demand for their products. The relevant paper forecasts sales for 1115 Rossmann stores. Apply Machine Learning algorithms for multiple time series. The data was obtained from Kaggle, are analyzed through the Python programming language and compared with the results of the Weka tool. Linear models work better than non-linear techniques. Specifically, the Elastic Net Method gives the best results. This result could improve sales forecasts for Rossmann stores.

2 Introduction

Rossmann has more than 3,000 drug stores in seven European countries. Rossmann store managers are currently required to provide their daily sales up to six weeks in advance. Store sales are influenced by many factors, such as supply, competition, school and government holidays, seasonality and location. With thousands of individual managers predicting sales based on their unique circumstances, the accuracy of the results may vary. The field of Machine Learning deals with the question of how to build computer programs that will automatically improve with experience. In recent years, many successful Machine Learning applications have been developed, from data mining programs that learn to detect fraudulent credit card, transactions, to automated information filtering systems that teach users reading preferences to autonomous vehicles that learn to drive passenger cars and various types of commercial vehicles. At the same time, significant developments in the theory and algorithms that form the basis of this field have taken place. For Machine Learning Algorithms see Hackeling's book[1]. For Methods applications see the books by Muller & Guido [2] and Aurelien Geron [3]. Finally, for a detailed use of the Weka tool, see Jason Brownlee's book [4].

3 Data and Variables

Analyze historical sales data for 1,115 Rossmann stores. The goal is to predict "Sales" and find the most appropriate method. Please note that some stores in the dataset have been temporarily closed for refurbishment. A total of 1115 time series are analyzed. More generally, we face a multi-time series prediction problem.

4 Exploratory Data Analysis

Figure 1 shows the frequencies for each day "Day of the Week" from the "Open" variable (closed and open). Each day is represented by a color and a number (Monday = 1,..., Sunday = 7). Mostly on closed days the operation (the category with the highest frequencies) is on Saturday and from the daily ones on Tuesday. On the other hand, the stores are open part-time on Sundays and almost as much on weekdays.

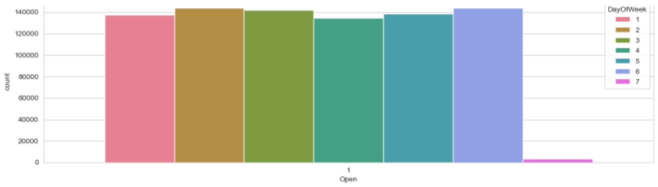


Figure 1: Bar chart for the days of the week "Open of the week" when they are open

In Figure 2, Time Plots for average and mean percentage change of sales over time are presented in Figure 2. Both time series do not seem to have any trend and therefore are stationary

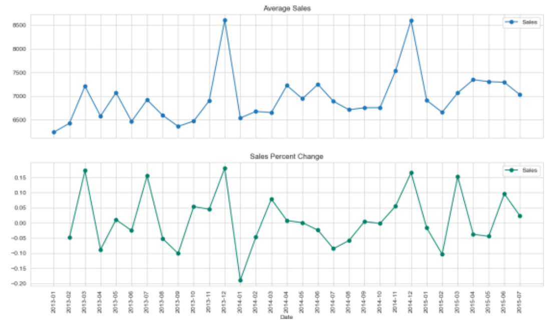


Figure 2: Time Plot for Average Sales and Percentage Change of the Sales over Time (year-month)

In Figure 3, we observe that the average sales and the average number of customers do not seem do differ significantly over the three years 2013, 2014, and 2015.

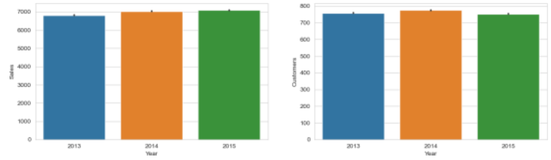


Figure 3: Bar Charts for Average Sales and Customers per year

In Figure 4, we graph a time plot for the average number of customers over time. The time series does not have any trend and is stationary. The distribution of that variable is very skewed to the right according to the boxplot.

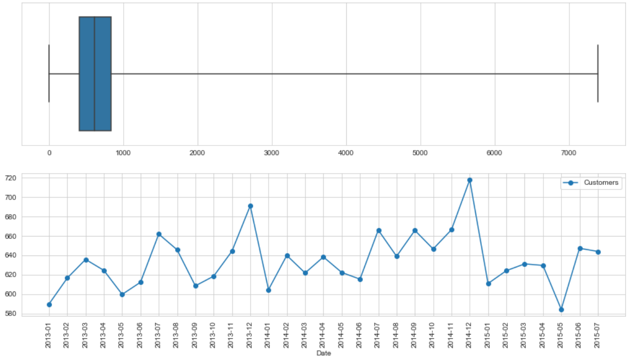


Figure 4: Time Plot and Box-plot for the Average Number of Customers over Time (year-month)

In Figure 5, we observe that the average sales and the average number of customers seem to differ significantly over the days of the week and logically the follow the same pattern. We note that on Mondays we have the highest frequencies (at the beginning of the week), then on Fridays (at the end of the week), from Mondays to Saturdays but Fridays the frequencies decrease.

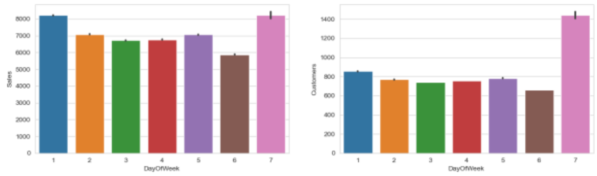


Figure 5: Bar Charts for Average Sales and Customers per Day of the Week

In Figure 6, we observe that the average sales and the average number of customers seem to differ significantly over the variable Promo. The promotion of the stores definitely decreases the sales and the average number of customers.

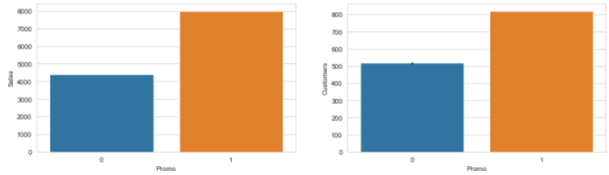


Figure 6: Bar Charts for Average Sales and Customers per Promo

In Figure 7, we graph a box-plot and a histogram for the average sales. The distribution of that variable Sales is very skewed to the right according to the boxplot and to the histogram, and there are many zeros (closed days) for this variable.

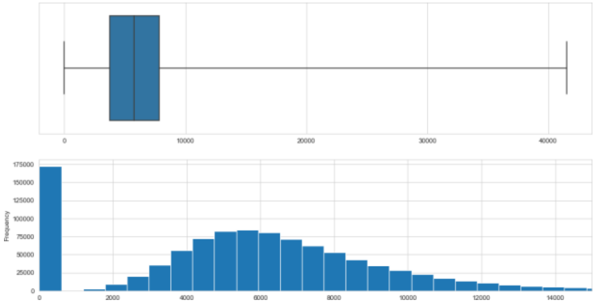


Figure 7: Box-plot and Histogram for the Average Sales

In Figure 8, we observe that the variable Store Type seems to differ significantly over the four different store models (a, b, c, d). Type a has the highest frequency and type b has the lowest frequency

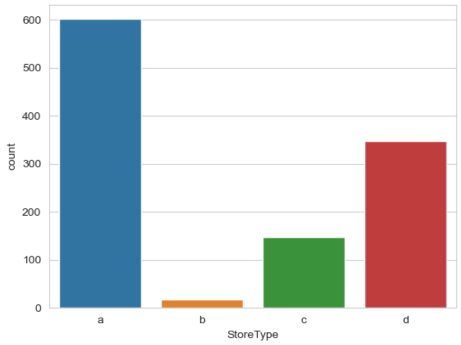


Figure 8: Bar Chart for Store Type

In Figure 9, we observe that the average sales and the average number of customers differ in relation to the four types of stores. Notice more direct sales and number of customers appearing in type b. Types a, c and d, have about the same sales, types c and d have about the same number of customers and type a has the lowest frequency for the average number of customers.

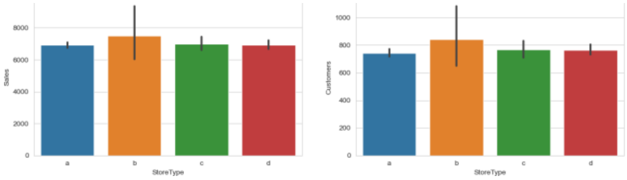


Figure 9: Bar Charts for Average Sales and Customers per Score Type

In Figure 10, we observe that the variable Assortment seems to differ significantly over the three different assortment level (a, b, c). Type a has the highest frequency then type c and type b has a very small frequency.

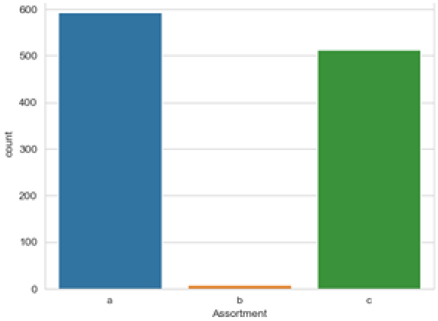


Figure 10: Bar Chart for Assortment

In Figure 11, we observe that the average sales and the average number of customers differ significantly in relation to the three levels of assortment. We observe more average sales, and the number of customers is displayed at level b. Level c has almost the same frequencies as level a for sales and in terms of customer levels, level a has a lower frequency than c.

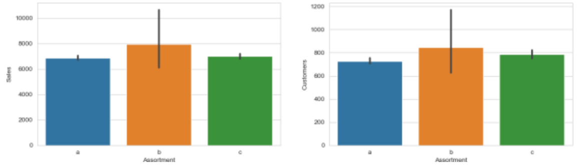


Figure 11: Average sales on each assortment level

In Figure 12, we observe that the Promo2 variable gives a bit higher frequency when a store is participating in a continuing and consecutive promotion.

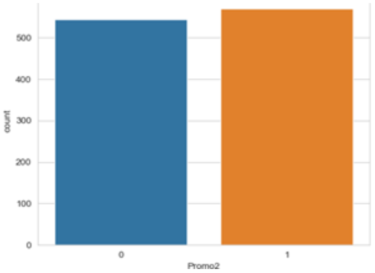


Figure 12: Bar Chart for Promo2

In Figure 13, on bar charts we observe that the average sales and the average number of customers differ significantly over the two categories for the variable Promo2. When a store is participating in a continuing and consecutive promotion has less sales and customers on the average.

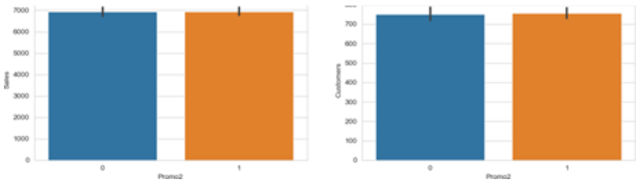


Figure 13: Bar Charts for Average Sales and Customers per Promo2

In Figure 14, we observe that there is no strong relationship between Sales and CompetitionDistance (distance in meters to the nearest competitor store).

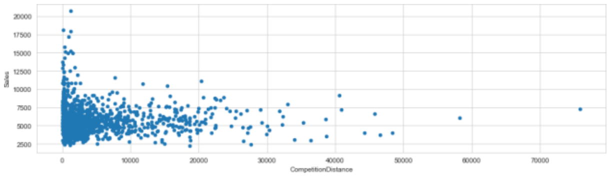


Figure 14: Scatterplot between Sales and CompetitionDistance

Figure 15 shows the density in terms of number of sales. As the density of competition in a particular area increases, so do the sales. Therefore, from the above diagram, we lead to the conclusion that the specific areas are densely populated.

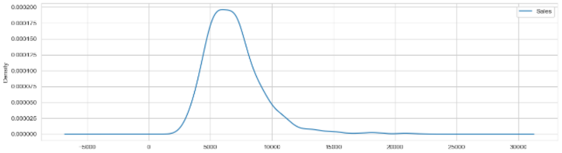


Figure 15: Sales Density

In Figure 16, we compute the correlation coefficients for the first five stores for the Sales. We note that there are high positive correlations over time between two different stores.

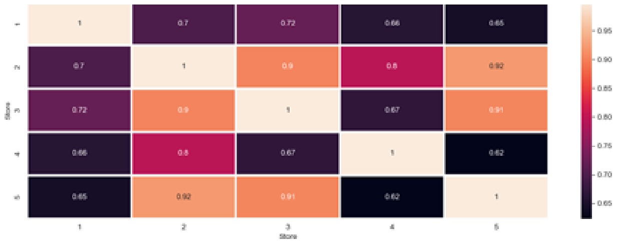


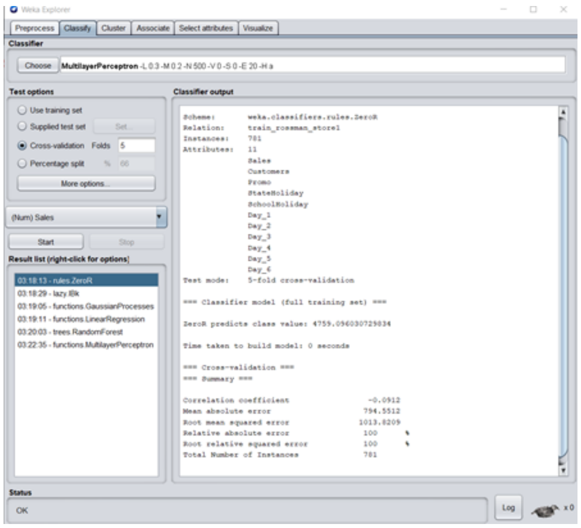
Figure 16: Correlation Coefficients between the First Five Stores for the Sales

5 Data classification and Model comparisons

Data classification is a form of data analysis that can be used to generate models from the training set, which describes important categories of data, or to predict future trends by finding the degree of accuracy. If the accuracy is considered acceptable, the rules can be applied to the classification of new sets of data. We therefore used 6 classification algorithms through the WEKA tool to predict their actions.

We compare six Machine Learning Methods, listed in Tables 1 and 2, to predict the Sales Variable in the test set. Train (70% of the first observations from each chronological order) and test set (30% of the last observations from each chronological order). The Root Mean Squared Error (RMSE) uses a standard criterion for predicting regression as a measurement.

We use the Weka software, which is a Data Mining and Machine Learning toolbox. It contains tools for data preparation, classification, regression, clustering, association rules mining, and visualization. Also, it's an open source Software.



Classifier: MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

Folds: 5

Percentage split %: 66

(Num) Sales

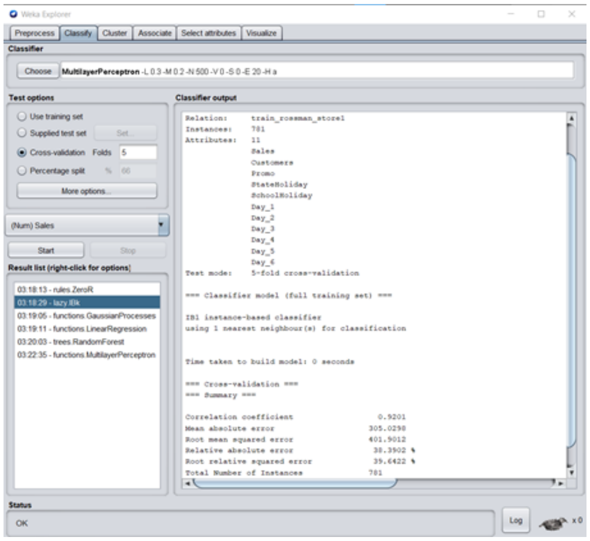
Result list:

- 03:18:13 - rules.ZeroR
- 03:18:29 - lazy.IBk
- 03:19:05 - functions.GaussianProcesses
- 03:19:11 - functions.LinearRegression
- 03:20:03 - trees.RandomForest
- 03:22:35 - functions.MultilayerPerceptron

```
Scheme:       weka.classifiers.rules.ZeroR
Relation:     train_rossman_store1
Instances:    781
Attributes:   11
              Sales
              Customers
              Promo
              StateHoliday
              SchoolHoliday
              Day_1
              Day_2
              Day_3
              Day_4
              Day_5
              Day_6
Test mode:    5-fold cross-validation

=== Classifier model (full training set) ===

ZeroR predicts class value: 4759.096030729834

Time taken to build model: 0 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  -0.0912
Mean absolute error                     794.5512
Root mean squared error                1013.8209
Relative absolute error                  100      %
Root relative squared error              100      %
Total Number of Instances                781
```

Status: OK

Figure 17: ZeroR using 5-fold-Cross-Validation



Classifier: MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

Folds: 5

Percentage split %: 66

(Num) Sales

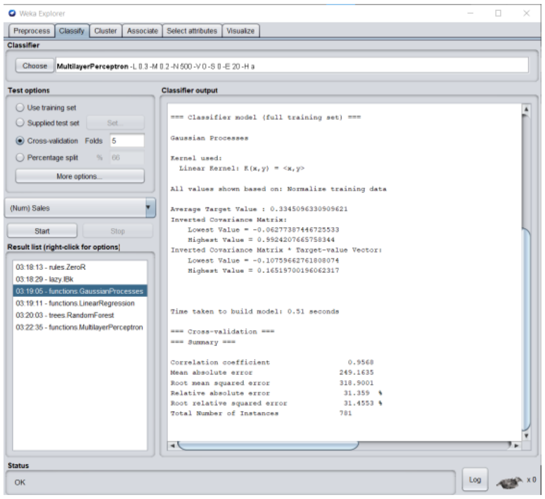
Result list:

- 03:18:13 - rules.ZeroR
- 03:18:29 - lazy.IBk
- 03:19:05 - functions.GaussianProcesses
- 03:19:11 - functions.LinearRegression
- 03:20:03 - trees.RandomForest
- 03:22:35 - functions.MultilayerPerceptron

```
Relation:     train_rossman_store1
Instances:    781
Attributes:   11
              Sales
              Customers
              Promo
              StateHoliday
              SchoolHoliday
              Day_1
              Day_2
              Day_3
              Day_4
              Day_5
              Day_6
Test mode:    5-fold cross-validation

=== Classifier model (full training set) ===

IB1 instance-based classifier
using 1 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                    0.9201
Mean absolute error                      305.0298
Root mean squared error                  401.9012
Relative absolute error                   38.3902 %
Root relative squared error               39.6422 %
Total Number of Instances                781
```

Status: OK

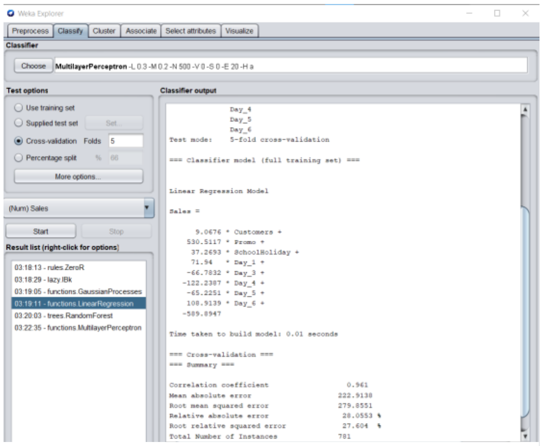
Figure 18: k-Nearest Neighbor using 5-fold-cross-validation.



```
MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
○ Use training set
○ Supplied test set
● Cross-validation  Folds 5
○ Percentage split  % 66
(Num) Sales
03:18:13 - rules.ZeroR
03:18:29 - lazy.IBk
03:19:05 - functions.GaussianProcesses
03:19:11 - functions.LinearRegression
03:20:03 - trees.RandomForest
03:22:35 - functions.MultilayerPerceptron
```

```
=== Classifier model (full training set) ===

Gaussian Processes

Kernel used:
  Linear Kernel: K(x,y) = <x,y>

All values shown based on: Normalize training data

Average Target Value : 0.3345096330909621
Inverted Covariance Matrix:
    Lowest Value = -0.06277387446725533
    Highest Value = 0.9924207665758344
Inverted Covariance Matrix * Target-value Vector:
    Lowest Value = -0.10759662761808074
    Highest Value = 0.16519700196062317



Time taken to build model: 0.51 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.9568
Mean absolute error                    249.1635
Root mean squared error                318.9001
Relative absolute error                 31.359  %
Root relative squared error             31.4553 %
Total Number of Instances              781
```

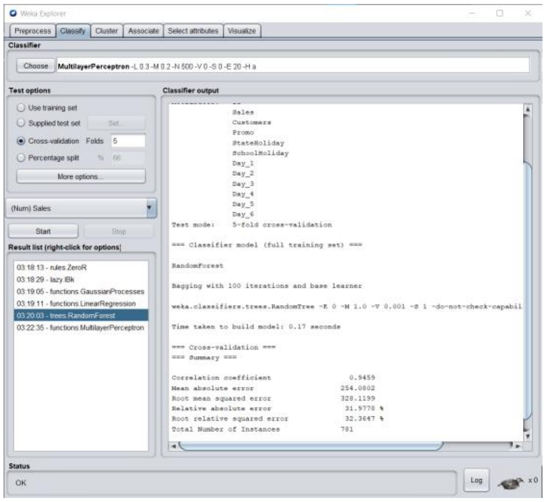
Figure 19: Gaussian Processes using 5-fold-Cross-Validation.



```
MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
○ Use training set
○ Supplied test set
● Cross-validation  Folds 5
○ Percentage split  % 66
(Num) Sales
03:18:13 - rules.ZeroR
03:18:29 - lazy.IBk
03:19:05 - functions.GaussianProcesses
03:19:11 - functions.LinearRegression
03:20:03 - trees.RandomForest
03:22:35 - functions.MultilayerPerceptron
```

```
              Day_4
              Day_5
              Day_6
Test mode:    5-fold cross-validation

=== Classifier model (full training set) ===


Linear Regression Model

Sales =

      9.0676 * Customers +
    530.5117 * Promo +
     37.2693 * SchoolHoliday +
     71.94   * Day_1 +
    -66.7832 * Day_3 +
   -122.2387 * Day_4 +
    -65.2251 * Day_5 +
    108.9139 * Day_6 +
   -589.8947

Time taken to build model: 0.01 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.961
Mean absolute error                    222.9138
Root mean squared error                279.8551
Relative absolute error                 28.0553 %
Root relative squared error             27.604  %
Total Number of Instances              781
```

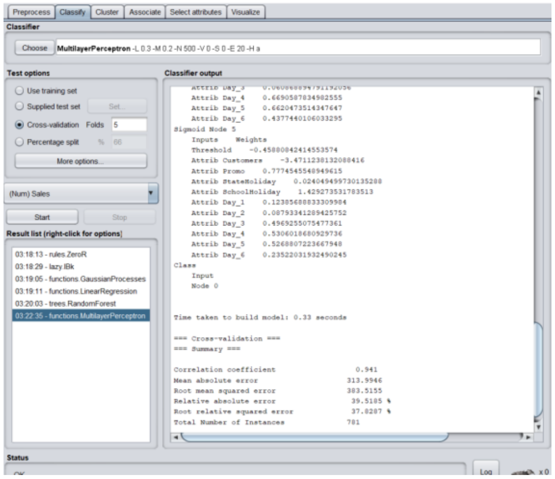
Figure 20: Linear Regression using 5-fold-Cross-Validation



```
MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
○ Use training set
○ Supplied test set
● Cross-validation  Folds 5
○ Percentage split  % 66
(Num) Sales
03:18:13 - rules.ZeroR
03:18:29 - lazy.IBk
03:19:05 - functions.GaussianProcesses
03:19:11 - functions.LinearRegression
03:20:03 - trees.RandomForest
03:22:35 - functions.MultilayerPerceptron
```

```
              Sales
              Customers
              Promo
              StateHoliday
              SchoolHoliday
              Day_1
              Day_2
              Day_3
              Day_4
              Day_5
              Day_6
Test mode:    5-fold cross-validation

=== Classifier model (full training set) ===

RandomForest

Bagging with 100 iterations and base learner

weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -do-not-check-capabil

Time taken to build model: 0.17 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.9459
Mean absolute error                    254.0802
Root mean squared error                328.1199
Relative absolute error                 31.9778 %
Root relative squared error             32.3647 %
Total Number of Instances              781
```

Figure 21: Random Forest using 5-fold-cross-validation



```
MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
○ Use training set
○ Supplied test set
● Cross-validation  Folds 5
○ Percentage split  % 66
(Num) Sales
03:18:13 - rules.ZeroR
03:18:29 - lazy.IBk
03:19:05 - functions.GaussianProcesses
03:19:11 - functions.LinearRegression
03:20:03 - trees.RandomForest
03:22:35 - functions.MultilayerPerceptron
```

```
    Attrib Day_4    0.6690587834982555
    Attrib Day_5    0.6620473514347647
    Attrib Day_6    0.4377440106033295
Sigmoid Node 5
    Inputs    Weights
    Threshold    -0.45880842414553574
    Attrib Customers    -3.4711238132088416
    Attrib Promo    0.7774545548949615
    Attrib StateHoliday    0.024049499730135288
    Attrib SchoolHoliday    1.429273531783513
    Attrib Day_1    0.12385688833309984
    Attrib Day_2    0.08793341289425752
    Attrib Day_3    0.4969255075477361
    Attrib Day_4    0.5306018680929736
    Attrib Day_5    0.5268807223667948
    Attrib Day_6    0.23522031932490245
Class
    Input
    Node 0


Time taken to build model: 0.33 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.941
Mean absolute error                    313.9946
Root mean squared error                383.5155
Relative absolute error                 39.5185 %
Root relative squared error             37.8287 %
Total Number of Instances              781
```

Figure 22: Neural Networks using 5-fold-cross-validation

Comparing the six RMSE-based methods, we conclude that the Elastic Net Method gives the best forecast results. The Lasso, Ridge and Linear Regression Method follow very closely and give almost the same results. Then follow from a relatively short distance the Gradient Boosting Regressor Method which gives similar results. The last Method The longest distance from the other Methods is the Random Forest Regressor Method. Random Forest is a very popular and successful method, but in this application, it seems that it fails to produce satisfactory results.

Based on the results given in the tables below, we conclude that the relationship between Sales and independent variables are linear and this is probably because the non-linear models like Gradient Boosting Regressor and Random Forest Regressor methods do not give results.

6 Tables - Classifiers comparison

Table 1: RMSE of Six Methods for the Test Set

	Python Methods	Root Mean Squared Error
1	Elastic Net	436.00
2	Lasso	438.46
3	Ridge	438.53
4	Linear Regression	438.54
5	Gradient Boosting Regressor	477.30
6	Random Forest Regressor	510.30

Table 2: RMSE using 5-fold Cross-Validation Methodology for store 1 via WEKA

	Weka Methods	Root Mean Square Error
1	Linear Regression	279.86
2	Gaussian Processes	318.90
3	Random Forest	328.12
4	Neural Networks	383.52
5	k-Nearest Neighbor	401.90
6	ZeroR rule	1013.82

7 Conclusions

We conclude that the sales of the Rossmann stores can be satisfactorily predicted by Machine Learning Methods. Graphic models such as Elastic Net, Lasso, and Ridge give even more satisfactory results than the non-linear methods such us Gaussian Processes, Gradient Boosting, Random Forest, Neural Networks, k-Nearest Neighbor. The results can be improved in the future with feature engineering methods [5] and more sophisticated methods, e.g., XGBoost [6], Prophet [7], and ensemble methods. More lags variables can be used as characteristics. Also smoothing methods, clustering, weighting, outliers may be effective in reducing sales errors.

Citations

[1] Gavin Hackeling, Mastering Machine Learning with scikit learn. Birmingham,UK: Packt Publishing, 2017.

[2] Andreas C. Muller, Sarah Guido, Introduction to Machine Learning with Python. 1005 Gravenstein Highway North,Sebastopol: O'Reilly, 2017.

[3] Aurelien Geron, Hands-On Machine Learning with Scikit-Learn,Keras & TensorFlow. 1005 Gravenstein Highway North,Sebastopol: O'Reilly, 2019.

[4] Jason Brownlee, Machine Learning Mastery With Weka. Jason Brownlee, 2017.

[5] Keshav Rawat, 'Rossmann Store Sales Prediction.' https://medium.com/analytics-vidhya/rossmann-store-sales-prediction-998161027abf#251f.

[6] Anton Lebedevich, 'My Top 10% Solution for Kaggle Rossman Store Sales Forecasting Competition.' https://mabrek.github.io/blog/kaggle-forecasting/.

[7] Matt Evanoff, 'Improving Sales with Analytics.' https://medium.com/@mevanoff24/ store-sales-analysis-c7a5a0bbaaa0.