EFFECT OF WEATHER ON RETAIL SALES

Thesis

Submitted By

13-23318-1	Saha Pratik
13-23204-1	Sarker, Sowrozit
13-23283-1	Akhter, Rubina
12-21878-2	Mostabi, Nabila

Department of Computer Science

Faculty of Science & IT

American International University Bangladesh

May, 2017

Declaration

We declare that this thesis is our original work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

_____

Saha, Pratik
13-23318-1
CSSE

_____

Sarker Sowrozit
13-23204-1
CSE

_____

Akhter Rubina
13-23283-1
CSE

_____

Mostabi Nabila
12-21878-2
CSE

Approval

The thesis titled **"EFFECT OF WEATHER ON RETAIL SALES"** has been submitted to the following respected members of the board of examiners of the department of computer science in partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Science on (**12 May,2017**) and has been accepted as satisfactory.

Shovra Das
Assistant Professor & Supervisor
Department of Computer Science
American International University-Bangladesh

Asif Ur Rahman
Assistant Professor & External
Department of Computer Science
American International University-Bangladesh

Dr. Dip Nandi
Associate Professor & Head(Undergraduate)
Department of Computer Science
American International University-Bangladesh

Professor Dr. Tafazzal Hossain
Dean
Faculty of Science & Information Technology
American International University-Bangladesh

Dr. Carmen Z. Lamagna
Vice Chancellor
American International University-Bangladesh

TO OUR PARENTS

ACKNOWLEDGEMENT

All praises to the most gracious and most merciful, the Almighty Allah who bestowed upon us the will for the successful completion of our thesis work within the scheduled time.

We would like to express our heartfelt gratitude and indebtedness to the thesis supervisor Shovra Das, Assistant Professor, Faculty of Science and Information Technology, American International University- Bangladesh, whose encouragement, continuous guidance, valuable suggestions, cooperation and cordial support from the initial to the final level to enable us complete the thesis successfully. His advice, initiative, moral support and patience are very gratefully acknowledged.

And also we are thankful to our department and our respected faculty members for their support and co-operation.

ABSTRACT

Monthly fluctuations in consumer spending are often attributed to the weather. This paper presents a model in which weather affects the productivity of time in nonmarket activities such as shopping or recreation and so, via time and budget constraints, may induce substitution in spending across goods and over time. Using monthly data on retail sales and weather data from the National Weather Service, we have found that unusual weather has a modest but significant role in explaining monthly sales fluctuations. However, lagged effects often offset original effects, so that weather's influence tends to wash out at a quarterly frequency.

TABLE OF CONTENTS

CHAPTER 2: Time Series Analysis

CHAPTER 3: TRAINING ASSESSMENT

CHAPTER 4: Result & Comparison

CHAPTER 5: CONCLUSION

LIST OF FIGURES

CHAPTER 1

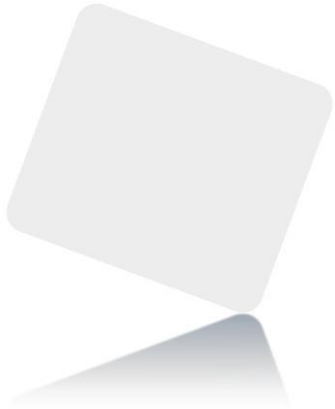
CHAPTER 2

CHAPTER 3

CHAPTER 4

LIST OF TABLES

CHAPTER 2



INTRODUCTION

1.1 Introduction:

Weather is often identified as the cause of month-to-month fluctuations in consumer spending. This is not a matter of regular seasonal variations, but rather exaggerations and departures from the seasonal cycle. Press reports on retail sales fluctuations during 1997 serve to illustrate the point. The unusually mild January 1997 was said to have brought consumers out to the stores and auto dealers' lots; a cool rainy spring kept them away from malls and restaurants; the return of seasonal weather in June sent them out to buy bathing suits and mountain bikes; and the unusually mild autumn delayed sales of cool weather attire.' Such arguments not only appear in the business press, but also figure into well-regarded macroeconomic forecasts: for example, both DRI and Macroeconomic Advisors predicted sizable drop backs in consumer spending in the second quarter of 1997, after the first-quarter binge.

Despite the frequency of arguments like this, the effects of weather on consumer spending have received little serious attention in economic research. Some micro studies have investigated effects of weather on sales for specific stores or in specific locations: for example, a notable early study found that sales of New York City department stores dropped off on rainy days (Linden 1959). While such results suggest that widespread deviations from normal weather could affect aggregate spending, to date there has been no systematic analysis of weather effects in the national data. Understanding the effects of weather is important for economic forecasters and monetary policy, because it helps distinguish changes in the underlying pace of economic activity from transitory shifts. Also, examining this issue may provide some interesting insights into intertemporal variations in consumption: for example, conceivably, the well-known lagged.

1.2 Background:

1.2.1 What is Forecasting?

Forecasting means future prediction of something after analyzing and observing previous history and present data. This assumption is based on experience and knowledge that grows with the time observing the effects of other thing on something particular.

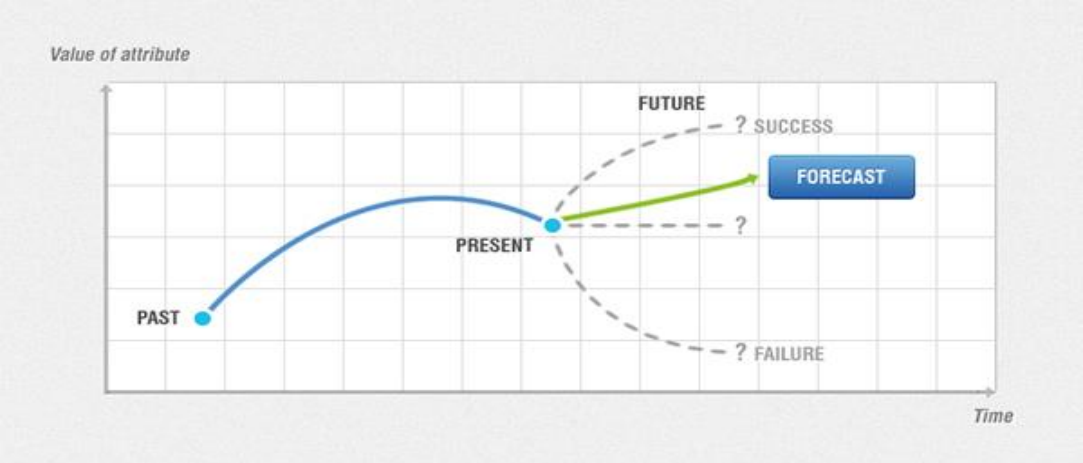


Figure 1.1 Forecasting

1.2.2 Weather Forecast:

This is something that everyone relies on every day. Weather forecast means prediction of weather. One will be able to know that how the weather is going to be in the next 10 days. Is it going to rain? Would he go outside with an umbrella for heavy rain or heavy sunshine? To answer these questions, many people have tried to predict weather applying many methods like **Time Series**, ANN etc. Still it is not so easy to understand the mood of nature. Still people of many corners trying to invent new approaches that will reveal the secrets of weather.

1.2.3 Retail Store Sales Forecast:

Retail store forecast means predicting the quantity of a product that a retail store should have in store at a particular period of time. It is like predicting the quantity of a product that needs to be in the inventory based on sales, weather and other relative impactors.

1.2.4 Regression Analysis Algorithm:

Regression is a data mining function that predicts a number. Age, weight, distance, temperature, income, or sales could all be predicted using regression techniques. For example, a regression model could be used to predict children's height, given their age, weight, and other factors.

Regression modeling has many applications in trend analysis, business planning, marketing, financial forecasting, time series prediction, biomedical and drug response modeling, and environmental modeling.

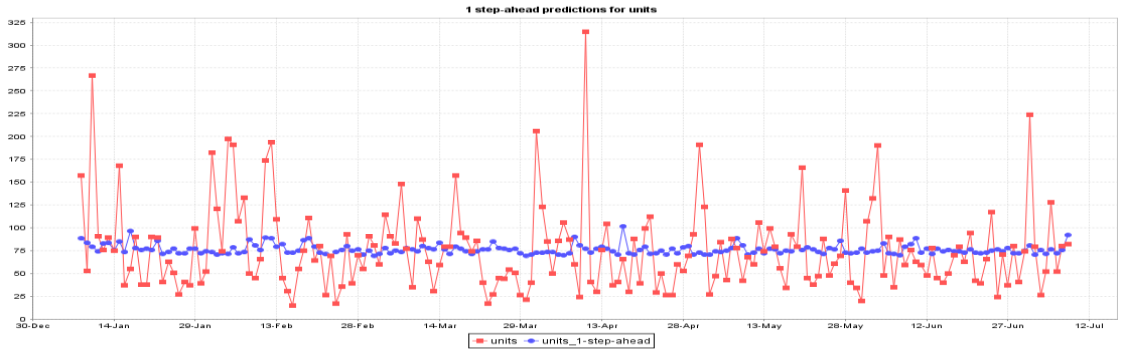


Figure 2.2 Regression analysis

1.2.4.1 Common Applications of Regression

Regression modeling has many applications in trend analysis, business planning, marketing, financial forecasting, time series prediction, biomedical and drug response modeling, and environmental modeling.

1.2.4.2 Testing a Regression Model

The Root Mean Squared Error and the Mean Absolute Error are statistics for

1.2.4.2.1 Root Mean Squared Error

The Root Mean Squared Error (RMSE) is the square root of the average squared distance of a data point from the fitted line.

1.2.4.2.2 Mean Absolute Error

The Mean Absolute Error (MAE) is the average of the absolute value of the residuals. The MAE is very similar to the RMSE but is less sensitive to large errors.

The relative measures give an indication of how the well forecaster's predictions are doing compared to just using the last known target value as the prediction. They are expressed as a percentage, and lower values indicate that the forecasted values are better predictions than just using the last known target value.

A score of >=100 indicates that the forecaster is doing no better (or even worse) than predicting the last known target value. Note that the last known target value is relative to the step at which the forecast is being made - e.g. a 12-step-ahead prediction is compared relative to using the target value 12 time steps prior as the prediction (since this is the last "known" actual target value).

1.2.5 What is Time Series Analysis?

Time series analysis is the process of using statistical techniques to model and explain a time-dependent series of data points. Time series forecasting is the process of using a model to generate predictions (forecasts) for future events based on known past events. Time series data has a natural temporal ordering - this differs from typical data mining/machine learning applications where each data point is an independent example of the concept to be learned, and the ordering of data points within a data set does not matter. Examples of time series applications include capacity planning, inventory replenishment, sales forecasting and future staffing levels.

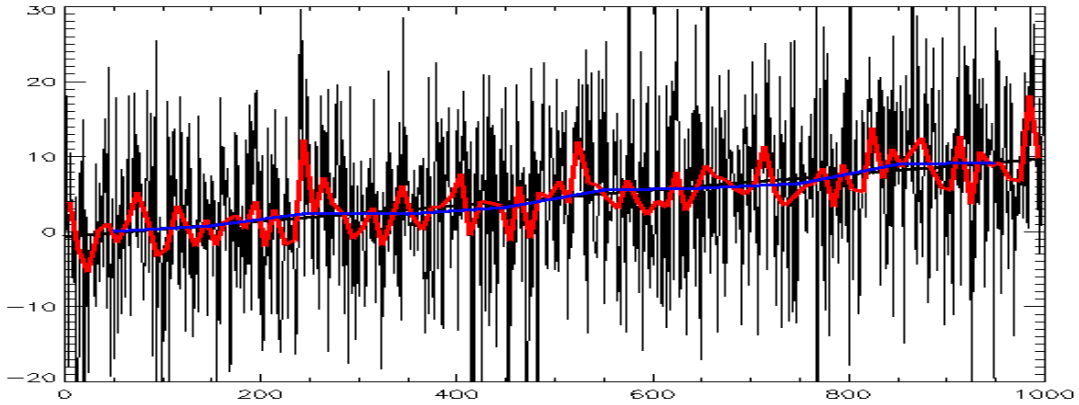


Figure 3.3 Time Series Analysis

1.3 Report Organization

Total five chapters are included in this report. The First chapter presents introductory part. The second chapter discusses about the Forecasting, Linear Regression & Time Series Analysis. Third chapter consists of training assessment. The Fourth chapter describes the result and comparison. Chapter Five consists of Supplementary part.

1.4 Challenges and Motivations

There are many challenges in the retail store network planning some of them are retailers fail in the evaluation of the potential of the market. Retailers ignore the seasonal randomness. The supply chain inefficiencies when the products have great demand then they are not available. The human resources are inefficient the employees are not available whenever necessary. The retailers face the difficulties in inventory management system; sometimes the retailers ignore the competition in the market. Retailers develop the plans that promotes the success and the highly target plan. The plans should be such that they help to obtain the maximum profit. The new product lines should be developed or they should be purchased with confidence. The supply chain mechanism should be efficient.

TIME SERIES ANALYSIS

2.1 Definition of Time Series Analysis:

Time series analysis is the process of using statistical techniques to model and explain a time-dependent series of data points. Time series forecasting is the process of using a model to generate predictions (forecasts) for future events based on known past events. Time series data has a natural temporal ordering - this differs from typical data mining/machine learning applications where each data point is an independent example of the concept to be learned, and the ordering of data points within a data set does not matter. Examples of time series applications include: capacity planning, inventory replenishment, sales forecasting and future staffing levels.
Weka (>= 3.7.3) now has a dedicated time series analysis environment that allows forecasting models to be developed, evaluated and visualized. This environment takes the form of a plugin tab in Weka's graphical "Explorer" user interface and can be installed via the package manager. Weka's time series framework takes a machine learning/data mining approach to modeling time series by transforming the data into a form that standard propositional learning algorithms can process. It does this by removing the temporal ordering of individual input examples by encoding the time dependency via additional input fields. These fields are sometimes referred to as "lagged" variables. Various other fields are also computed automatically to allow the algorithms to model trends and seasonality. After the data has been transformed, any of Weka's regression algorithms can be applied to learn a model. An obvious choice is to apply multiple linear regression, but any method capable of predicting a continuous target can be applied - including powerful non-linear methods such as support vector machines for regression and model trees (decision trees with linear regression functions at the leaves). This approach to time series analysis and forecasting is often more powerful and more flexible that classical statistical techniques such as ARMA and ARIMA.
The above mentioned "core" time series modeling environment is available as open-source free software in the CE version of Weka. The same functionality has also been wrapped in a Spoon Perspective plugin that allows users of Pentaho Data Integration (PDI) to work with time series analysis within the Spoon PDI GUI. There is also a plugin step for PDI that allows models that have been exported from the time series modeling environment to be loaded and used to make future forecasts as part of an ETL transformation. The perspective and step plugins for PDI are part of the enterprise edition.

2.2 Using the Time Series Environment

Once installed via the package manager, the time series modeling environment can be found in a new tab in Weka's Explorer GUI. Data is brought into the environment in the normal manner by loading from a file, URL or database via the Preprocess panel of the Explorer. The environment has both basic and advanced configuration options. These are described in the following sections.

2.2.1 Basic Configuration

The basic configuration panel is shown in the screenshot below

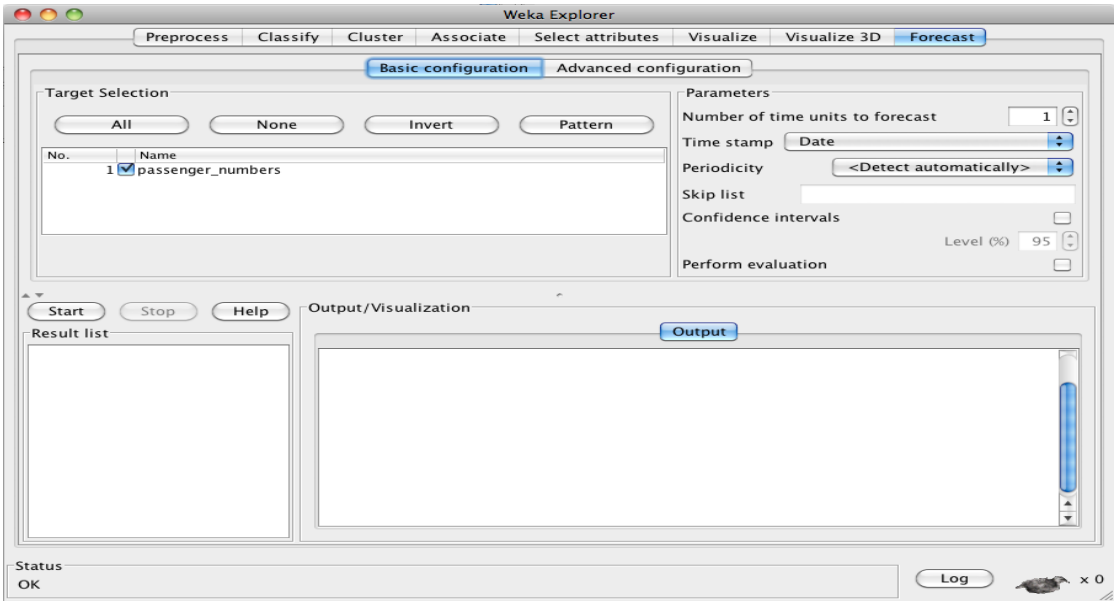


Figure 2.1: Basic Configuration of Time Series Environment in WEKA

In this example, the sample data set "airline" (included in the package) has been loaded into the Explorer. This data is a publicly available benchmark data set that has one series of data: monthly passenger numbers for an airline for the years 1949 - 1960. Aside from the passenger numbers, the data also includes a date time stamp. The basic configuration panel automatically selects the single target series and the "Date" time stamp field. In the Parameters section of the GUI (top right-hand side), the user can enter the number of time steps to forecast beyond the end of the supplied data. Below the time stamp drop-down box, there is a drop-down box for specifying the periodicity of the data. If the data has a time stamp, and the time stamp is a date, then the system can automatically detect the periodicity of the data. Below this there check boxes that allow the user to opt to have the system compute confidence intervals for its predictions and perform an evaluation of performance on the training data. More details of all these options are given in subsequent sections.

The following screenshot shows the results of forecasting 24 months beyond the end of the data.

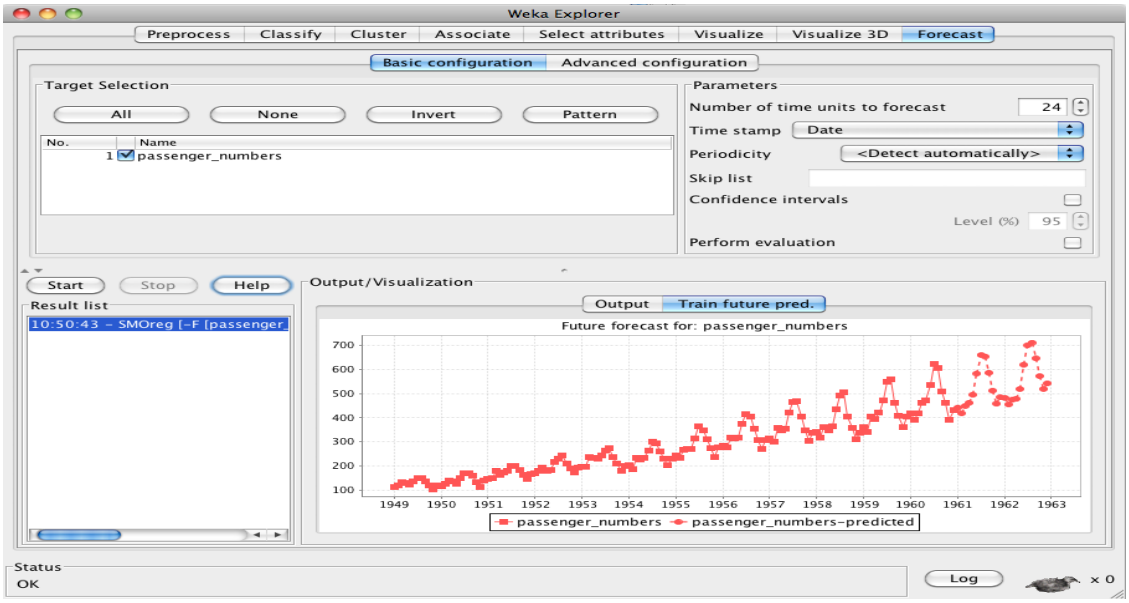


Figure 2.2: Basic Configuration of Time Series Environment in WEKA

2.2.1.1 Target selection

At the top left of the basic configuration panel is an area that allows the user to select which target field(s) in the data they wish to forecast. The system can jointly model multiple target fields simultaneously in order to capture dependencies between them. Because of this, modeling several series simultaneously can give different results for each series than modeling them individually. When there is only a single target in the data then the system selects it automatically. In the situation where there are potentially multiple targets the user must select them manually. The screenshot below shows some results on another benchmark data set. In this case the data is monthly sales (in liters per month) of Australian wines. There are six categories of wine in the data, and sales were recorded on a monthly basis from the beginning of 1980 through to the middle of 1995. Forecasting has modeled two series simultaneously: "Fortified" and "Dry-white".

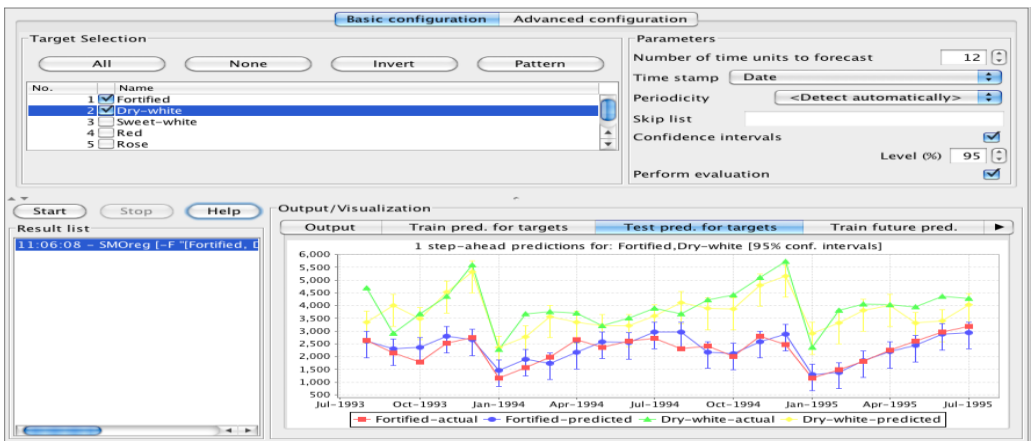


Figure 2.3: Target Selection in Time Series Environment in WEKA

2.2.1.2 Basic parameters

At the top right of the basic configuration panel is an area with several simple parameters that control the behavior of the forecasting algorithm.

2.2.1.2.1 Number of time units

The first, and most important of these, is the **Number of time units** to forecast text box. This controls how many time steps into the future the forecaster will produce predictions for. The default is set to 1, i.e. the system will make a single 1-step-ahead prediction. For the airline data we set this to 24 (to make monthly predictions into the future for a two-year period) and for the wine data we set it to 12 (to make monthly predictions into the future for a one-year period). The units correspond to the periodicity of the data (if known). For example, with data recorded on a daily basis the time units are days.

2.2.1.2.2 Time stamp

Next is the **Time stamp** drop-down box. This allows the user to select which, if any, field in the data holds the time stamp. If there is a date field in the data, then the system selects this automatically. If there is no date present in the data, then the "<Use an artificial time index>" option is selected automatically. The user may select the time stamp manually; and will need to do so if the time stamp is a non-date numeric field (because the system can't distinguish this from a potential target field). The user also has the option of selecting "<None>" from the drop-down box in order to tell the system that no time stamp (artificial or otherwise) is to be used.

2.2.1.2.3 Periodicity

Underneath the Time stamp drop-down box is a drop-down box that allows the user to specify the **Periodicity** of the data. If a date field has been selected as the time stamp, then the system can use heuristics to automatically detect the periodicity - "<Detect automatically>" will be set as the default if the system has found and set a date attribute as the time stamp initially. If the time stamp is not a date, then the user can explicitly tell the system what the periodicity is or select "<Unknown>" if it is not known. Periodicity is used to set reasonable defaults for the creation of lagged variables (covered below in the **Advanced Configuration** section). In the case where the time stamp is a date, Periodicity is also used to create a default set of fields derived from the date. E.g. for a monthly periodicity, **month of the year** and **quarter** fields are automatically created.

2.2.1.2.4 Skip list

Below the Periodicity drop-down box is a field that allows the user to specify time periods that should not count as a time stamp increment with respect to the modeling, forecasting and visualization process. For example, consider daily trading data for a given stock. The market is closed for trading over the weekend and on public holidays, so these time periods do not count as an increment and the difference, for example, between market close on Friday and on the following Monday is one-time unit (not three). The heuristic used to automatically detect periodicity can't cope with these "holes" in the data, so the user must specify a periodicity to use and supply the time periods that are not to consider as increments in the **Skip list** text field.

The Skip list field can accept strings such as "weekend", "sat", "tuesday", "mar" and "october", specific dates (with optional formatting string) such as "2011-07-04@yyyy-MM-dd", and integers (that get interpreted differently depending on the specified periodicity). For daily data an integer is interpreted as the day of the year; for hourly data it is the hour of the day and for monthly data it is the month of the year. For specific dates, the system has a default formatting string ("yyyy-MM-dd'T'HH:mm:ss") or the user can specify one to use by suffixing the date with "@<format>". If all dates in the list have the same format, then it only has to be specified once (for the first date present in the list) and then this will become the default format for subsequent dates in the list.

The following screenshots show an example for the "appleStocks2011" data (found in sample-data directory of the package). This file contains daily high, low, opening and closing data for Apple computer stocks from January 3rd to August 10th 2011. The data was take from Yahoo finance (http://finance.yahoo.com/q/hp?s=AAPL&a=00&b=3&c=2011&d=07&e=10&f=2011&g=d). A five-day forecast for the daily closing value has been set, a maximum lag of 10 configured (see "Lag creation" in Section 3.2), periodicity set to "Daily" and the following Skip list entries provided in order to cover weekends and public holidays:

weekend, 2011-01-17@yyyy-MM-dd, 2011-02-21, 2011-04-22, 2011-05-30, 2011-07-04

Note that it is important to enter dates for public holidays (and any other dates that do not count as increments) that will occur during the future time period that is being forecasted.

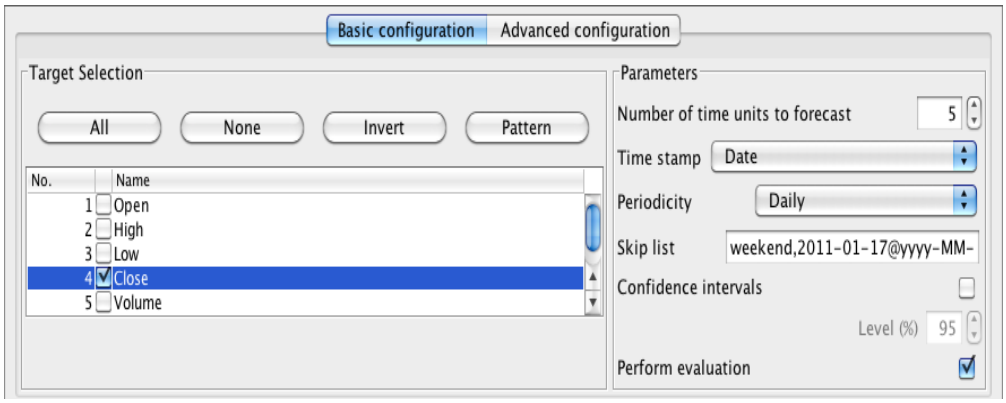


Figure 2.4: Skip List Time Series Environment in WEKA

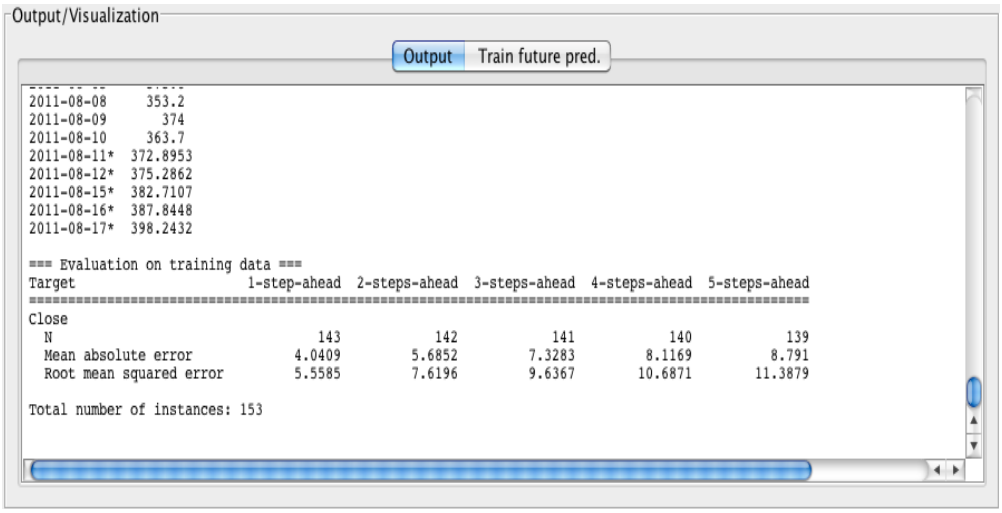


Figure 2.5: Evaluation on training data for 5-steps-ahead

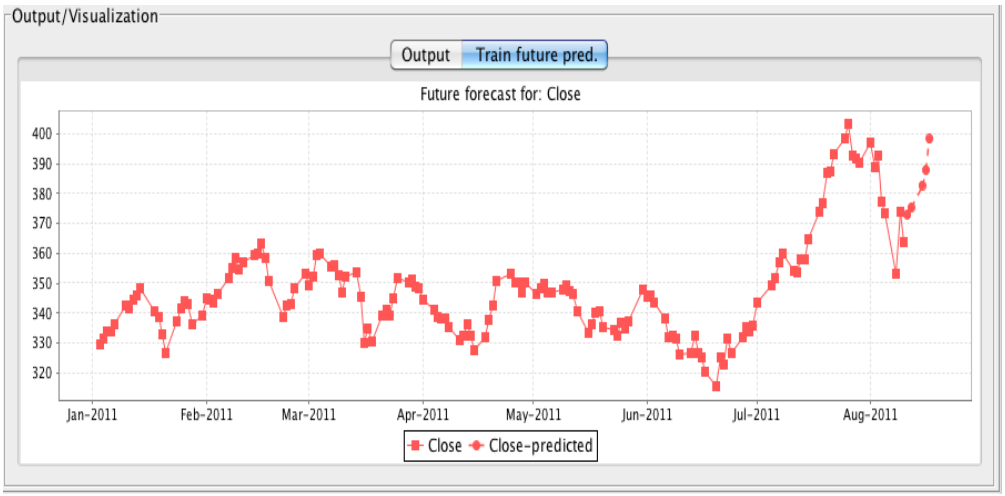


Figure 2.6: Output Graph

2.2.1.2.5 Confidence intervals

Below the Time stamp drop-down box is a check box and text field that the user can opt to have the system compute confidence bounds on the predictions that it makes. The default confidence level is 95%. The system uses predictions made for the known target values in the training data to set the confidence bounds. So, a 95% confidence level means that 95% of the true target values fell within the interval. Note that the confidence intervals are computed for each step-ahead level independently, i.e. all the one-step-ahead predictions on the training data are used to compute the one-step-ahead confidence interval, all the two-step-ahead predictions are used to compute the two-step-ahead interval, and so on.

2.2.1.2.6 Perform evaluation

By default, the system is set up to learn the forecasting model and generate a forecast beyond the end of the training data. Selecting the **Perform evaluation** check box tells the system to perform an evaluation of the forecaster using the training data. That is, once the forecaster has been trained on the data, it is then applied to make a forecast at each time point (in order) by stepping through the data. These predictions are collected and summarized, using various metrics, for each future time step forecasted, i.e. all the one-step-ahead predictions are collected and summarized, all the two-step-ahead predictions are collected and summarized, and so on. This allows the user to see, to a certain degree, how forecasts further out in time compare to those closer in time. The Advanced Configuration panel allows the user to fine tune configuration by selecting which metrics to compute and whether to hold-out some data from the end of the training data as a separate test set. The following screenshot shows the default evaluation on the Australian wine training data for the "Fortified" and "Dry-white" targets.

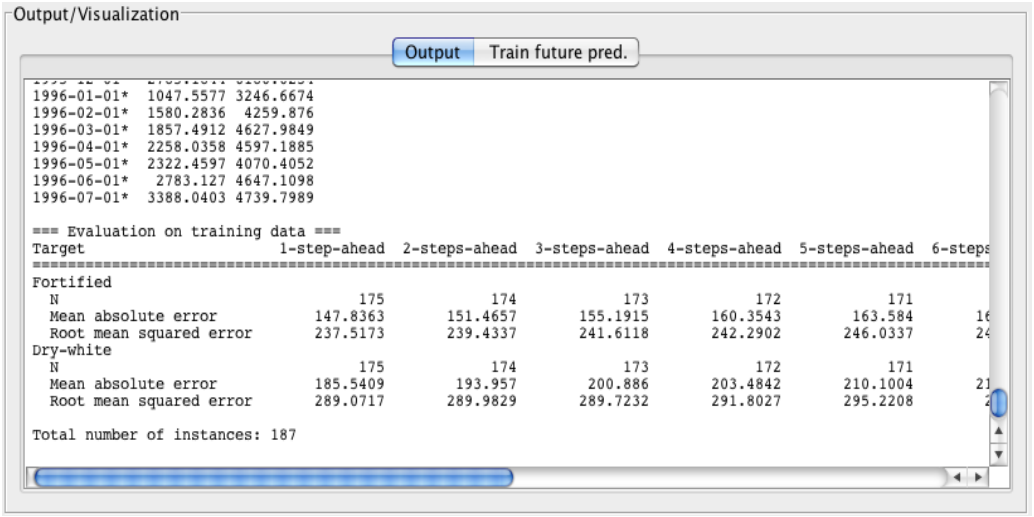


Figure 2.7: Evaluation on training data for 6-steps-ahead

2.2.1.3 Output

Output generated by settings available from the basic configuration panel includes the training evaluation (shown in the previous screenshot), graphs of forecasted values beyond the end of the training data (as shown in Section 3.1), forecasted values in text form and a textual description of the model learned. There are more options for output available in the advanced configuration panel (discussed in the next section). The next screenshot shows the model learned on the airline data. By default, the time series environment is configured to learn a linear model, that is, a linear support vector machine to be precise. Full control over the underlying model learned and its parameters is available in the advanced configuration panel.

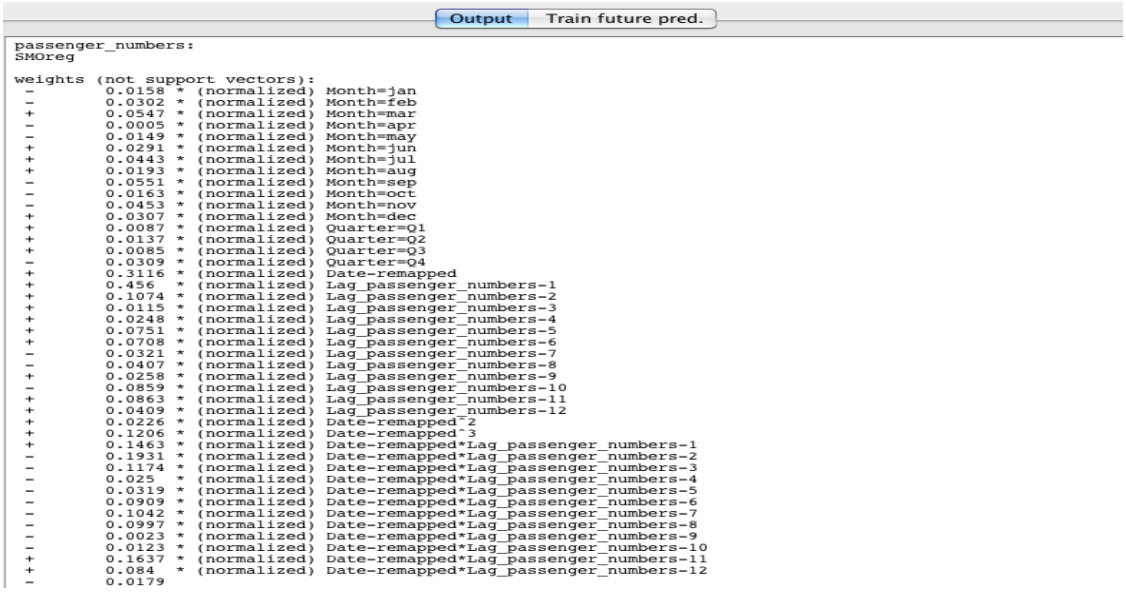


Figure 2.8: Output

Results of time series analysis are saved into a **Result list** on the lower left-hand side of the display. An entry in this list is created each time a forecasting analysis is launched by pressing the **Start** button. All textual output and graphs associated with an analysis run are stored with their respective entry in the list. Also stored in the list is the forecasting model itself. The model can be exported to disk by selecting **Save forecasting model** from a contextual popup menu that appears when right-clicking on an entry in the list.

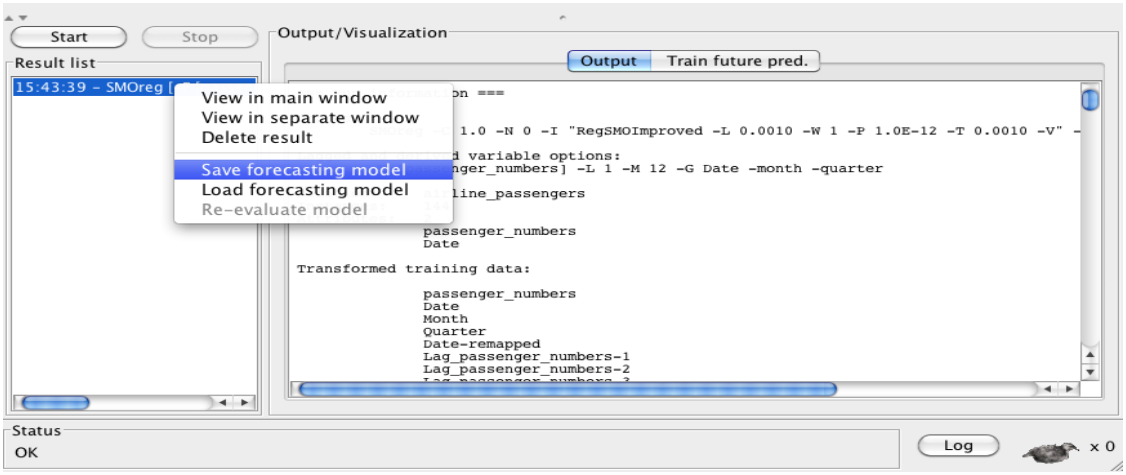


Figure 2.9: Save Forecasting Model

It is important to realize that, when saving a model, the model that gets saved is the one that is built on the training data corresponding to that entry in the history list. If performing an evaluation where some of the data is held out as a separate test set (see below in Section 3.2) then the model saved has only been trained on part of the available data. It is a good idea to turn off hold-out evaluation and construct a model on all the available data before saving the model.

2.2.2 Advanced Configuration

The advanced configuration panel gives the user full control over a number of aspects of the forecasting analysis. These include the choice of underlying model and parameters, creation of lagged variables, creation of variables derived from a date time stamp, specification of "overlay" data, evaluation options and control over what output is created. Each of these has a dedicated sub-panel in the advanced configuration and is discussed in the following sections.

2.2.2.1 Base learner

The **Base learner** panel provides control over which Weka learning algorithm is used to model the time series. It also allows the user to configure parameters specific to the learning algorithm selected.

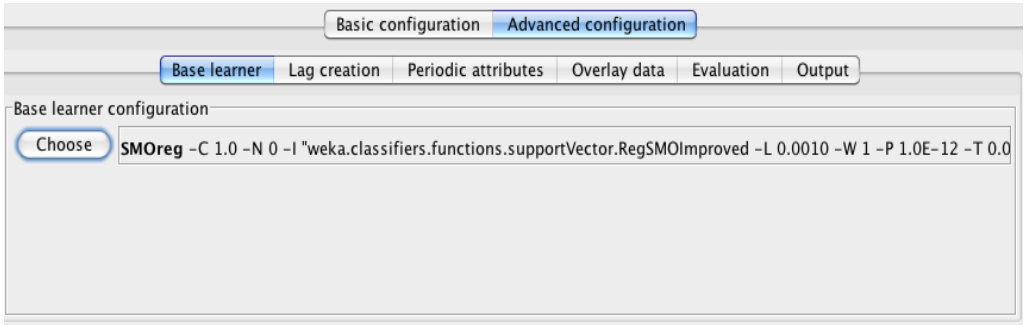


Figure 2.10: Advanced Configuration of Time Series Environment in WEKA

By default, the analysis environment is configured to use a linear support vector machine for regression (Weka's SMOreg). This can easily be changed by pressing the **Choose** button and selecting another algorithm capable of predicting a numeric quantity.

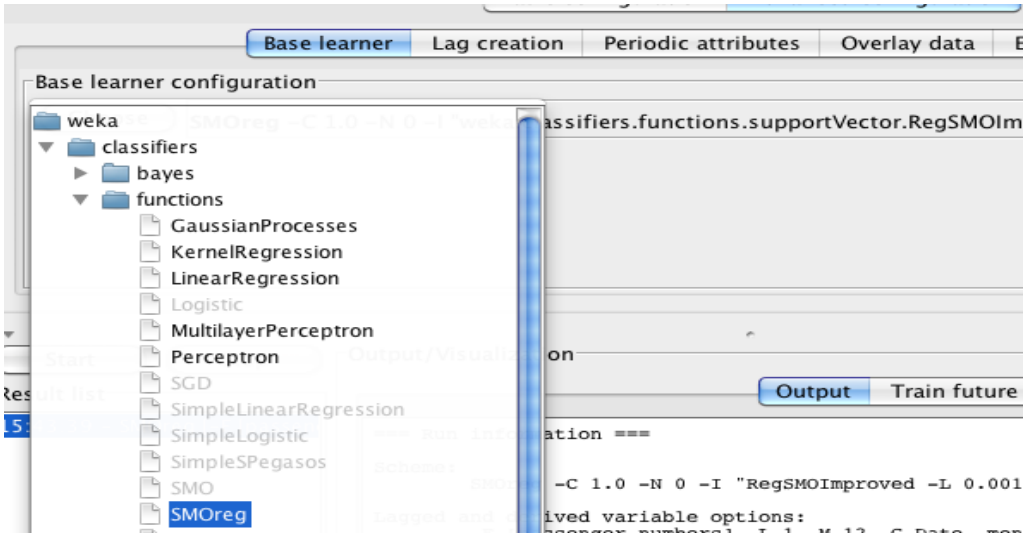


Figure 2.11: Selection of an Algorithm

Adjusting the individual parameters of the selected learning algorithm can be accomplished by clicking on the **options panel**, found immediately to the right of the **Choose** button. Doing so brings up an options dialog for the learning algorithm.

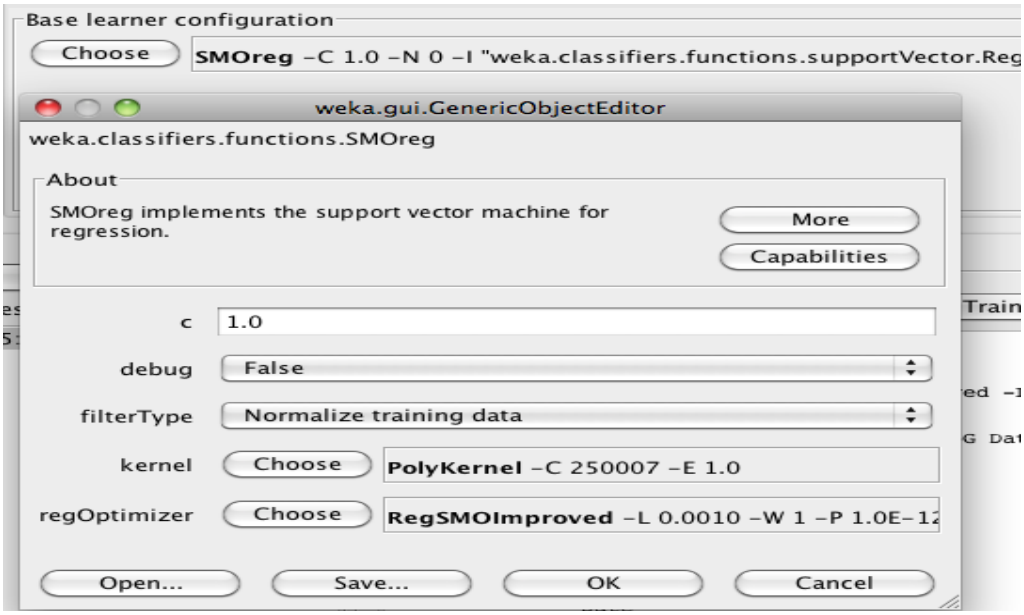


Figure 2.12: Classifier functions in WEKA

2.2.2.2 Lag creation

The **Lag creation** panel allows the user to control and manipulate how lagged variables are created. Lagged variables are the main mechanism by which the relationship between past and current values of a series can be captured by propositional learning algorithms. They create a "window" or "snapshot" over a time period. Essentially, the number of lagged variables created determines the size of the window. The basic configuration panel uses the **Periodicity** setting to set reasonable default values for the number of lagged variables (and hence the window size) created. For example, if you had monthly sales data then including lags up to 12 time steps into the past would make sense; for hourly data, you might want lags up to 24 time steps or perhaps 12.

The left-hand side of the lag creation panel has an area called lag length that contains controls for setting and fine-tuning lag lengths. At the top of this area there is an **Adjust for variance** check box which allows the user to opt to have the system compensate for variance in the data. It does this by taking the log of each target before creating lagged variables and building the model. This can be useful if the variance (how much the data jumps around) increases or decreases over the course of time. Adjusting for variance may, or may not, improve performance. It is best to experiment and see if it helps for the data/parameter selection combination at hand. Below the adjust for variance check box is a **Use custom lag lengths** check box. This allows the user to alter the default lag lengths that are set by the basic configuration panel. Note that the numbers shown for the lengths are not necessarily the defaults that will be used. If the user has selected "<Detect automatically>" in the periodicity drop-down box on the basic configuration panel, then the actual default lag lengths get set when the data gets analyzed at run time. The **Minimum lag** text field allows the user to specify the minimum previous time step to create a lagged field for - e.g. a value of 1 means that a lagged variable will be created that holds target values at time - 1. The **Maximum lag** text field specifies the maximum previous time step to create a lagged variable for - e.g. a value of 12 means that a lagged variable will be created that holds target values at time - 12. All time periods between the minimum and maximum lag will be turned into lagged variables. It is possible to fine tune the creation of variables within the minimum and maximum by entering a range in the **Fine tune lag selection** text field. In the screenshot below we have weekly data so have opted to set minimum and maximum lags to 1 and 52 respectively. Within this we have opted to only create lags 1-26 and 52.

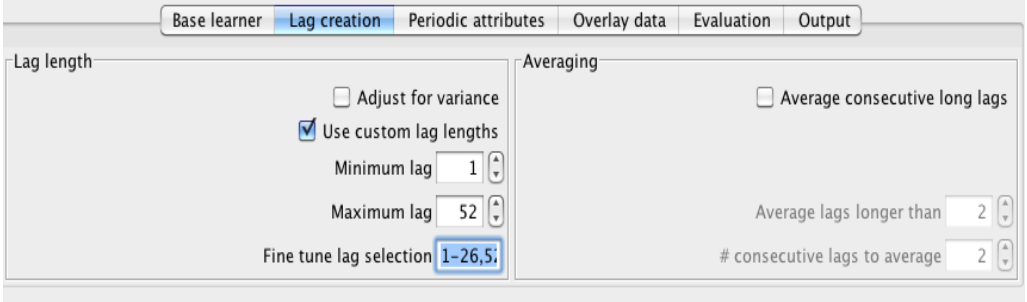


Figure 2.13: Lag Creation in WEKA

On the right-hand side of the lag creation panel is an area called Averaging. Selecting the **Average consecutive long lags** check box enables the number of lagged variables to be reduced by averaging the values of several consecutive (in time) variables. This can be useful when you want to have a wide window over the data but perhaps don't have a lot of historical data points. A rule of thumb states that you should have at least 10 times as many rows as fields (there are exceptions to this depending on the learning algorithm - e.g. support vector machines can work very well in cases where there are many more fields than rows). Averaging a number of consecutive lagged variables into a single field reduces the number of input fields with probably minimal loss of information (for long lags at least). The **Average lags longer than** text field allows the user to specify when the averaging process will begin. For example, in the screenshot above this is set to 2, meaning that the time - 1 and time - 2 lagged variables will be left untouched while time - 3 and higher will be replaced with averages. The **# consecutive lags to average** controls how many lagged variables will be part of each averaged group. For example, in the screenshot above this is also set to 2, meaning that time - 3 and time - 4 will be averaged to form a new field; time - 5 and time - 6 will be averaged to form a new field; and so on. Note that only consecutive lagged variable will be averaged, so in the example above, where we have already fine-tuned the lag creation by selecting lags 1-26 and 52, time - 26 would never be averaged with time - 52 because they are not consecutive.

2.2.2.3 Periodic attributes

The **Periodic attributes** panel allows the user to customize which date-derived periodic attributes are created. This functionality is only available if the data contains a date time stamp. If the time stamp is a date, then certain defaults (as determined by the Periodicity setting from the basic configuration panel) are automatically set. For example, if the data has a monthly time interval then month of the year and quarter are automatically included as variables in the data. The user can select the **customize** checkbox in the date-derived periodic creation area to disable, select and create new custom date-derived variables. When the checkbox is selected the user is presented with a set of pre-defined variables as shown in the following screenshot:

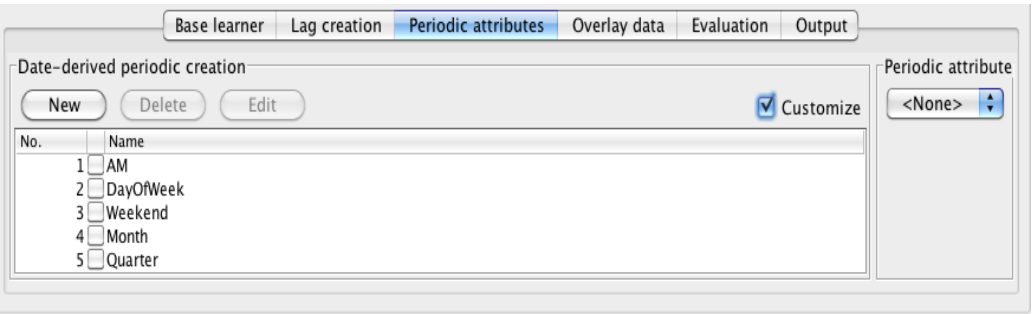


Figure 2.14: Periodic Attributes in WEKA

Leaving all of the default variables unselected will result in no date-derived variables being created. Aside from the predefined defaults, it is possible to create custom date-derived variables. A new custom date-derived variable, based on a rule, can be created by pressing the **New** button. This brings up an editor as shown below:

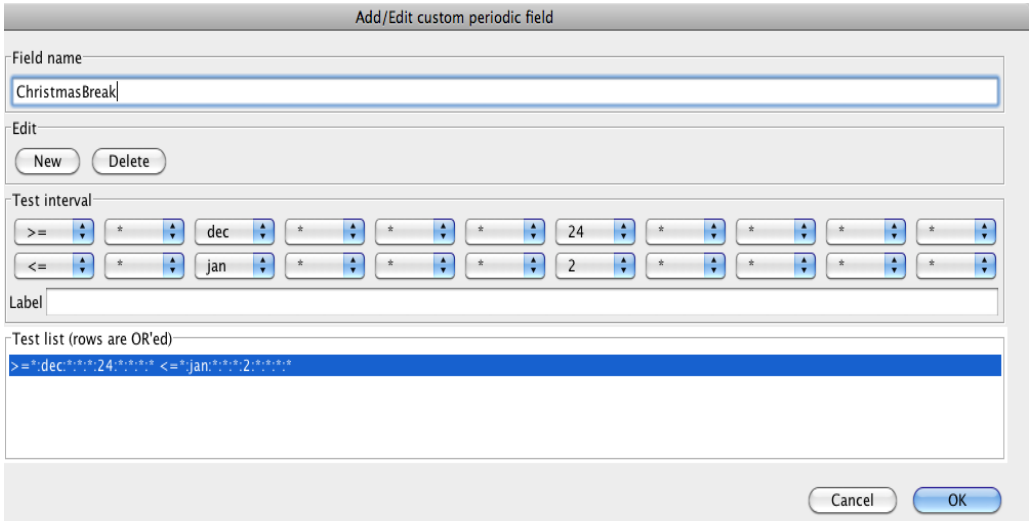


Figure 2.15: Edit custom period field in WEKA

In this example, we have created a custom date-derived variable called "ChistmasBreak" that comprises a single date-based test (shown in the list area at the bottom of the dialog). This variable is boolean and will take on the value 1 when the date lies between December 24th and January 2nd inclusive. Additional tests can be added to allow the rule to evaluate to true for disjoint periods in time.

The **Field name** text field allows the user to give the new variable a name. Below this are two buttons. The **New** button adds a new test to the rule and the **Delete** button deletes the currently selected test from the list at the bottom. Selecting a test in the list copies its values to the drop-down boxes for the upper and lower bounds of the test, as shown in the Test interval area of the screenshot above. Each drop-down box edits one element of a bound. They are (from left to right): **comparison operator, year, month of the year, week of the year, week of the month, day of the year, day of the month, day of the week, hour of the day, minute of the hour and second.** Tool tips giving the function of each appear when the mouse hovers over each drop-down box. Each drop-down box contains the legal values for that element of the bound. Asterix characters ("*") are "wildcards" and match anything.

Below the Test interval area is a **Label** text field. This allows a string label to be associated with each test interval in a rule. All the intervals in a rule must have a label, or none of them. Having some intervals with a label and some without will generate an error. If all intervals have a label, then these will be used to set the value of the custom field associated with the rule instead of just 0 or 1. Evaluation of the rule proceeds as a list, i.e. from top to bottom, and the first interval that evaluates to true is the one that is used to set the value of the field. A default label (i.e. one that gets assigned if no other test interval matches) can be set up by using all wildcards for the last test interval in the list. In the case where all intervals have labels, and if there is no "catch-all" default set up, then the value for the custom field will be set to missing if no interval matches. This is different to the case where labels are not used and the field is a binary flag - in this case, the failure to match an interval results in the value of the custom field being set to 0.

2.2.2.4 Overlay data

The **Overlay data** panel allows the user to specify fields (if any) that should be considered as "overlay" data. The default is not to use overlay data. By "overlay" data we mean input fields that are to be considered external to the data transformation and closed-loop forecasting processes. That is, data that is not to be forecasted, can't be derived automatically and will be supplied for the future time periods to be forecasted. In the screenshot below, the Australian wine data has been loaded into the system and Fortified has been selected as the target to forecast. By selecting the **Use overlay data** checkbox, the system shows the remaining fields in the data that have not been selected as either targets or the time stamp. These fields are available for use as overlay data.

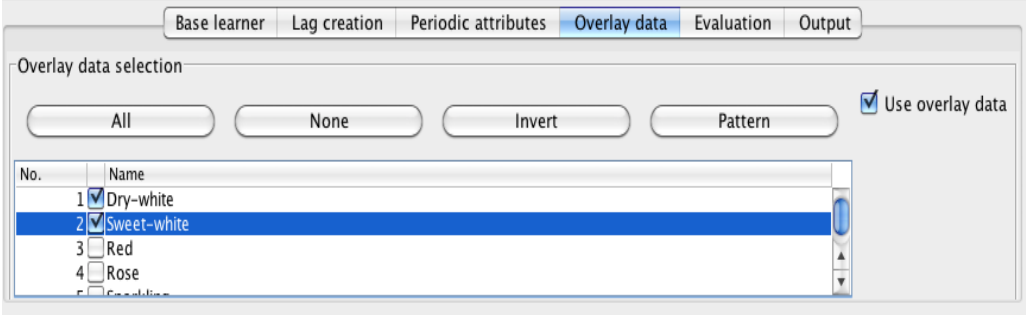


Figure 2.16: Overlay data in advanced configuration in WEKA

The system will use selected overlay fields as inputs to the model. In this way it is possible for the model to take into account special historical conditions (e.g. stock market crash) and factor in conditions that will occur at known points in the future (e.g. irregular sales promotions that have occurred historically and are planned for the future). Such variables are often referred to as intervention variables in the time series literature.

When executing an analysis that uses overlay data the system may report that it is unable to generate a forecast beyond the end of the data. This is because we don't have values for the overlay fields for the time periods requested, so the model is unable to generate a forecast for the selected target(s). Note that it is possible to evaluate the model on the training data and/or data held-out from the end of the training data because this data does contain values for overlay fields. More information on making forecasts that involve overlay data is given in the documentation on the forecasting plugin step for Pentaho Data Integration.

2.2.2.5 Evaluation

The **Evaluation** panel allows the user to select which evaluation metrics they wish to see, and configure whether to evaluate using the training data and/or a set of data held out from the end of the training data. Selecting **Perform evaluation** in the **Basic configuration** panel is equivalent to selecting **Evaluate on training** here. By default, the mean absolute error (MAE) and root mean square error (RMSE) of the predictions are computed. The user can select which metrics to compute in the Metrics area in on the left-hand side of the panel. The available metrics are:

1. Mean absolute error (MAE): sum (abs (predicted - actual)) / N

2. Mean squared error (MSE): sum ((predicted - actual) ^2) / N

3. Root mean squared error (RMSE): sqrt(sum((predicted - actual)^2) / N)

4. Mean absolute percentage error (MAPE): sum(abs((predicted - actual) / actual)) / N

5. Direction accuracy (DAC): count(sign(actual_current - actual_previous) == sign(pred_current - pred_previous)) / N

6. Relative absolute error (RAE): sum (abs (predicted - actual)) / sum (abs (previous_target - actual))

7. Root relative squared error (RRSE): sqrt(sum((predicted - actual)^2) / N) / sqrt(sum(previous_target - actual)^2) / N)

The relative measures give an indication of how the well forecaster's predictions are doing compared to just using the last known target value as the prediction. They are expressed as a percentage, and lower values indicate that the forecasted values are better predictions than just using the last known target value. A score of >=100 indicates that the forecaster is doing no better (or even worse) than predicting the last known target value. Note that the last known target value is relative to the step at which the forecast is being made - e.g. a 12-step-ahead prediction is compared relative to using the target value 12 time steps prior as the prediction (since this is the last "known" actual target value).

The text field to the right of the **Evaluate on held out training** check box allows the user to select how much of the training data to hold out from the end of the series in order to form an independent test set. The number entered here can either indicate an absolute number of rows, or can be a fraction of the training data (expressed as a number between 0 and 1).

2.2.2.6 Output

The **Output** panel provides options that control what textual and graphical output are produced by the system. The panel is split into two sections: Output options and Graphing options. The former controls what textual output appears in the main Output area of the environment, while the latter controls which graphs are generated.

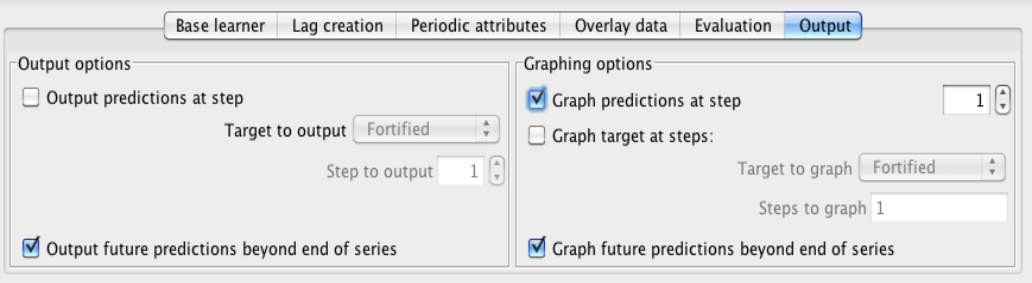


Figure 2.17: Selection of the attributes for output in advanced configuration in WEKA

In the Output area of the panel, selecting **Output predictions at step** causes the system to output the actual and predicted values for a single target at a single step. The error is also output. For example, the 5-step ahead predictions on a hold-out test set for the "Fortified" target in the Australian wine data is shown in the following screenshot.

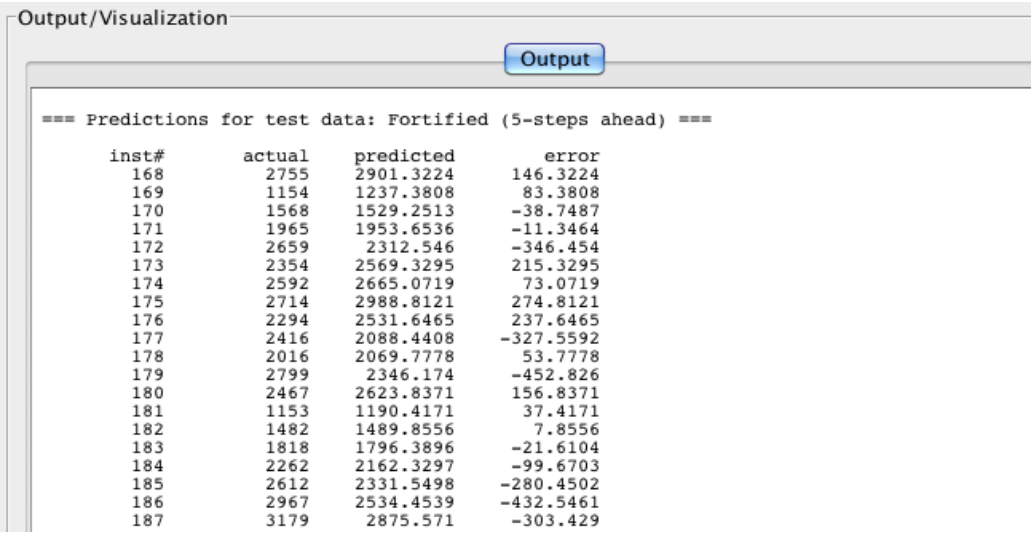


Figure 2.18: Output in advanced configuration in WEKA

Selecting **Output future predictions beyond the end of series** will cause the system to output the training data and predicted values (up to the maximum number of time units) beyond the end of the data for all targets predicted by the forecaster. Forecasted values are marked with a "*" to make the boundary between training values and forecasted values clear.

In the Graphing options area of the panel the user can select which graphs are generated by the system. Similar to the textual output, the predictions at a specific step can be graphed by selecting the **Graph predictions at step** check box. Unlike the textual output, all targets predicted by the forecaster will be graphed. Selecting the **Graph target at steps** checkbox allows a single target to be graphed at more than one step - e.g. a graph can be generated that shows 1-step-ahead, 2-step-ahead and 5-step ahead predictions for the same target. The **Target to graph** drop-down box and the **Steps to graph** text field become active when the **Graph**

target at steps checkbox is selected. The following screenshot shows graphing the "Fortified" target from the Australian wine data on a hold-out set at steps 1,2,3,6 and 12.

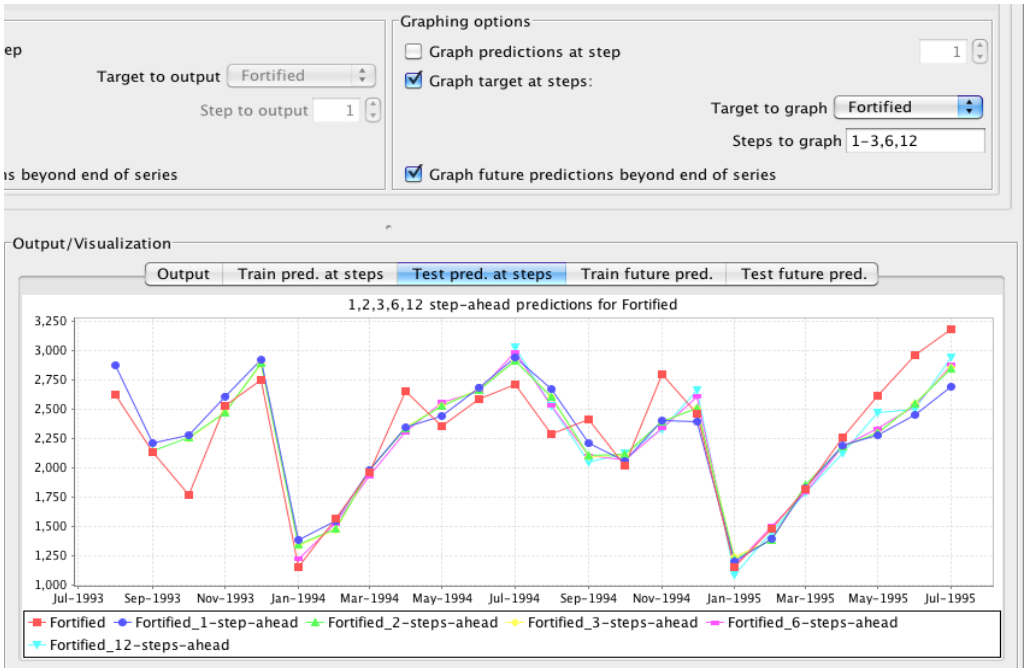


Figure 2.19: Output Graph in advanced configuration in WEKA

2.3 Time Series Application:

a. Economic Forecasting
b. Sales Forecasting
c. Budgetary Analysis
d. Stock Market Analysis
e. Yield Projections
f. Process and Quality Control
g. Inventory Studies
h. Workload Projections
i. Utility Studies
j. Census Analysis

TRAINING ASSESSMENT

3. Training Assessment

3.1 Dataset:

For this research, we collected the dataset from Walmart. The data set included sales data for 111 products whose sales may be affected by the weather (such as milk, bread, umbrellas, etc.). These 111 products are sold in stores at 45 different Walmart locations. Some of the products may be a similar item (such as milk) but have a different id in different stores/regions/suppliers. The 45 locations are covered by 20 weather stations (i.e. some of the stores are nearby and share a weather station).

There were three files. One file contained various weather info collected from 20 weather stations of 45 locations of a year. It contained weather info like Max Temperature, Min Temperature, Average Temperature, Snowfall, Dew Point, Precip Total, Sea Level, Wind Speed etc. There was another file which contained info about which store is located near which weather station, this file is vital for finding how the weather condition was near a particular store by linking store number with weather station number. Finally, there was a file with sales data for a year. It contained info about which store sold which item and how many units in a particular date. The names of store locations and item names were changed with code names like “Store One”, “Item One” etc.

3.2 Pre-processing:

We processed the data using python. We selected “Item Five” and “Store Two”. So, our goal was to find weather’s effect on item five sold at store two. We made a single file from the three dataset. We choose average temperature and weather condition for checking whether they have any impact on Item Five sold on Store Two. The common factor between weather and sales data was the date. From there we constructed a single file for making a training set.

3.3 Storage:

We used MySQL as our data storage. Using python, we generated a csv file. Then we exported the csv file in a MySQL database. The database provided us with the opportunity to run complex query to fetch appropriate data.

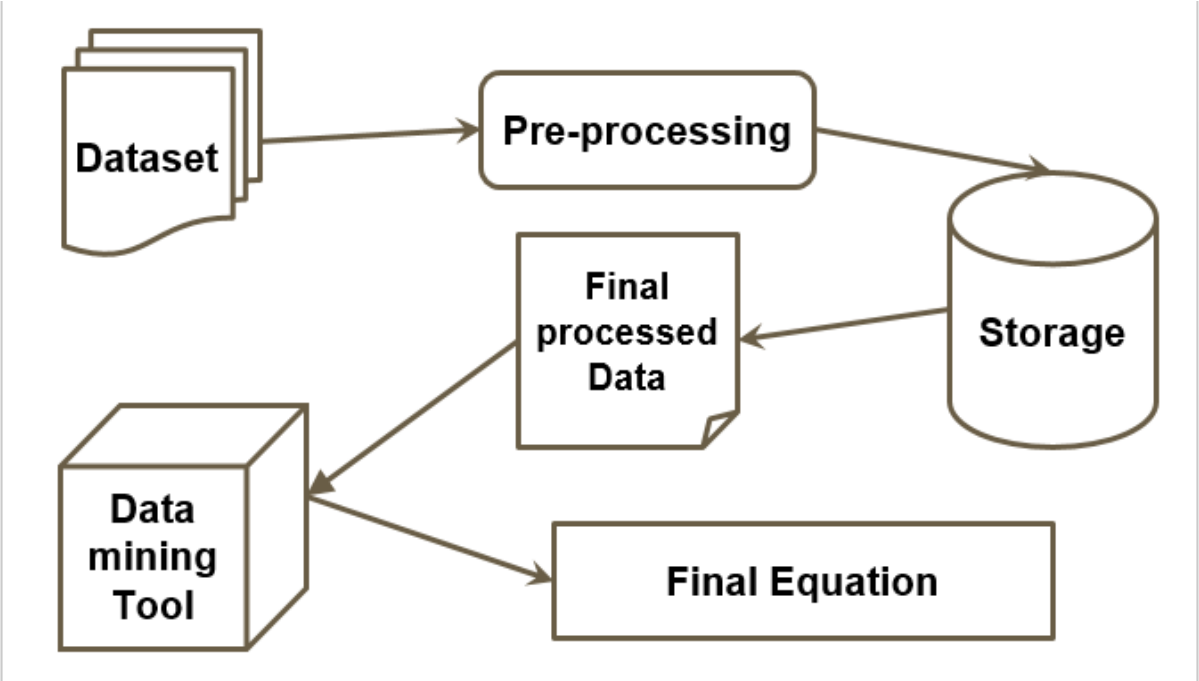


Figure 3: Our Approach

Table 3.1: Store-2 Item-5 Sells Information

date	item_nbr	units	store_nbr	station_nbr	tmp	con
01-01-12	5	191	2	14	42	0
01-02-12	5	147	2	14	36	0
01-03-12	5	104	2	14	42	0
01-04-12	5	58	2	14	45	0
01-05-12	5	138	2	14	48	0
01-06-12	5	145	2	14	52	0
01-07-12	5	71	2	14	49	0
01-08-12	5	157	2	14	47	0
01-09-12	5	53	2	14	42	0
01-10-12	5	267	2	14	42	0.99
01-11-12	5	91	2	14	43	0.99
01-12-12	5	76	2	14	28	0
1/13/2012	5	89	2	14	35	0
1/14/2012	5	75	2	14	44	0
1/15/2012	5	168	2	14	45	0
1/16/2012	5	37	2	14	61	0
1/17/2012	5	55	2	14	44	0
1/18/2012	5	90	2	14	32	0
1/19/2012	5	38	2	14	42	0
1/20/2012	5	38	2	14	38	0
1/21/2012	5	90	2	14	32	0
1/22/2012	5	89	2	14	52	1.98
1/23/2012	5	41	2	14	46	0
1/24/2012	5	63	2	14	50	1.32
1/25/2012	5	51	2	14	45	1.32
1/26/2012	5	27	2	14	44	0.33
1/27/2012	5	41	2	14	49	0
1/28/2012	5	37	2	14	39	0
1/29/2012	5	99	2	14	43	0
1/30/2012	5	39	2	14	50	0
1/31/2012	5	52	2	14	55	0
02-01-12	5	182	2	14	51	0
02-02-12	5	121	2	14	52	0.99
02-03-12	5	74	2	14	55	2.97
02-04-12	5	197	2	14	41	0
02-05-12	5	191	2	14	43	0
02-06-12	5	107	2	14	41	0
02-07-12	5	133	2	14	45	0
02-08-12	5	50	2	14	33	0
02-09-12	5	45	2	14	35	2.31
02-10-12	5	66	2	14	34	2.97
02-11-12	5	174	2	14	24	0
02-12-12	5	194	2	14	27	1.32
2/13/2012	5	109	2	14	33	2.97
2/14/2012	5	45	2	14	38	4.29
2/15/2012	5	31	2	14	55	1.98

2/16/2012	5	15	2	14	48	0.33
2/17/2012	5	55	2	14	48	1.65
2/18/2012	5	75	2	14	43	2.31
2/19/2012	5	111	2	14	44	0
2/20/2012	5	64	2	14	50	3.3
2/21/2012	5	80	2	14	47	0
2/22/2012	5	26	2	14	54	0
2/23/2012	5	69	2	14	55	0
2/24/2012	5	17	2	14	43	0
2/25/2012	5	36	2	14	45	0
2/26/2012	5	93	2	14	52	0
2/27/2012	5	39	2	14	56	0
2/28/2012	5	70	2	14	61	0.66
2/29/2012	5	55	2	14	52	0
03-01-12	5	91	2	14	58	0
03-02-12	5	81	2	14	52	0
03-03-12	5	60	2	14	45	0
03-04-12	5	114	2	14	50	0
03-05-12	5	91	2	14	59	0
03-06-12	5	83	2	14	62	0
03-07-12	5	148	2	14	67	0
03-08-12	5	77	2	14	51	3.3
03-09-12	5	35	2	14	48	0
03-10-12	5	110	2	14	50	1.32
03-11-12	5	87	2	14	52	1.32
03-12-12	5	63	2	14	63	0
3/13/2012	5	31	2	14	66	0.33
3/14/2012	5	59	2	14	70	0
3/15/2012	5	79	2	14	73	0
3/16/2012	5	79	2	14	69	0.33
3/17/2012	5	157	2	14	71	0
3/18/2012	5	94	2	14	68	1.65
3/19/2012	5	89	2	14	61	3.96
3/20/2012	5	74	2	14	55	1.32
3/21/2012	5	86	2	14	55	4.29
3/22/2012	5	40	2	14	51	0.99
3/23/2012	5	17	2	14	56	0
3/24/2012	5	27	2	14	63	0
3/25/2012	5	45	2	14	64	0
3/26/2012	5	44	2	14	68	0
3/27/2012	5	54	2	14	69	0
3/28/2012	5	51	2	14	68	0
3/29/2012	5	26	2	14	70	0.33
3/30/2012	5	21	2	14	70	0.33
3/31/2012	5	40	2	14	71	1.65
04-01-12	5	206	2	14	76	0.33
04-02-12	5	123	2	14	71	1.98
04-03-12	5	85	2	14	60	3.3
04-04-12	5	50	2	14	57	0.33

04-05-12	5	86	2	14	57	0
04-06-12	5	106	2	14	60	0.33
04-07-12	5	87	2	14	65	2.31
04-08-12	5	60	2	14	60	0
04-09-12	5	24	2	14	65	0.33
04-10-12	5	315	2	14	67	0.66
04-11-12	5	41	2	14	62	0
04-12-12	5	30	2	14	64	0
4/13/2012	5	76	2	14	68	4.29
4/14/2012	5	104	2	14	70	0.99
4/15/2012	5	37	2	14	62	2.31
4/16/2012	5	41	2	14	59	0
4/17/2012	5	66	2	14	61	0
4/18/2012	5	30	2	14	65	0
4/19/2012	5	88	2	14	68	4.29
4/20/2012	5	39	2	14	57	0
4/21/2012	5	99	2	14	59	0
4/22/2012	5	112	2	14	62	0
4/23/2012	5	29	2	14	59	0
4/24/2012	5	50	2	14	66	0
4/25/2012	5	26	2	14	77	0
4/26/2012	5	26	2	14	75	0
4/27/2012	5	60	2	14	79	0
4/28/2012	5	53	2	14	72	3.3
4/29/2012	5	69	2	14	73	1.65
4/30/2012	5	93	2	14	72	3.3
05-01-12	5	191	2	14	73	1.98
05-02-12	5	123	2	14	76	0
05-03-12	5	27	2	14	80	0
05-04-12	5	47	2	14	76	0
05-05-12	5	84	2	14	76	0
05-06-12	5	43	2	14	78	0.33
05-07-12	5	88	2	14	66	0
05-08-12	5	78	2	14	64	0
05-09-12	5	42	2	14	65	0
05-10-12	5	68	2	14	66	0
05-11-12	5	60	2	14	62	1.32
05-12-12	5	106	2	14	65	1.98
5/13/2012	5	74	2	14	66	0
5/14/2012	5	99	2	14	69	0
5/15/2012	5	79	2	14	71	0
5/16/2012	5	56	2	14	71	0
5/17/2012	5	34	2	14	72	0
5/18/2012	5	93	2	14	75	0
5/19/2012	5	79	2	14	78	1.98
5/20/2012	5	166	2	14	66	2.97
5/21/2012	5	45	2	14	70	4.29
5/22/2012	5	38	2	14	69	0
5/23/2012	5	47	2	14	78	0

5/24/2012	5	88	2	14	81	0
5/25/2012	5	48	2	14	80	0
5/26/2012	5	61	2	14	81	0
5/27/2012	5	69	2	14	78	0
5/28/2012	5	141	2	14	79	1.98
5/29/2012	5	40	2	14	76	3.96
5/30/2012	5	34	2	14	74	1.98
5/31/2012	5	20	2	14	70	0
06-01-12	5	107	2	14	60	2.97
06-02-12	5	132	2	14	70	3.3
06-03-12	5	190	2	14	77	2.97
06-04-12	5	48	2	14	81	0
06-05-12	5	90	2	14	79	0
06-06-12	5	35	2	14	72	4.29
06-07-12	5	87	2	14	71	1.32
06-08-12	5	59	2	14	73	0.33
06-09-12	5	76	2	14	75	0
06-10-12	5	63	2	14	81	0
06-11-12	5	59	2	14	81	0
06-12-12	5	48	2	14	76	0
6/13/2012	5	78	2	14	75	0
6/14/2012	5	45	2	14	80	0
6/15/2012	5	40	2	14	76	2.31
6/16/2012	5	50	2	14	81	0
6/17/2012	5	70	2	14	81	0
6/18/2012	5	79	2	14	81	0
6/19/2012	5	63	2	14	81	0
6/20/2012	5	94	2	14	82	0
6/21/2012	5	42	2	14	76	2.97
6/22/2012	5	39	2	14	81	0.33
6/23/2012	5	66	2	14	84	0
6/24/2012	5	117	2	14	83	0
6/25/2012	5	24	2	14	86	0
6/26/2012	5	71	2	14	88	0
6/27/2012	5	37	2	14	88	0.33
6/28/2012	5	80	2	14	85	0
6/29/2012	5	41	2	14	84	0.33
6/30/2012	5	74	2	14	84	0
07-01-12	5	224	2	14	82	0
07-02-12	5	79	2	14	84	0
07-03-12	5	26	2	14	87	0
07-04-12	5	52	2	14	86	0
07-05-12	5	128	2	14	85	0
07-06-12	5	52	2	14	85	0
07-07-12	5	80	2	14	86	0
07-08-12	5	82	2	14	84	0
07-09-12	5	41	2	14	87	0.99
07-10-12	5	100	2	14	83	1.98

After getting the final table including date, item number, store number, weather station number, unit, temperature & condition. We take weather data as a numeric number as like rain=0, snowfall=1, dew=0.5 etc. We converted the file into. arff format. Then we performed several operations in WEKA in advanced settings which included a package, time series forecasting.

RESULT & COMPARISON

4. Result and Comparison:

4.1 Regression Result for 1-Step Monthly

```
Relation:     item_units
Instances:    214
Attributes:   7
              Date
              item_nbr
              units
              store_nbr
              station_nbr
              tmp
              con

Transformed training data:

              units
              Date-remapped
              Lag_units-7

units:

Linear Regression Model

units =

      0.1081 * Lag_units-7 +
     67.762

=== Predictions for training data: units (1-step ahead) ===
```

Figure 4.1: Regression Result for 1-Step Monthly

4.2 Root Mean Squared Error for 1-Step Monthly

```
=== Evaluation on training data ===
Target                     1-step-ahead
==========================================
units
  N                                183
  Mean absolute error          33.9322
  Root mean squared error      46.9509

Total number of instances: 190

=== Evaluation on test data ===
Target                     1-step-ahead
==========================================
units
  N                                 24
  Mean absolute error          24.4687
  Root mean squared error       27.599

Total number of instances: 24
```

Figure 4.2: Root Mean Squared Error for 1-Step Monthly

4.3 Regression Graph for 1-Step Monthly Train Prediction

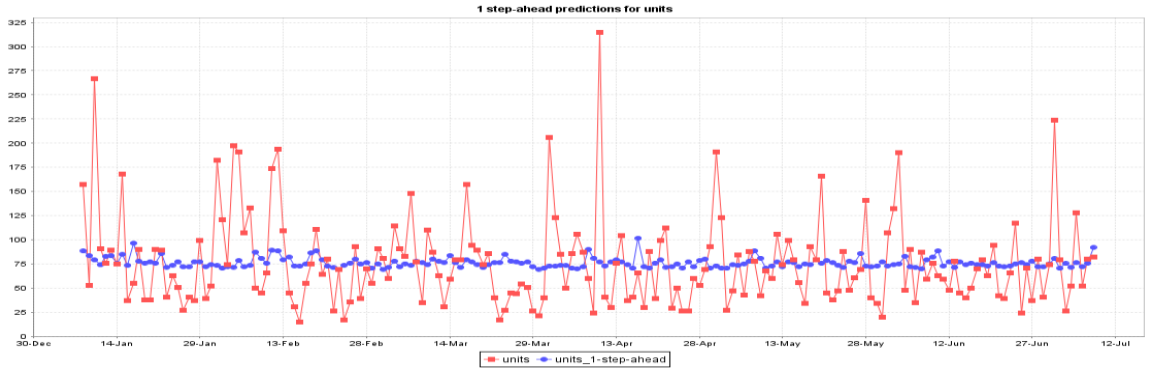


Figure 4.3: Regression Graph for 1-Step Monthly Train Prediction

4.4 Regression Graph for 1-Step Monthly Test Prediction

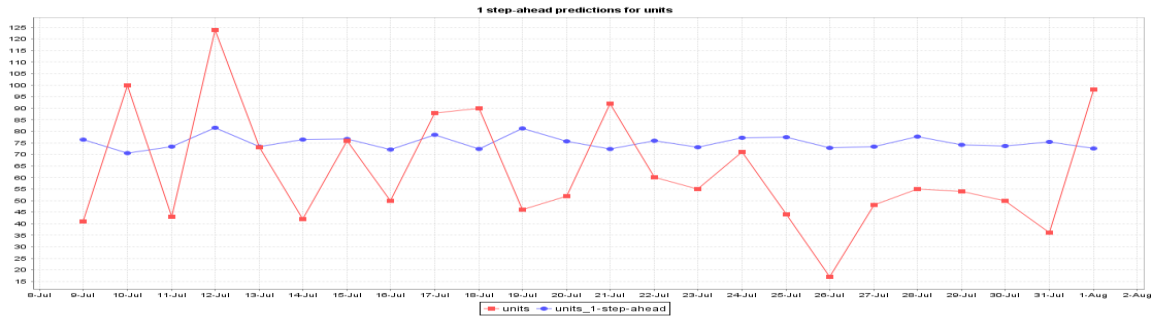


Figure 4.4: Regression Graph for 1-Step Monthly Test Prediction

4.5 Regression Graph for 1-Step Monthly Train for Future Prediction

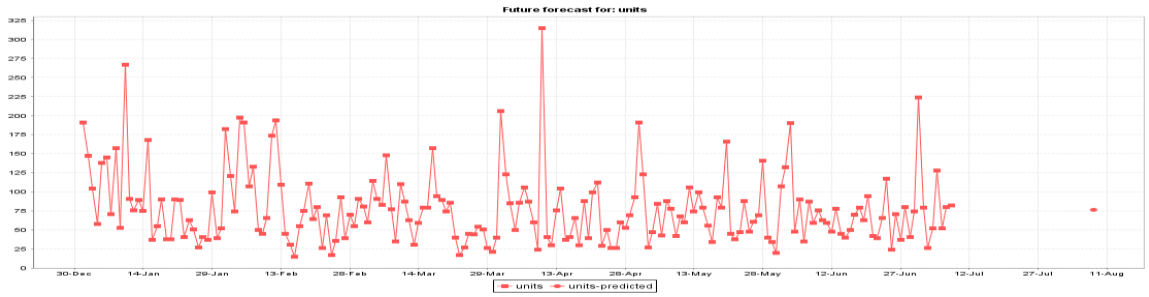


Figure 4.5: Regression Graph for 1-Step Monthly Train for Future Prediction

4.6 Regression Graph for 1-Step Monthly Test for Future Prediction

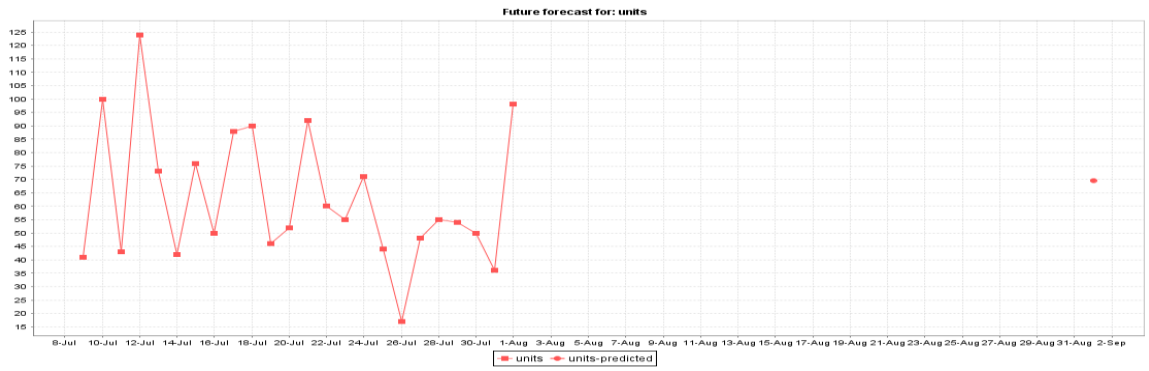


Figure 4.6: Regression Graph for 1-Step Monthly Test for Future Prediction

4.7 Regression Result for 7-Step Monthly

```
Relation:     item_units
Instances:    214
Attributes:   7
              Date
              item_nbr
              units
              store_nbr
              station_nbr
              tmp
              con

Transformed training data:

              units
              Date-remapped
              Lag_units-7

units:

Linear Regression Model

units =

      0.1081 * Lag_units-7 +
     67.762

=== Predictions for training data: units (1-step ahead) ===
```

Figure 4.7: Regression Result for 7-Step Monthly

4.8 Root Mean Squared Error for 7-Step Monthly

```
=== Evaluation on training data ===
Target                  1-step-ahead  2-steps-ahead  3-steps-ahead  4-steps-ahead  5-steps-ahead  6-steps-ahead  7-steps-ahead
==========================================================================================================================
units
  N                              183            182            181            180            179            178            177
  Mean absolute error        33.9322        33.7417        33.7588        32.9019        32.9909        33.1388        33.2945
  Root mean squared error    46.9509        46.8044        46.8782        44.8711        44.9784        45.1018        45.2271

Total number of instances: 190

=== Evaluation on test data ===
Target                  1-step-ahead  2-steps-ahead  3-steps-ahead  4-steps-ahead  5-steps-ahead  6-steps-ahead  7-steps-ahead
==========================================================================================================================
units
  N                               24             23             22             21             20             19             18
  Mean absolute error        24.4687        23.9978        23.7509        23.4352        22.4867        23.6501        23.0525
  Root mean squared error     27.599         27.2148        27.1099        26.9442        25.9303        26.6038        26.1019

Total number of instances: 24
```

Figure 4.8: Root Mean Squared Error for 7-Step Monthly

4.9 Regression Result for 7-Step Monthly Train at Steps

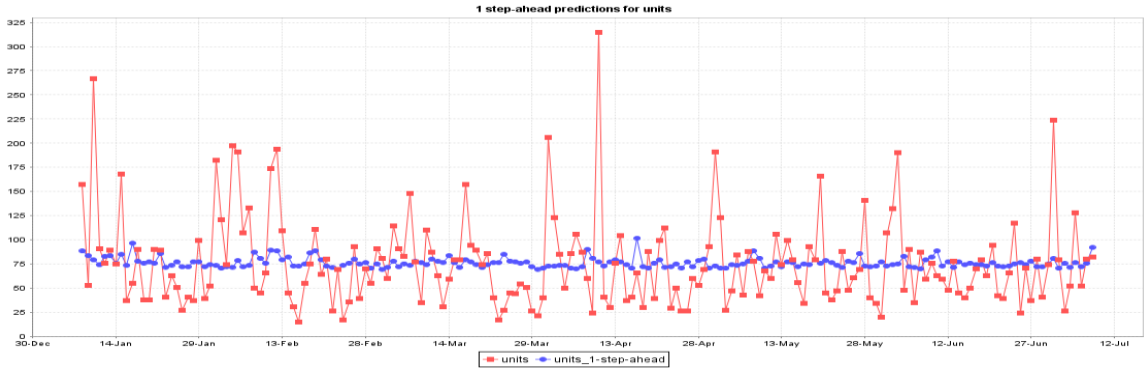


Figure 4.9: Regression Result for 7-Step Monthly Train at Steps

4.10 Regression Result for 7-Step Monthly Test Predict at Steps

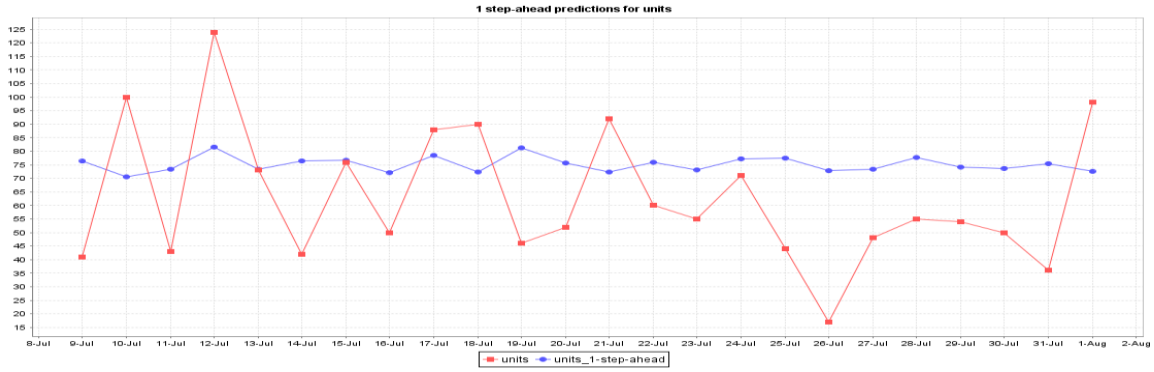


Figure 4.10: Regression Result for 7-Step Monthly Test Predict at Steps

4.11 Regression Result for Seven 7-Monthly Train for Prediction

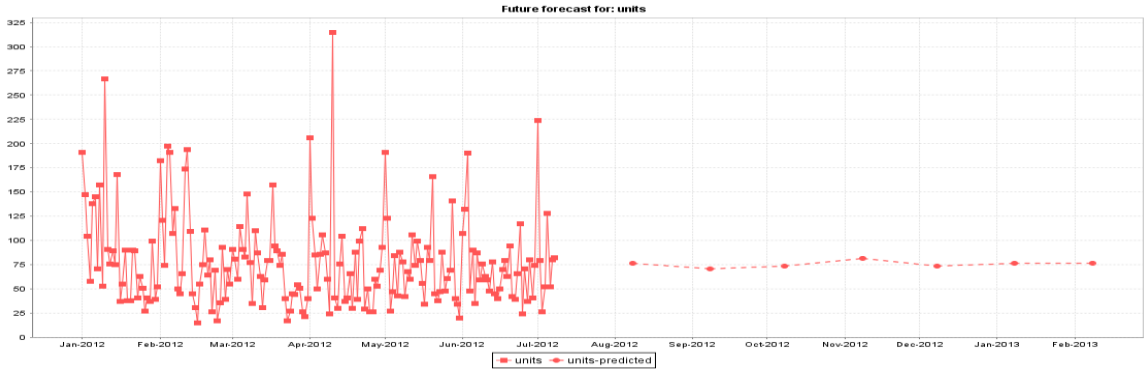


Figure 4.11: Regression Result for Seven 7-Monthly Train for Prediction

4.12 Regression Result for Seven 7-Monthly Test for Prediction

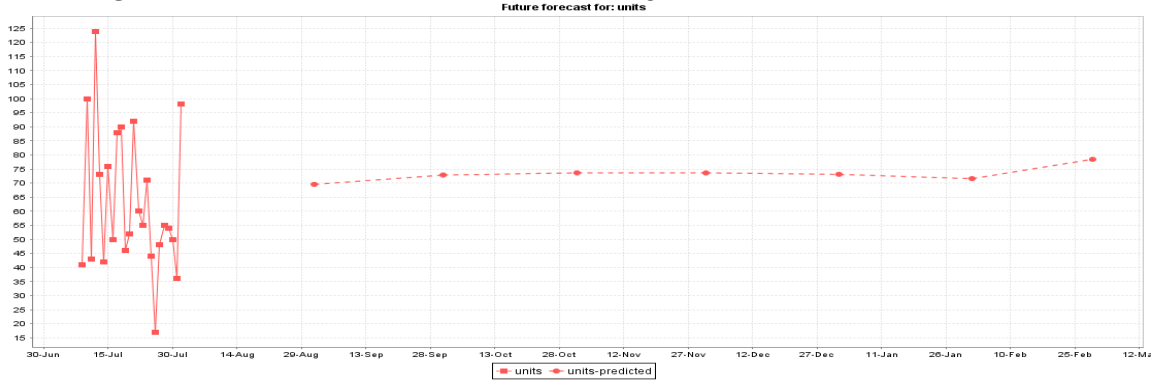


Figure 4.12: Regression Result for Seven 7-Monthly Test for Prediction

So Here is Our Final Equation….

units = 0.1081 * Lag_units-7 + 67.762

Here,

Lagged variables are the main mechanism by which the relationship between past and current values of a series can be captured by propositional learning algorithms.

CONCLUSION

5.1 CONCLUDING REMARKS:

Time Series Analysis and Regression Model appear to be a good solution to predict quantity of products at a retail store. However, it depends very much on the accurate weather forecast and consumer's mood swings. Mining data for a long period of time using a good analysis model leads to a better result for anything. That is why we have tried get a solution for the retailers so that they do not have to face much loss and wastage of products monthly or yearly. We have tried to find out a mathematical equation applying linear regression model and time series analysis, following which they can minimize their loss and maximize their interest in business.

5.2 Recommendation for future improvement:

Due to restrictions to time, we only conducted research on a few products, we will predict weathers effect on a large scale of products. We Have a plan to make an application in the future where every general user can enter a product and their store location, the app will predict the weather's effect on those automatically.

REFERENCES

[1] Martha Starr-McCluer **"The Effects of Weather on Retail Sales"**, https://www.federalreserve.gov/pubs/feds/2000/200008/200008pap.pdf

[2] Walmart Recruiting II: Sales in Stormy Weather (Apr 2, 2015) Retrieved from https://www.kaggle.com/c/walmart-recruiting-sales-in-stormy-weather

[3**]** Mark Hall (Mar 24, 2014) **"Time Series Analysis and Forecasting with Weka"**, Retrieved from https://wiki.pentaho.com/display/DATAMINING/Time+Series+Analysis+and+Forecasting+with+Weka

[4] **"Oracle® Data Mining Concepts11g Release 1 (11.1) B28129-02",** (Sep 2007) Retrieved from http://www.comp.dit.ie/btierney/Oracle11gDoc/datamine.111/b28129/regress.htm#CBBJIFEJ

[5] Kyle B.Murray , FabrizioDiMuro , AdamFinn , Peter Popkows kiLeszczyc **"The effect of weather on consumer spending "** http://www.sciencedirect.com/science/article/pii/S0969698910000822

[6] Christopher D. Carroll, Jeffrey C. Fuhrer and David W. Wilcox **"Does Consumer Sentiment Forecast Household Spending? If So, Why?"**

[7] Jeffrey A. **"Miron Seasonal Fluctuations and the Life Cycle-Permanent Income Model of Consumption"**

[8] Anita S. Harsoor, Anushree Patil **"FORECAST OF SALES OF WALMART STORE USING BIG DATA APPLICATIONS"**

[9] J. Davies & D. Elliott **"Analysing the impact of weather and climate on official statistics time series."**

[10] Nabilah Filzah Mohd Radula, Zalinda Othman, Azuraliza Abu Bakar **"Uncertain Time Series in Weather Prediction."**

[11] Kyle B.Murray , FabrizioDiMuro , AdamFinn , Peter Popkows kiLeszczyc **"The effect of weather on consumer spending "** http://www.sciencedirect.com/science/article/pii/S0969698910000822