COMP 4332 Project 2

Predicting future malfunctional components of ASUS computer modules

Lo Tsz Hin (20111661),

Ng Kam Pang (20122165)

Thusitha Chandrapala (20064923)

1) Introduction

The goal of this project is to predict future malfunctional components of ASUS computer modules. A set of historical sales details and repair details are given as data.

The following sections will outline how the raw data was processed, and learning algorithms used to obtain the final prediction. The problem was hardly a machine learning problem, but rather a curve fitting problem.

The group tried to work on two solutions independently. In **Section 1**, we highlight the attempt using the statistical package R as a tool. In **Section 2** we built another solution separately using Microsoft Excel.

kaggle.com registration:

- Team names
    - Section 1: chi lo
    - Section 2: darthTC,

The PAKDD website provides sales and repair data. The summery of the available data as given in the website is quoted below.

- SaleTrain.csv : the historical sales data for the period January/2005 to February/2008
- Output_TargetID_Mapping.csv : Used to map the output IDs
- RepairTrain.csv : the historical repair data for the period February/2005 to December/2009
- SampleSubmission.csv : A sample submission file with all zero output. The prediction time period is January/2010 to July/2011.

2) Section 1: Using R for cleaning and modeling

a) Obtaining and cleaning data

The amount of preprocessing and data cleaning needed for the project is minimal. The steps are outlined as below.

Following preprocessing steps are applied for both sales and repair data. The 'dataframe' based utilities in R was quite useful in this work.

- The Field names are changed for convenience.
- The date and month information are extracted. First the text information is converted to integers. Then an index is created integrating both year and month using the formula t=year*12+month

Since only repair data logs were used in building the final models, following steps are carried out only for the repair data.

- For each module/component combination a block ID is created
- Assigning zero values for time periods without repairs.
- Creating a time index, starting from Jan 2010

b) Building decay models for predicting repairs

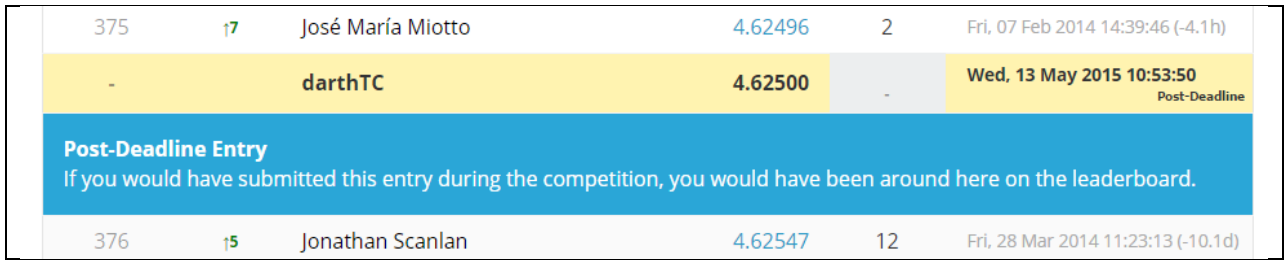
Using the sales data in the model might have been beneficial, but in our view since repair logs for a 5 month period is already given, whatever information embedded in sales log should be visible in the available repair log as well.

When the repair data for each module was observed after preprocessing, it seemed that the repair rate went down with time. So fitting a decay model to the available data seems to be the simplest and most efficient way of obtaining a prediction for the repairs in the given period. Decay models were considered for each component/module combination separately.

c) Following methods were tried

i) Holt-Winters model

This is a time series prediction model based on exponential smoothing. It even supports double-exponential smoothing to capture any seasonal effects present in the data. Fitting a Holt-Winters model did not give a good result.



Although this is a powerful model, since we only had 5 data points, we believe that the data is not adequate for this model.

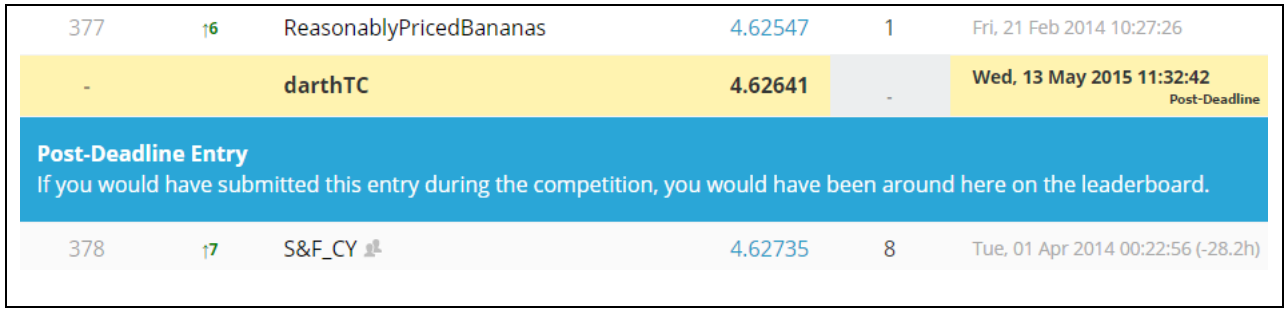
ii) Polynomial decay functions

Two decay functions were considered

- P1: $r = \frac{a}{b+t}$
- P2: $r = \frac{a+ct}{b+t}$

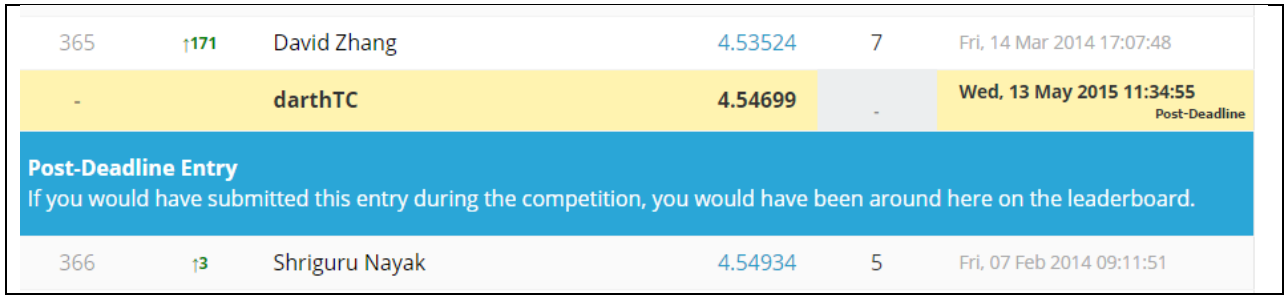
Where *r* is the number of repairs in a month, *t* is the month index, *a,b,c* being model parameters.

Since some of the data was not monotonically decreasing, the first model had issues with the fits. There were more than 50 instances of the gradient becoming zero while calculating the least squares estimates.



377	↑6	ReasonablyPricedBananas	4.62547	1	Fri, 21 Feb 2014 10:27:26
-		**darthTC**	**4.62641**	-	**Wed, 13 May 2015 11:32:42** Post-Deadline
Post-Deadline Entry If you would have submitted this entry during the competition, you would have been around here on the leaderboard.					
378	↑7	S&F_CY	4.62735	8	Tue, 01 Apr 2014 00:22:56 (-28.2h)

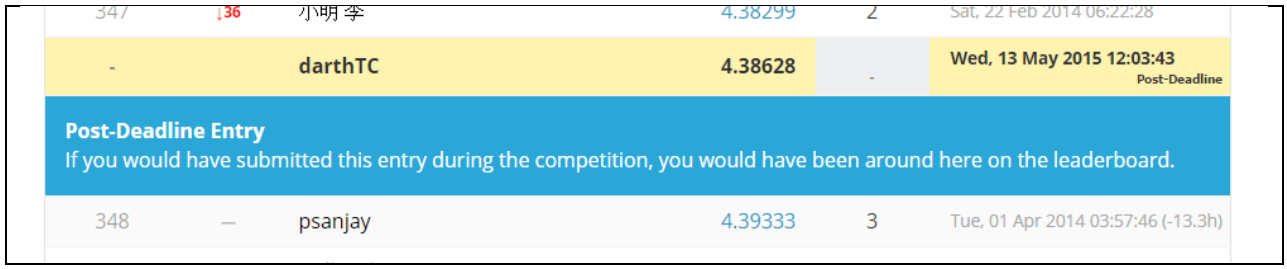
The second model is more robust with the fits. A slight performance increase could be seen as well.



365	↑171	David Zhang	4.53524	7	Fri, 14 Mar 2014 17:07:48
-		**darthTC**	**4.54699**	-	**Wed, 13 May 2015 11:34:55** Post-Deadline
Post-Deadline Entry If you would have submitted this entry during the competition, you would have been around here on the leaderboard.					
366	↑3	Shriguru Nayak	4.54934	5	Fri, 07 Feb 2014 09:11:51

iii) Fitting exponential decay

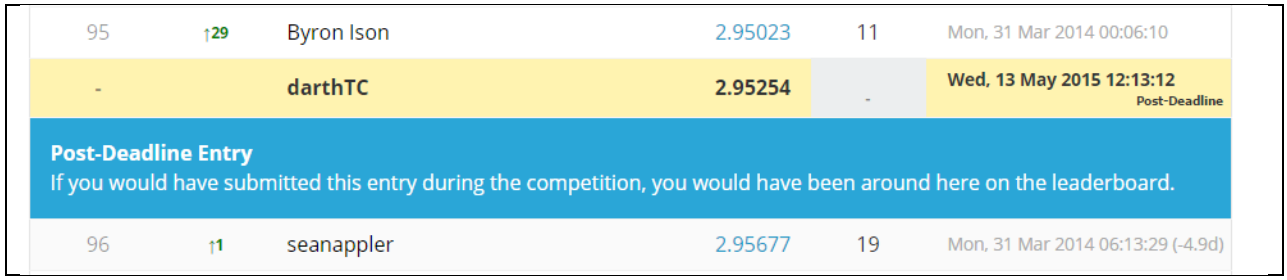
Exponential decays are commonly occurring in nature. Therefore it was decided to fit exponential decay functions to data. Fitting an exponential function is equivalent to fitting a linear model to the natural logarithm of the data. The 'lm' model in R could be utilized for this.



347	↓36	小明 李	4.38299	2	Sat, 22 Feb 2014 06:22:28
-		**darthTC**	**4.38628**	-	**Wed, 13 May 2015 12:03:43** Post-Deadline
Post-Deadline Entry If you would have submitted this entry during the competition, you would have been around here on the leaderboard.					
348	—	psanjay	4.39333	3	Tue, 01 Apr 2014 03:57:46 (-13.3h)

The high error rate with the exponential fit could be attributed to the fact that some fits were actually not decay functions, but results in exponentially increasing time series values. These fits were manually

adjusted to a predefined exponential decay rate. This decay is controlled by the “DEFAULT_DECAY” parameter, which was set of -0.05 based on the final result.



95	↑29	Byron Ison	2.95023	11	Mon, 31 Mar 2014 00:06:10
-		**darthTC**	**2.95254**	-	**Wed, 13 May 2015 12:13:12** Post-Deadline
Post-Deadline Entry If you would have submitted this entry during the competition, you would have been around here on the leaderboard.					
96	↑1	seanappler	2.95677	19	Mon, 31 Mar 2014 06:13:29 (-4.9d)

With this adjustment, we were able to obtain a score of 2.95, which placed us within the top 100 submissions.

3) Section 2 : Using Microsoft Excel

Since the amount of data was small, we were used to model using Microsoft Excel. First, we decided to observe how the data looks like. Below are our observations.

For each module category, there are different component categories.

module category	component category	Number of components
M1	02, 04-06, 09-13, 15-17, 19-22, 24-28, 30-31	23
M2	01-02, 04-17, 19-22, 24-31	28
M3	01-06, 09-13, 15-28, 30-31	27
M4	01-02, 04-06, 09-13, 15-17, 19-22, 24-28, 30-31	24
M5	01-02, 04-06, 09-13, 15-17, 19-22, 24, 26-28, 30-31	23
M6	01-06, 09-13, 15-17, 19-22, 24-28, 30-31	25
M7	01-06, 09-13, 15-17, 19-22, 24-28, 30-31	25
M8	01-02, 04-06, 09-13, 15-17, 19-22, 24-28, 30-31	24
M9	01-02, 04-06, 09-13, 15-17, 19-28, 30-31	25

We have to estimate how many products will require maintenance or repair services from 2010 Jan- 2011 Jul, which accounts for 19 months, for each of the module-component. Thus, there are 4256 estimates to be made in total.

After scanning through the dataset, we decide to give a few tests on the estimations.

a) Trials and errors

i) First try

We use the average of the data set in RepairTrain.csv for each module-component in each month, which is quite straightforward and we do not expect to get a good result. In the end, we get a score: 15.98880, which is quite bad comparing to the other competitors. It gives us a hint that the dataset does not follow a constant movement so we need to dip deeper into the dataset in order to find out the pattern of the data.

ii) Second try

We decide to check the range of the number of repairs so we try all-1 and all-0 estimates. Surprising, it gives a much better result than the average method, which get scores of 4.85855 and 4.62735 respectively. We get another hint that most estimates are 0 from the results.

iii) Third try

We decide to make a linear regression on the dataset. We regress on the whole dataset for each month. However, there are so many outliers that largely affect the results despite there are many 0 estimates which we presume it is correct. At last, we get a score about 9.93300.

iv) Fourth try

We decide to give different weights to each module-component in each month. And our target is to make the smallest estimate that we believe is close to the actual estimates. Thus, we give a higher weighting to the months which have smaller number of repair services. But still, we do not get a better result. We only scores 6.44258.

v) Fifth try

After so many tries, we decide to return to the simple method, which is the past data of last year. Surprising, the result turns out to be the best one. It scores 4.62171. After that, we use the data of 2008, which performs slightly better than the 2009 data. It scores 4.61748. During the process, we have set the outliners, which are very large, to 0. From this try, we can conclude that the true data follows a similar shape of the past two years.

b) Reconstruction of the model

After a few trial and errors, we can conclude that the above methods had reached a limit, which is about 4.61748 score. And the reason may be not using the SaleTrain.csv and the methods we used are not appropriate. Thus, we decide to build up a new model.

c) Formatting and Re-grouping the data

At the beginning, we regroup the data first.

We number the year and month from Jan 2005 to Dec 2009 as 1-60, which means Jan 2005 is numbered as 1, Feb 2005 is numbered as 2, and so on and so forth. The processed data is saved

in the Excel sheet called repairTrain.mc.cc.timerepair. Then, we calculate the total of repairs for each month in each year, which name as time.delta, for each module and each component using the RepairTrain.csv. The processed data is saved in the Excel sheet called repairTrain.mc.cc.timedelta. Moreover, we calculate the total of repairs for each month in each year for component using the RepairTrain.csv. The processed data is saved in the Excel sheet called repairTrain.cc.timedelta.

Similarly, we regroup the data in SaleTrain.csv into the same format as the RepairTrain.csv. We first number the year and month from Jan 2005 to Dec 2009 as 0-59, which means Jan 2005 is numbered as 0, Feb 2005 is numbered as 1, and so on and so forth. Then for each month and each year, we calculate the total sales for each module and each component. The processed data is saved in the Excel sheet called saleTrain.mc.cc.time. Next, we calculate the total sales for each component. The processed data is saved in the Excel sheet called saleTrain.cc.time.

d) Making Prediction

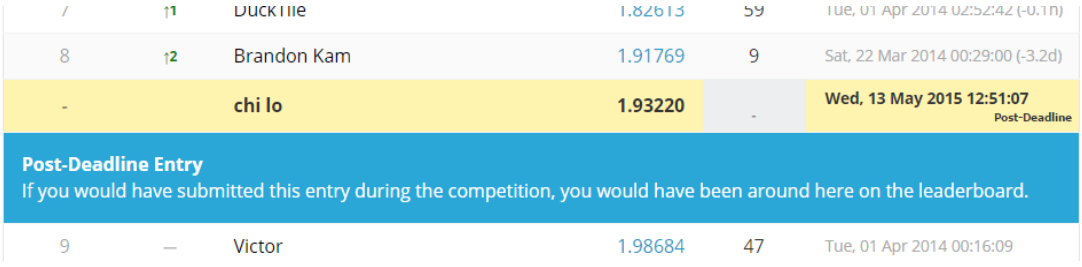
In the next step, we calculate the nerps for each component category by creating a relationship between the number of repairs and the accumulated number of sales after one month of the sale. Then, we create a coefficient by using the nerps for each component category to facilitate the prediction. The processed data is saved in the Excel sheet called Total.

Then, we calculate the probability of repairs needed by using the offset and indirect function in excel for each component in each time period. And we calculate a number called size for each component.

In the final step, we first calculate the repair volume of the last two months for each module and each component which believe is the most relevant data. Then, we make prediction by creating a relationship between the probability and the size calculated before.

Result

At last, we get a result around here which is quite excellent.



4) Code Explanation

Section 1: The R script which generates the submission files is submitted.

Section 2: The Microsoft Excel template for section is submitted

5) Online help and references

PAKDDcup online forum provided valuable insights into the project.

- https://www.kaggle.com/c/pakdd-cup-2014/forums/t/7574/shall-we-start-discussing-the-ideas/41511
- http://www.r-bloggers.com/exponential-decay-models/
- http://a-little-book-of-r-for-time-series.readthedocs.org/en/latest/src/timeseries.html

6) Work Breakdown

The codes and the report for Section 1 were carried out by Thusitha Chandrapala (20064923). The codes and report of Section 2 was carried out by Lo Tsz Hin (20111661), Ng Kam Pang (20122165).

7) Conclusion

Two separate attempts were made using two different platforms. Section 1 describes the approach using R to fit simple decay models to the repair data only. Section 2 describes how Sales and repair data were combined using Microsoft Excel to obtain better prediction accuracy. The second method achieves a score of 1.93 in the leaderboard.