MIDDLESEX UNIVERSITY
COURSEWORK 2

Summer, 2008/09

BIS4435

Industrial Data Management for Decision Support

Roman V Belavkin

This assignment is worth 20% of the overall grade. The submission date is **Monday, July 13, 2009**. You should do the assignment **individually**.

Contents

1 Overall Description

The aim of the coursework is to apply the data–mining techniques studied in the course to a real dataset and evaluate the results. An example dataset will be given in a separate file, but you may use any other dataset as long as it fulfils the requirements stated later (mainly, the dataset must be large enough to be worth analysing). You also can choose which method (or methods) of data–mining you will apply (e.g. a linear model, artificial neural network). However, you will have to justify your choice. The main goal of your data analysis should be the ability to **explain** and **predict** the data and possibly the phenomena behind it. Your work will be assessed based on a written report, which should include your results in each task, the summaries and analysis of these results and your conclusions. This report should also demonstrate your understanding of how the information and knowledge you found can be applied in economic or business context.

The coursework is divided into three tasks each of which is worth several marks out of total 100. In Task 4.1, you should explain the data and the data–miming technique you will use. In Task 4.2, you will have to evaluate the method on your data. In Task 4.3, you will require critically assess your results and suggest what advantages your analysis could give a company.

2 The Data

You are encouraged to find and choose a dataset that you will use in the coursework.[1] There are many data–mining resources available, which have links to some freely available datasets. For example, you can find free datasets at these two sites:

`fx.sauder.ubc.ca`	ForEx data at The University of British Colombia
`fimi.cs.helsinki.fi`	Frequent Itemset Mining Implementations Repository

For example, on the first site you can retrieve the daily exchange rates of top 10 currencies for some period of time. Table 1 shows an extract of such data for the last two years with British Pound as the base currency and using the price notation. The table has 12 columns (date, weekday and names of ten currencies) and many rows (for two years period there will be about 600 rows).

You can use a similar or different dataset for your coursework. It is important, however, that your data is

Multidimensional — each case (or each event) is described by several (more than one) variables. The variables (or dimensions) are usually the columns of your database table. Note that you can create a multidimensional dataset by using multiple copies of one variable, but at different time moments, as used in autoregressive models.

[1] If this is your **resit** coursework, then you must use a different dataset on your second attempt.

Table 1: Daily exchange rates of top 10 currencies in price notation. Base currency: GBP. Retrieved from PACIFIC Exchange Rate Service by Werner Antweiler, University of British Columbia.

Date	Wdy	USD	CAD	EUR	JPY	CHF	AUD	HKD	NZD	KRW	MXN
2006/01/03	Tue	0.57467	0.49665	0.68840	0.0049392	0.44410	0.42404	0.074115	0.39156	0.00057214	0.053986
2006/01/04	Wed	0.56853	0.49373	0.68752	0.0048879	0.44406	0.42441	0.073325	0.39118	0.00056927	0.053718
2006/01/05	Thu	0.56929	0.49005	0.68887	0.0049089	0.44575	0.42590	0.073422	0.39136	0.00057238	0.053759
2006/01/06	Fri	0.56508	0.48513	0.68641	0.0049396	0.44506	0.42527	0.072881	0.39043	0.00057100	0.053461
2006/01/09	Mon	0.56677	0.48475	0.68379	0.0049416	0.44297	0.42600	0.073123	0.39231	0.00057977	0.053662
2006/01/10	Tue	0.56689	0.48752	0.68370	0.0049449	0.44267	0.42444	0.073137	0.39279	0.00057722	0.053481
2006/01/11	Wed	0.56684	0.48950	0.68789	0.0049738	0.44520	0.42851	0.073128	0.39512	0.00057565	0.053404
2006/01/12	Thu	0.56785	0.48847	0.68337	0.0049673	0.44153	0.42639	0.073258	0.39351	0.00058275	0.053732
2006/01/13	Fri	0.56448	0.48607	0.68327	0.0049322	0.44082	0.42517	0.072821	0.39372	0.00057162	0.053371
2006/01/16	Mon	0.56619	0.48902	0.68614	0.0049244	0.44246	0.42701	0.073022	0.39498	0.00057607	0.053646
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
2007/12/17	Mon	0.49582	0.49222	0.71239	0.0043798	0.43025	0.42484	0.063584	0.37360	0.00053111	0.045683

Representative — the dataset must be large enough (i.e. have sufficient number of cases) so that your analysis had some statistical significance. The cases (or events) are usually the rows of your database table.

Thus, we need data that can be displayed in a table with several columns and usually large number of rows (at least greater than the number of columns). Typical datasets used in data–mining have dozens of variables (columns) and hundreds or even thousands of cases (rows).

Another important requirement is that your data must be **numerical**. For example, weekdays in Table 1 are represented by symbols. If you want to include weekdays in your analysis, they will have to be converted into numbers (e.g. 1, 2, 3 for Mon, Tue, Wed). Many data–mining applications can do this automatically, but often you have to handle this yourself.

3 Software Required

The main software you will need for the coursework is Microsoft Excel, which is installed on the University computers. MS Excel has several function that can be used to analyse the data (e.g. multiple linear regression). In addition, you can use the neural network software called BrainCel developed by Promised Land Technology (`http://www.promland.com/`). BrainCel works as an Add–On to MS Excel, and it will be available in the teaching labs for the course. BrainCel comes with a tutorial and an example, which will be covered in the labs.

Alternatively, you can use Oracle datawarehouse, which has many data–mining functions.

It is advised to save the data file into a directory on your `H:` drive, where you will keep your work. In each task of the coursework, you will need to re–arrange the data in a variety of ways. So, it is a good practice to keep a backup copy the original file and use another copy for your work. Please, contact your class tutor, if you have any difficulties in downloading the data.

4 Tasks and Assessment

This section will explain the tasks that you are required to do in your coursework. In the first task, you should introduce the data you will analyse, the main objectives of your analysis and data–mining techniques that you will be using. In the second task, you will have to perform the analysis and report the main results. In the third task, you will have to draw conclusions based on your results, and suggest how these results can be exploited in economic terms (e.g. how your results can support decisions in a business situation). In each task, you can gain up to 30% of the whole mark. In addition, up to 10% of the mark will be given to the style and presentation quality of your report.

4.1 Introduction of data, problem and methods (30%)

Choose, retrieve and prepare your dataset. For example, you can use the websites, mentioned in Section 2. This dataset should be saved in a file format suitable for the application that will be used to analyse it. For example, if you are using Microsoft Excel, then you can download the file in MS Excel format or CSV spreadsheet.

In your report, introduce the data as follows:

1. Explain what is the name of the dataset and what is its source. Do not forget to give appropriate credit to the owners/creators of the dataset.

2. Show in a table a sample of the data, such as shown in Table 1. Make sure that you do not print the whole database in your table, as it can be too large (e.g. your dataset may contain more than thousand of rows). It is sufficient to show only the first 10 rows of the table as in Table 1.

3. Explain the **variables** in your data. You can mention how many variables there are, if they are numerical or not, what do the numbers represent (i.e. what are the units).

4. Explain the **cases**. You can mention how many cases there are, does each case refer to different variables in a particular time moment or different time moments and a particular variable (i.e. corresponding to regressive or autoregressive models).

After introducing the data, you have to explain what kind of information your analysis will be looking for. In particular, this information must point at a dependency between variables. You have to explain which variables you are expecting to depend on each other and why.

Finally, you have to report briefly on the methods used to analyse the data. For example, if you are building a linear model, then you have to explain the main concepts of linear regression and relate it to your dataset. You can choose to use several different methods and compare their results.

4.2 Data analysis and results (30%)

In this task, you should analyse the data using the methods you have selected, and experiment how good each method can **explain** the relationships within the data and how well it can **predict** the data. Thus, you will have to split the data into two parts:

- The first part is called the **training set**, and it usually consists of about 70% of randomly selected cases from your original dataset. The training set is used to train your model so that it could learn the relationships between the variables.

- The second part is called the **testing set**, and it is the remaining part of your data that was not used during training. The testing set is used to evaluate the error that your model makes when it tries to predict new data.

If you are using MS Excel and BrainCel, then you can use the function *random select and move* to randomly split your data into the training and testing sets. However, before you do that, you should prepare all the formulae in your spreadsheet. Some of the formulae that you may enter into your spreadsheet are:

- Linear regression formula if you are using linear models (see section A.2 in the Supplementary materials).

- Absolute and average error formulae if you are comparing predicted values with the actual ones (see section A.4). The average errors can be used to evaluate how well a model predicts the data and also to compare different models.

If you are using the BrainCel neural network, then you will also have to name the regions of your spreadsheet corresponding to the training and testing sets. This is explained in Section A.3.

In your report, you should explain how you performed the data analysis and report the results of your experiments. Your report should focus on

Explanation of the dependencies. You have to report the values produced by your model that you think explain some dependencies (or prove the absence of such). For example, high absolute values of the regression coefficients in linear models point at strong linear dependency between the variables. Similarly, high absolute values of the weights in neural networks suggest strong dependency.

Prediction of new data. You have to report the average absolute errors your models make when predicting the data in the testing set. You can compare this error with the errors made for the training set or errors produced by other models (methods) on the same data. This will allow you to conclude which method works better on your data.

Remember the performance of many data analysis techniques depends on the choice of **parameters**. For example, the performance of artificial neural networks depends on their topology, the learning rate parameter, the number of training cycles and so on. Therefore, if you are using parametric techniques, make sure you experiment with different values of these parameters and try to find the values that improve the performance.

4.3 Evaluation and analysis of the results (30%)

In Task 4.2, you should have obtained many results. These can be regression coefficients of linear models, weights of the neural network after training, average errors of prediction by different models and so on. In this task, you should demonstrate your ability to summarise these results, present the findings in a compact and understandable form and make evaluation of the results reflecting on how these results can help in making intelligent decisions. Your report has to include the following components:

Summary of the results This can be accompanied by a table or a chart. For example, you can show on one comparative chart the average errors of your models as functions the parameters. This will help you to compare the performance of these models.

Evaluation of the results Write what do you think about the results you obtained. Do your results support the hypothesis about the dependencies you have expected? What new knowledge about your data have you learnt due to the analysis? Do not be discouraged if you did not obtain the results you expected. Try to be neutral and honest in your evaluation.

Decision support You should think about the practical applications of your findings. For example, how could you use the new knowledge in a business setting? It may help you to think about the decisions a manager could make by using your model. For example, if your model can forecast with certain accuracy the exchange rates of the currencies, how can such a forecast be used to decide which currency to buy or sell?

4.4 Presentation (10%)

All reports in science or business should be well presented. This means at the very least producing a clear (typed or nicely hand–written) document with good spelling, grammar and easy to understand English. On top of that you should consider the length of your report. There is no word limit, but a useful report should be just long enough to describe the work and make the recommendations in a well reasoned way. A sensible limit is about 10 pages of typed text, and fewer pages are acceptable if you have written up your work well. Beyond this, you are probably being a bit too verbose. Also, if you are developing several models, you might like to think of a sensible way to present each of them and its evaluation. Tables, graphs, careful labelling and numbering are all well established and effective presentation tools. If you put a table or a figure in your report, always describe it in the text (i.e. say what this table or a figure shows).

Things to avoid are:

- Listing endless streams of data (do not print the tables with the whole dataset as they can be too large);
- Including graphs or diagrams that you do not describe in the text;
- Forgetting to label the axes on the charts;
- Using 3D charts to display 2D information.
- Including material irrelevant to the work.

Note also that you do not need to use coloured charts, as these can be quite expensive to print. A lot of of information can be displayed using black and white patterns or gradations of grey.

5 Assignment Submissions

Submit your report to your Student Office or Learning Support Centre on **Monday, July 13, 2009** by 16:00 hours. Do not include a disk or any other materials. Make sure that your work is clearly labelled with your name, your student number, your campus, the course and the name of the module leader. Also ensure that it is securely bound and that it is easy for me to open and write on each page. Failure to do these things will result in loss of marks for presentation.

A Supplementary Material

The main theoretical material for this coursework is described in units on linear models, feed–forward neural networks and clustering algorithms. This section provides additional information on how to create linear and artificial neural network models in MS Excel with the BrainCel add–on.

A.1 Tips on working with the data in MS Excel

If you use the MS Excel spreadsheet for your coursework, then this section will give you some advice on how to prepare and organise the data in the spreadsheet and how to do the main experiments with the models. Usually, all the variables in your dataset are distinguished either as the independent (input) or the dependent (output) variables. We shall denote the independent variables as x_1, x_2 and so on, while the dependent as y_1, y_2. In fact, for this coursework you can just consider one output variable, and we shall denote it simply as y. For example, if you are using ForEx data, as in Table 1, you can choose to predict EUR based on the values of all other currencies. In this case, variable y represents the values of EUR, and $x_1, \dots, x_m$ are the other currencies in the table. Usually, the output variable is placed in the last column of the table (column y) as shown below:

x_1	x_2	$\cdots$	x_m	y
X_{11}	X_{12}	$\cdots$	X_{1m}	Y_1
X_{21}	X_{22}	$\cdots$	X_{2m}	Y_2
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
X_{n1}	X_{n2}	$\cdots$	X_{nm}	Y_n

It is hoped that variable y depends on the input variables $x_1, \dots, x_m$, and this is why we refer to y as the 'dependent' variable. However, usually it is not known how much y depends on $x_1, \dots, x_m$, and even what kind of dependency it is. The goal of many data analysis and data miming techniques is to understand and simulate this dependency as good as we can. This is done by creating a model (mathematical model) of this dependency from the data available, which means that one tries to find a function $y \approx f(x_1, \dots, x_m)$ that computes the value of y from the values of $x_1, \dots, x_m$ (the sign $\approx$ here means 'approximately equals').

It may be a good idea to create not one, but several different models and compare their performance on the same data. For example, you can use a linear model and a neural network model. The following two sections will give you some tips on creating these models. Each model has some advantages and some disadvantages, and it is good to understand which they are from your own experience. For example, one model can give a better explanation of the dependency, while other can be more precise. Section A.4 will give you tips on how to evaluate the precision of the models by computing the errors and their average errors.

To test the predictive power of a model, you should split the data into two parts — one for training and one for testing the model. We shall call these two parts of data the *training set* and the *testing set*. However, before you split the data, it is advised to prepare the table with all the functions necessary to do the computations. These are the functions for the forecasts using the models and also all the functions to compute the absolute errors and the average errors. The table below is an example of such a table. It is advised that you first put all the formulae in the first row, and after making sure that they compute correctly, copy and paste them across the remaining table.

Table 2: An example of how you can prepare the table for an experiment.

x_1	x_2	$\dots$	x_m	y	Linear model	Abs.e	ANN model	Abs.e
X_{11}	X_{12}	$\dots$	X_{1m}	Y_1	?	?	?	?
X_{21}	X_{22}	$\dots$	X_{2m}	Y_2	?	?	?	?
X_{31}	X_{32}	$\dots$	X_{3m}	Y_3	?	?	?	?
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$				
X_{n1}	X_{n2}	$\dots$	X_{nm}	Y_n	?	?	?	?
Average error:						?		?

After you have prepared such a table, you can split it into the training and testing sets. Normally, we use about 70% of data for training and 30% for testing. Sometimes you can just use the upper 70% part of the table for training and the lower part for testing. However, in some datasets, such as describing time–series, such a split would mean that the data for training and testing has very different properties (e.g. they may refer to quite different time periods). Therefore, it is usually much better to split the data randomly. The BrainCel software provides a function called *Random Select and Move* under the Preprocessing menu which can split the table into two subsets. You are advised to use this function to split the data by moving randomly $\approx$ 70% of the data, which will be your training set, and use the remaining $\approx$ 30% for testing.

To split the table, you should select the whole table including all the formulae and define a name for it (Insert/Name/Define in MS Excel). You can use the named region to randomly select and move 70% of rows for the training set. The remaining 30% will be your testing set, and it will already contain all the necessary functions for computing the forecasts, the absolute errors and the average errors. The advantage of using this method is that all the models use exactly the same testing set to make their predictions. Thus, the comparison of their errors is more reliable than using different testing sets for each model.

A.2 Linear models in MS Excel

Models are used to explain and predict data. One of the most common type of models is the linear model. The main assumption of this model is that the relationship between the variables in the data can be explained by linear functions. A linear function between two variables describes a line (hence the name linear), linear dependency between three variables describes a plane, linear functions between more than three variables describe linear manifolds also known as *hyperplanes*.

You probably remember that a straight line is described by the following formula:

$$y = a + bx \ ,$$

where x and y are the pair of variables, and a and b are the parameters of the line. Variable x is sometimes referred to as the independent or the input variable, while y as the dependent or the output variable (the one we are trying to explain or predict based on x). Of course here, the choice which is the input and which is the output is purely symbolic (we can always invert the linear function $x = (y - a)/b$).

Parameter a is called the *intercept*, and it defines the point at which the line crosses the y axis (i.e. when $x = 0$). Parameter b is called the *slope* (or the *gradient*), and it defines how steep the line is. Positive b means that the line goes up (positive trend), while negative b means that the line goes down (negative trend). Parameters a and b uniquely define the line — when the numbers a and b are known, we can find the value of y for any value of x (this is how we make the prediction — compute y for the desired value x).

In order to build such a linear model, we need to know n data points — pairs of x and y values: (X_1, Y_1), (X_2, Y_2),..., (X_n, Y_n). Using this data, we can find intercept a and slope b of a line that 'fits' these points as good as possible. Here, we understand a good fit in terms of a small average error. The most simple way of building the linear model is the *mean square linear regression*, and MS Excel has several functions that are based on this method. These functions are `INTERCEPT` (to compute a based on several known data points) and `SLOPE` (to compute b), and they can be used for a linear model between two variables. You are advised to refer to the following website:
`http://www.bized.ac.uk/timeweb/crunching/crunch_analysis_illus.htm`

which has good instructions on how to use these functions in MS Excel. In addition, MS Excel provides a function `FORECAST`, which combines the computations of a, b and the forecast $y = a + bx$ in one step.

A linear function of multiple variables can be used to create a linear model for several variables. The linear function of m input variables defines a hyperplane, and it is described by the following formula

$$y = a + b_1 x_1 + \cdots + b_m x_m$$

As you can see, the only difference here is that now we have m slopes: b_1, b_2,...,b_m. To compute these parameters, we need at least $n = m + 2$ data points

x_1	x_2	$\cdots$	x_m	y
X_{11}	X_{12}	$\cdots$	X_{1m}	Y_1
X_{21}	X_{22}	$\cdots$	X_{2m}	Y_2
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
X_{n1}	X_{n2}	$\cdots$	X_{nm}	Y_n

This is why datasets should contain at least as many cases (rows) as the number of variables (columns) plus one.

The MS Excel Data Analysis package has the multiple **regression** function that computes parameters a (the intercept) and b_1,...,b_m (the slopes) based on the given dataset. These parameters can be used to build a linear model with m input variables as follows.

The regression function will ask you to input the data range of your spreadsheet containing the data (usually your training set). If everything is done correctly, the output of the regression function is a table containing values of the intercept and the slopes. You can copy these values and use them in the linear equation to value of the output variable. Note that if you are testing model's prediction, then you should compute the parameters using only the training set, but test the model's forecast on the testing set using the same parameters.

Below are the steps that will help you to create multiple regression model.

1. Prepare a column somewhere in your spreadsheet to store the parameters of the multiple regression model:

$$\begin{array}{c} a \\ b_1 \\ \vdots \\ b_m \end{array}$$

   It is advised to use a column because the Data Analysis package outputs the values in a column, and it is easier to copy and paste the values.

2. In your main table, add an extra column to enter the multiple linear formula for the prediction. The formula is

$$y = a + b_1 x_1 + \cdots + b_m x_m$$

   where m is the number of input variables, $x_1, \ldots, x_m$. In MS Excel, you can use relative references to the cells on the same row. Parameters a, $b_1, \ldots, b_m$ are the values computed by the regression function in step 1. In MS Excel, you use absolute references to the cells in Step 1, where you will copy the parameters.

If you are using the model only to explain the data, then you can simply use the regression function to compute the parameters based on the whole data. However, if you want also to evaluate how well can the model predict new data, then you should only use the training set to compute the model's parameters, and then use the model to predict the values in the testing set. The following procedure can be used:

1. Enter the linear regression formula in the extra column of your table, as explained above.

2. To evaluate the precision of your model, enter additional formulae, such as the absolute error and the average error, as explained in Section A.4.

3. When the table is ready and contains all the necessary formulae (including the other models if you are using them), you can randomly select and move the **training set** from the table (use the BrainCel preprocessing menu). The remaining table will be your **testing set**.

4. Use the regression function of the Data Analysis package to find the model parameters based only on the training set.

5. Copy and paste the values of the parameters of the model into the column you have prepared earlier. Your table with the testing set should automatically compute the predictions and all the errors.

By examining the regression coefficients of the model (parameters $b_1, \dots, b_m$), it is possible to evaluate how significant is the dependency between the input and the output variables. If the value of b_i is close to zero, then it suggests that there is almost no linear dependency between variable x_i and y. Remember, however, that lack of linear dependency does not yet imply independence (the variables can be related non–linearly). Conversely, high absolute values suggest strong linear dependency. Moreover, positive values mean positive correlation, while negative values mean anti–correlation between x_i and y. When making conclusions about the dependency, remember that correlation does not imply causation.

A.3 Neural networks as models

As you will learn in the course, artificial neural networks can be trained and learn the relationship between the input and the output variables. Thus, you can train a neural network on your data and use it as a model. Although each neuron in the network can implement a linear model, a network of several neurons with at least one hidden layer can simulate non–linear relationships. This can be an advantage as many real–world processes are non–linear.

Recall that a neural network usually takes several inputs and has several outputs. If you use a dataset with m input variables and only one output variable, then you need a net with m inputs and only one output. We can consider the network as some function of m variables that we shall use to model our data:

$$y = f(x_1, \dots, x_m)$$

This function is more complex than the linear regression model, described above, and it has more parameters. Indeed, neural networks can have different topologies (i.e. the number of layers and the number of nodes in each layer); the nodes in the network can use different activation functions (e.g. linear, step or sigmoid); the network can be trained with different values of the training parameters (e.g. the learning rate).

A neural network model can be created in MS Excel using the BrainCel software. You can create a network (called 'expert' in BrainCel) with m input nodes and one output node (to predict one variable based on m input values). The number of hidden nodes is up to you to decide.

Before you can train the network on your data and use it for prediction, you need to name the regions in your spreadsheet where you have the training and the testing sets. Please, refer to Section A.1 for information on preparing the data in MS Excel.

After the spreadsheet has been prepared and the network created, you may train the network on the training set (the 70% of your data) and make the predictions using the testing set (the remaining 30%). If you have earlier created all the formulae in your table for computing the errors, then the spreadsheet should also compute all the results automatically.

Because the neural model has many parameters, the results will depend greatly on finding good values of these parameters, and there is a lot of room for experimentation. In order to achieve better results with neural networks (e.g. small average errors of prediction), try to experiment with the following parameters:

- Network topology (i.e. the number of nodes in the hidden layers);
- Different transfer functions (linear or log);
- Different learning schedule (i.e. the learning rate parameter, the number of training cycles).

If the network predicts the data well, then it means it has learnt the relationships between the variables. You can evaluate these relationships by examining the weights of individual nodes. Like the regression coefficients in linear models, weights that have high absolute values suggest high dependency between the variables. For example, if variable x_i is connected to ith input of a particular node, and weight w_i corresponding to this variable has high absolute value after training, then it means that variable x_i has significant influence on the output of the node. This, however, does not mean yet that variable x_i is related significantly to the output variable y. This is because in neural networks the input variables are not connected directly to the output variables (as in linear models), but there are nodes of the hidden layers between the input and the output variables. For this reason, it is more difficult to understand the dependency between the variables in neural network models, even if they produce good results. Sometimes, however, it is possible to say which variables do not have significant influence. Indeed, if some weights after training have values close to zero, then it means that the variables connected to them have little if no influence on the output variables.

A.4 How to evaluate the models

The quality of the model can be assessed by computing the error between the known data (y) and model's prediction of this data ($f(x)$). The difference between the actual and the predicted value can be use to compute the *absolute error*:

$$c(y, f(x)) = |y - f(x)|$$

Here $|\cdot|$ means the absolute value (i.e. the positive part of the number). In MS Excel, you should use function `ABS` of the difference between the data and the model's forecast. The absolute error measures the distance between the data and the models' prediction (distance is always positive, hence the use of absolute values).

Sometimes it is convenient to show the error as a percentage of the data that the model predicted. You can do this by dividing the absolute error by the value of y and then instructing MS Excel to format the cell as a percentage:

$$Error\% = \frac{|y - f(x)|}{y} 100\%$$

You can decide yourself which format to use. However, make sure that you are using and comparing the errors represented in the same format.

In order to understand how the model performs in a long term, we need to compute the errors for many cases and then take its average (the mean error). The *average absolute error* over k predictions is computed as

$$Mean(|y - f(x)|) = \frac{|y_1 - f(x_1)| + \cdots + |y_k - f(x_k)|}{k}$$

In MS Excel, you can use function `AVERAGE` to find the mean error. By computing and comparing the average errors of different models you can evaluate which models perform better (i.e. make better predictions).