Homework Assignment 1 - CSE 291 Spring 2011
Data Mining with Linear Regression

Jihoon Kim
UCSD Division of Biomedical Informatics
j5kim@ucsd.edu

Hari Damineni
UCSD Computer Enginering & Science
hdaminen@cs.ucsd.edu

André Christoffer Andersen
NTNU Industrial Economics
andrecan@stud.ntnu.no

April 12, 2011

1 Introduction

In this assignment, we build a linear regression model to estimate the donation dollar amount for a national fund raiser campaign. Our model is built in steps below:

1. Data pre-processing which includes feature manipulation, cleaning and transformation
2. Feature reduction using statistical analysis and 3-fold cross-validation
3. Linear regression

R is used to pre-process the data. Feature set is reduced by using univariate analysis as well as by building regression models on feature subsets using 3-fold cross validation to identify the features which lead to models with least RMSE. We used a Java program which iteratively generates feature subsets for input into the linear regression models. Finally, linear regression was employed to fit the pre-processed data using the Rapid Miner software package. 10 fold cross-validation with regularization was used to obtain a linear regression model with an average RMSE of 9.509.

Data type	Features	Note
Numeric	342	
Categorical	123	
Date	8	
Ordinal	5	
Outcome	2	TARGET_B, TARGET_D
Unique identifier	1	CONTROLN

Table 1: Number of features by data type in original data

2 Dataset

The data set in question is the KDD-CUP-98 data set form the The Second International Knowledge Discovery and Data Mining Tools Competition. The data set is a collection of data on people who have donated to a national veterans organization. The task is to predict donation amount in dollars associated with the response to form the 97NK fund raising campaign, i.e., the 1997 mailing campaign using "blank cards with labels". The original data set contains 481 features with 95412 instances. The data points are reduced to 4843 in order to use the information of only those donors who actually responded to the fund raising campaign.

3 Pre-processing

Pre-processing is a necessary step in model building to make training data usable for regression and reduce error by transforming and eliminating features. Regression requires that all data is represented as real numbers and that no data is missing. This naturally divides pre-processing in to three main steps: feature manipulation, data cleaning and feature selection.

Feature Manipulation Each feature goes through a thorough reviewed and is processed with heuristic and statistical methods. Six different types of manipulations are conducted:

1. *add*: introduces a new feature associated with a feature having at least one missing value, e.g., RAMNT_9 had 3788 missing values, thus a new binary feature MISSING_RAMNT_9 can be introduced with 3788 1's.

2. *aggregate*: collapses a multi-category feature into a binary feature, e.g., WEALTH1 has 10 ordinal values and can be replaced with WEALTH1_HIGH to have a value of 1 if WEALTH1 is greater or equal to 7 and otherwise 0.

3. *dummify*: replaces a categorical feature having k values with $k-1$ dummy (i.e. binary) features, e.g., RECENCY has 6 categorical values and can be further replaced

with 5 binary features.

4. *recode*: replaces a feature of date format to a numeric value based on date difference, e.g., MAXADATE can be replaced by MAXADATE_WEEKS using date difference in weeks from March 1st, 1997.

5. *remove*: discards redundant features, e.g., DOB can be removed as AGE already exists.

6. *split* : replaces a feature with multi-modalities by multiple separate features, e.g., feature RFA can be split into three features RECENCY, FREQUENCY and AMOUNT.

Data Cleaning Data cleaning process starts out by recoding all data to real values. Missing data is imputed by the mean of available values within the feature in question. After missing data is filled out we normalize with Z-scoring, sometimes called standardization. This is used in order to keep the data homogeneous across features. Because we are using a linear regression model using euclidean distance, or rather RMSE, to compute the model error all parameters should have the same scale for a fair comparison. We are, however, not normalizing binary features. This is done in order to maintain the a sparse data structure. Since we are doing both data augmentation and normalization with the whole data set and not just one training set at a time, there is some risk of information leakages through the data cleaning process — see section 6 for limitations.

Feature Reduction Once feature manipulation and data cleaning are complete, we use univariate analysis and systematic cross-validation on different feature subsets in order to reduce the set of features from the initial 481 features down to 30 most predictive features. Univariate analysis selects features useful for prediction. For binary feature, t-test is performed to compare the mean of outcome values between value-1-group and value-0-group. The cutoff for p-value is decided based on Bonferroni correction to overcome multiple testing problem. For numeric feature, Spearman correlation is used to screen predictive features. A feature is kept for model fitting if its absolute correlation is greater than 0.6.

4 Linear Regression using Rapid Miner

Rapid Miner (RM) is used for model building. First, the pre-processed data is loaded into the RMs data repositories. Later, a user process is created which connects the dataset to the input of a cross-validation operator. The cross-validation operator performs 10-fold cross validation by fitting linear regression models on the training data and applying the built model on the test data. Root Mean Squared Error (RMSE) is used as the error measure in our modeling. Regression coefficients and RMSE are obtained as the outputs of the RM process.

It should be noted that all the data processing is done outside of RM using R as we feel that R is more efficient in string processing and statistical analysis. Our RM process acts on the pre-processed data and performs modeling only.

Regularization Regularization penalizes large weights in the regression model. This has the effect of forcing a single solution and generating quicker learning convergence. This is especially needed when features could be (approximately) linearly dependent. We started with a regularization strength of 1E-8. Our regularization parameter tuning process involved using values with different increasing orders of magnitude. In the tuning process, we observed that values other than 1E-8 were yielding either similar or worse RMSE results. The reason for the no- or worse- effect of regularization strength could be because of our elaborate data cleaning and data reduction techniques which eliminated linearly correlated features and reduced the effective features to 30 and thereby increased the chances to obtain unique model co-efficients.

5 Results

Using the 10-fold cross validation, we obtained an average RMSE of 9.509. The result of the final model is shown in table 2. The features are sorted by the magnitude of T-statistic. Out of the 30 features, 13 are significant. While the first ten features are positively correlated to outcome, the last three are negatively correlated. `LASTGIFT` has the largest positive contribution to the outcome prediction. The average increase in outcome is 0.49 dollars per dollar increase in `LASTGIFT`, the most recent gift, with all other features held constant. On the negative side, `MINRAMT` had the largest contribution for predicton. The average decrease in outcome is 0.43 dollars per dollar increase in `MINRAMT`, the smallest gift to date, with all other features held constant.

6 Discussion

One obvious limitation to using linear regression is that we assume that all features have linear relationship with the target feature. This is of course only an approximation at best and could be outright wrong. However, regression does account for this by ignoring features that do not conform, that is, it sets that features weight. This is unfortunate if there is some kind of informative underlying non-linear feature that was not captured. We have tired to incorporate such features through feature transformation.

There might be some limited information leakage in that we performed normalisation and filling of missing data with means as a prepossessing step using all available data rather than as a step right before cross-validation. This can skew our results towards lower RMSE than would be achieved on entirely separated test data. Furthermore, we did not add a binary feature in order to capture latent information in missing data. This, however, would make the model error larger.

One of the techniques we used to reduce the feature set is to use a Java program to uniformly generate subsets of features. The Java program was run with the settings of generating 100 subsets of 30 features each. We ran 3-fold cross validation on linear regression models on each of these subsets to find the feature set which produces the lowest RMSE. While this heuristic produces a good feature subset, it does not guarantee the best one. Running the Java code with different settings could produce different feature subsets and the final regression could be different and could be worse if highly predictive features are left out. We used the Java based feature reduction technique in combination with the statistical technique so that the significant features as specified by the statistical analysis are always included in the feature subset. The combination of both heuristic and statistical techniques gave us good confidence in our feature reduction strategy and we observed that the loss in accuracy due to different settings of our Java program was not statistically significant.

Name	Coeff	Std. Error	Std. Coeff	Tolerance	**T-Stat**	P-value	Significance
LASTGIFT	0.49	0.01	0.44	0.39	33.31	0.00	***
AVGGIFT	0.20	0.02	0.17	0.51	11.49	0.00	***
RFAAMOUNT_2	0.37	0.03	0.12	0.42	10.75	0.00	***
RFAAMOUNT_6	0.13	0.02	0.05	0.63	5.40	0.00	***
MISSING_PEPSTRFL	0.94	0.26	1.14	0.93	3.56	0.00	***
RFAAMOUNT_8	0.08	0.02	0.03	0.63	3.27	0.00	***
RFAFREQUENCY_4	0.33	0.13	0.17	0.77	2.63	0.01	***
MISSING_LIFESRC	0.64	0.25	0.59	1.00	2.56	0.01	**
MISSING_RFARECENCY_16_F	0.72	0.33	1.57	0.98	2.16	0.04	**
MISSING_RFARECENCY_16_N	0.71	0.33	1.56	0.98	2.15	0.04	*
RFAAMOUNT_7	0.04	0.02	0.02	0.72	1.96	0.06	*
MISSING_RFARECENCY_14_A	0.54	0.35	1.26	0.99	1.54	0.16	
MISSING_RFARECENCY_14_L	0.53	0.35	1.25	0.99	1.54	0.16	
MAXRAMNT	0.01	0.01	0.01	0.63	1.29	0.27	
MISSING_RFARECENCY_18_N	0.37	0.32	0.76	0.99	1.18	0.33	
MISSING_RFARECENCY_18_F	0.37	0.32	0.74	0.99	1.16	0.35	
RFAAMOUNT_3	0.03	0.03	0.01	0.51	1.13	0.37	
MISSING_RDATE_23	0.24	0.41	0.08	0.99	0.59	0.56	
MISSING_RDATE_10	0.19	0.40	0.07	0.99	0.48	0.63	
MISSING_RAMNT_10	0.19	0.40	0.07	0.99	0.48	0.64	
MISSING_GARDENIN	0.15	0.34	0.07	1.00	0.45	0.66	
(Intercept)	-0.60	Infinity	NaN	NaN	0.00	1.00	
MISSING_RAMNT_18	-0.12	0.29	-0.07	0.98	-0.42	0.68	
MISSING_RDATE_18	-0.12	0.29	-0.07	0.98	-0.42	0.68	
MISSING_RDATE_8	-0.24	0.28	-0.16	0.98	-0.88	0.39	
MISSING_RAMNT_8	-0.25	0.28	-0.16	0.98	-0.89	0.38	
MISSING_RDATE_15	-0.51	0.41	-0.17	0.98	-1.24	0.30	
RFAFREQUENCY_2	-0.21	0.12	-0.11	0.75	-1.71	0.11	
RFAAMOUNT_4	-0.07	0.03	-0.03	0.52	-2.49	0.01	**
RFAFREQUENCY_3	-0.65	0.13	-0.33	0.77	-5.13	0.00	***
MINRAMNT	-0.43	0.02	-0.59	0.76	-27.94	0.00	***

Table 2: The results of the final model. Features are sorted by the descending order of T-statistic for clear representation of positive and negative features. ***: $P < 0.001$, **: $P < 0.01$ and *: $P < 0.05$

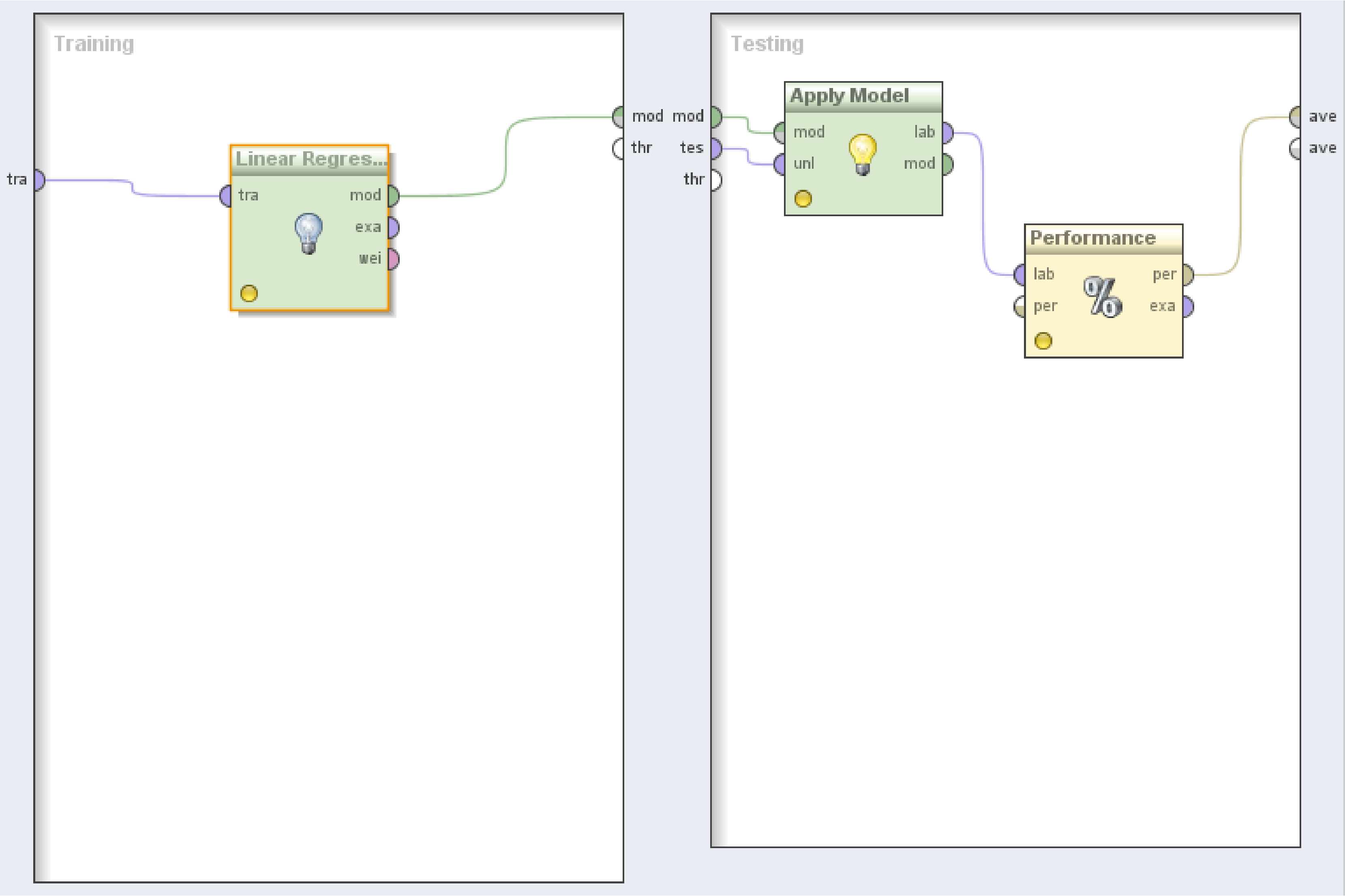

Figure 1: RapidMiner process tree of linear regression

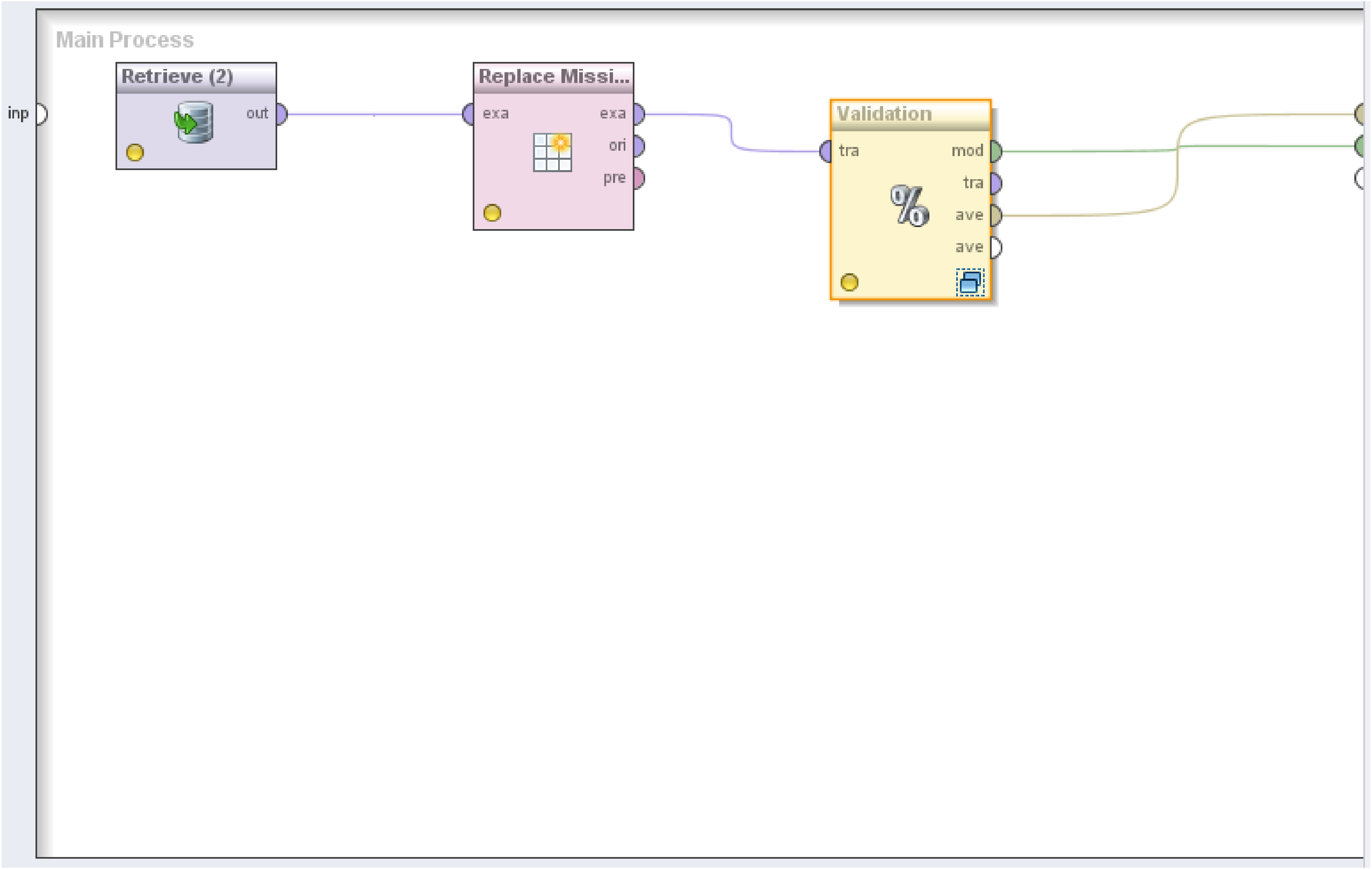

Figure 2: RapidMiner process tree of cross-validation