TIM 245 - Data Mining: Homework #2

Due: *June 3, 2017*

Instructor: Tyler Munger

Panos Karagiannis
ID: -

Contents

Problem 1: Exploratory Analysis and Data Cleaning

Use a combination of visual and quantitative tools to answer the following questions for the five input attributes (Xs) and the target (Y):

1. What is the typical value (central tendency)?
2. What is the uncertainty (spread) for a typical value?
3. What is a good distributional fit for the data (symmetric, skewed, long-tailed)?
4. Does the attribute affect other attributes (correlation)?
5. Does the attribute contain outliers (extreme values)?

It may be useful to format the answers as a table. Include any relevant plots and descriptive statistics in an appendix section.

Answer:

After removing the column `name` from our dataset we have the following attributes:
Xs $\rightarrow$ `economy`, `family`, `health`, `freedom`, `government_corruption`
Y $\rightarrow$ `happiness_score`

For each of the aforementioned attributes follow the answers to Questions $1-5$:

`economy`:

1. Typical Value: Calculating the mean for this data using R we get that the average is approximately 0.9539
2. Spread: The standard deviation (sample) for this data is 0.4125954. Also the mean absolute difference is $mad = 0.43276$
3. Distributional fit: Drawing a histogram of the data we see that the distribution looks *relatively* symmetric around the mean.
4. Correlation: The Pearson coefficient between `economy` and every other numerical attribute of this dataset is depicted in Table 1. We see that `economy` is most heavily correlated with family ($\sigma = 0.67$) and health ($\sigma = 0.84$).
5. Outliers: Based on the histogram there does not seem to be significant outliers. Also, the standard deviation as well as the *mad* are relatively small.

`family`:

1. Typical Value: Calculating the mean for this data using R we get that the average is approximately 0.7936
2. Spread: The standard deviation (sample) for this data is 0.2667. Also the mean absolute difference is $mad = 0.2817$

 continued on next page...

3. Distributional fit: Drawing a histogram of the data we see that the distribution looks skewed to the left. This means that there are few datapoints with small `family` values.

4. Correlation: The Pearson coefficient between `family` and every other numerical attribute of this dataset is depicted in Table 1. We see that `family` is most heavily correlated with economy ($\sigma = 0.67$) and `happiness_score` ($\sigma = 0.73$).

5. Outliers: Since the histogram is skewed there are outliers in this attribute.

`health`:

1. Typical Value: Calculating the mean for this data using R we get that the average is approximately 0.5576

2. Spread: The standard deviation (sample) for this data is 0.2293. Also the mean absolute difference is $mad = 0.2478$

3. Distributional fit: Drawing a histogram of the data we see that the distribution looks *relatively* symmetric around the mean.

4. Correlation: The Pearson coefficient between `health` and every other numerical attribute of this dataset is depicted in Table 1. We see that `health` is most heavily correlated with `economy` ($\sigma = 0.84$) and `happiness_score` ($\sigma = 0.77$).

5. Outliers: Even if the histogram is not skewed, we see that outliers do exist since the standard deviation as well as the *mad* are large relative to the mean.

`freedom`:

1. Typical Value: Calculating the mean for this data using R we get that the average is approximately 0.3710

2. Spread: The standard deviation (sample) for this data is 0.1455. Also the mean absolute difference is $mad = 0.1662$

3. Distributional fit: Drawing a histogram of the data we see that the distribution contains outliers. Therefore, this distribution is only *relatively* symmetric around the mean.

4. Correlation: The Pearson coefficient between `health` and every other numerical attribute of this dataset is depicted in Table 1. We see that `freedom` is **not** heavily correlated with any other attribute. For example, it is most heavily correlated with `government_corruption` ($\sigma = 0.50$) and `happiness_score` ($\sigma = 0.57$).

5. Outliers: Based on the histogram for this data we see that outliers exist in the attribute. This is also reinforced by the fact that the standard deviation as well as the *mad* are large relative to the typical value.

`government_corruption`:

1. Typical Value: Calculating the mean for this data using R we get that the average is approximately 0.1376
2. Spread: The standard deviation (sample) for this data is 0.1110. Also the mean absolute difference is $mad = 0.0792$
3. Distributional fit: Drawing a histogram of the data we see that the distribution is very skewed to the right.
4. Correlation: Based on Table 1 this attribute does not seem to be heavily correlated with any other attributes. It most heavily correlated with `freedom` and `happiness_score`.
5. Outliers: Based on the histogram for this data we see that outliers exist in the attribute. This is also reinforced by the fact that the standard deviation as well as the *mad* are large relative to the typical value.

`happiness_score`:

1. Typical Value: Calculating the mean for this data using R we get that the average is approximately 5.382
2. Spread: The standard deviation (sample) for this data is 1.141. Also the mean absolute difference is $mad = 1.371$
3. Distributional fit: Drawing a histogram of the data we see that the distribution is *relatively* symmetric around the mean.
4. Correlation: Based on Table 1we see that `happiness_score` is most heavily correlated with `economy` ($\sigma = 0.79$), `family` ($\sigma = 0.73$) and `health` ($\sigma = 0.77$).
5. Outliers: Based on the histogram for this data we see that outliers do not exist for the attribute. This is also becomes apparent from the fact that the standard deviation and the *mad* are small compared to the mean.

Table 1: Upper Triangular Correlation Matrix

	economy	family	health	freedom	government_corruption	happiness_score
economy	1	0.67	0.84	0.36	0.29	0.79
family	-	1	0.59	0.45	0.21	0.73
health	-	-	1	0.34	0.25	0.77
freedom	-	-	-	1	0.50	0.57
government_corruption	-	-	-	-	1	0.40
happiness_score	-	-	-	-	-	1

$\Longrightarrow$ All histograms are included in the Appendix.

Problem 2: Prediction

Based on the results from the Exploratory Data Analysis, how well does the data-set fit the assumptions of linear regression? For example, is the data normally distributed?

Answer:

By definition, linear regression attempts to estimate the type and strength of linear relationships in the training data set. Given our previous analysis, as well as the scatter plots we produced, we see that the **Xs** which have a strong linear relationship with `happiness_score` (Y) are: `economy`, `family` and `health`. This is expected as these attributes have high correlation coefficients with `happiness_score` i.e. $\sigma = 0.79, 0.73, 0.77$, respectively. Nevertheless, the attributes `freedom` and `government_corruption` seem to not be linearly correlated with `happiness_score`.

$\Longrightarrow$ All scatter plots are included in the Appendix.

Problem 3: Prediction

What is the interpretation of the coefficients of the linear model? For example, which attributes have the strongest correlation with happiness? Does the model make sense?

Answer:

When we compile the model using the command:

```
fit.lm(happiness_score) .,data =train)
```

we get the following best fit straight line:

$$happiness\_score = \begin{bmatrix}\beta_0 & \beta_1 & \beta_2 & \beta_3 & \beta_4 & \beta_5\end{bmatrix} \begin{bmatrix} 1 \\ economy \\ family \\ health \\ freedom \\ government\_corruption \end{bmatrix}$$

Since we are taking a random sample in order to test our data we see that every time we run the above command we get quite different results. For example:

$$happiness\_score = \begin{bmatrix}2.208 & 0.604 & 1.22 & 1.596 & 1.64 & 0.91\end{bmatrix} \begin{bmatrix} 1 \\ economy \\ family \\ health \\ freedom \\ government\_corruption \end{bmatrix}$$

Problem 3: Prediction continued on next page...

$$happiness\_score = \begin{bmatrix} 2.352 & 0.769 & 0.881 & 1.767 & 1.150 & 1.04 \end{bmatrix} \begin{bmatrix} 1 \\ economy \\ family \\ health \\ freedom \\ government\_corruption \end{bmatrix}$$

,are both results given by our regression model but for different samples. This might be because of the fact that we are only training on 66% of the total data ($\sim$ 100 data points). Hence, there is a lot of variability in choosing our sample.

The coefficients indicate the strength and type of the relationship. Firstly, we see that all coefficient are positive which is expected based on the correlation matrix (Table 1).

Moreover, the greater the magnitude of a coefficient, the greater the correlation between the corresponding attribute and the dependent variable. In this model we see that `government_corruption` has a small coefficient which is expected since as we discussed earlier this attribute has a very small correlation with `happiness_score`. Nevertheless, based on our previous analysis we would expect the coefficient of `economy` to be much larger than the coefficient of `freedom` because the latter is very weakly correlated to `happiness_score`.

Finally, based on the magnitude of the coefficients we see that the attributes that have the strongest correlation to happiness are: `freedom`, `health`, `family`.

Problem 4: Prediction

Provide an assessment of the model's performance. Do you think the model's performance is good? Explain why.

Answer:

Using R we get that $MSE = 0.32$ while the minimum and maximum value are 3.592 and 7.355, respectively. This signifies that the predicted value is not very inaccurate since the MSE is small compared to the values of the coefficients. This is also reinforced by the fact that RAE is small (0.08). Moreover in order to create a baseline for our model, we create a simple linear regression model where we use only one attributes and pick then pick the model with the minimum *MSE*. In this case we choose the attribute `economy`, whose $MSE = 0.46$. Therefore, we see that our model performs better than the baseline.

Problem 4: Prediction continued on next page...

Problem 5: Prediction

How are the coefficients different for the linear model created using lm()? .Experiment with lambda values: $0.0005, 0.005, 0.5, and 5$. How do the coefficients change as we adjust the value of lambda up and down?

Answer:

As expected, when we use the more sophisticated linear regression methods the coefficients' magnitude decreases compared to the simple `lm()` method. This is logical since *lasso, ridge and elastic nets* all use regularization methods because their loss function penalizes the magnitude of the coefficients. As we increase the parameter α the general trend is that the magnitude of the coefficients decreases more and more. For example the coefficient of `economy` becomes: $0.6195772, 0.6193755, 0.5958423, 0.5958423$ for $\alpha = 0.0005, 0.005, 0.5, 5$, respectively.

Problem 6: Prediction

Which model would you recommend to a nation that is trying to predict the happiness of their citizens?

Answer:

The model that I would suggest in predicting the happiness of the citizens would be the *elastic net* model. *Elastic net* is a hybrid model that uses parts from both lasso and ridge regression. By combining both the $L1$ and $L2$ norm in the penalty function, *elastic net* could give promising results. This is also reinforced by the fact that the MSE for *elastic net* is 0.3512 ,whereas, for *lasso* and *ridge* the MSE is $0.3524, 0.3522$, respectively.

Problem 7: Prediction

What is the interpretation of the selected model? What can we say about the relationship between the input attributes and the happiness score?

Extra Credit:
Show that $coef(lm(\mathbf{Y} \sim \mathbf{X}))$ is equivalent to $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}$

Answer:

The model that we have created can be used to predict the happiness of people based on 6 predefined attributes. The coefficients of our model tell us the mean change in the response variable (*happiness*), for one unit of change in the predictor variable while holding other predictors in the model constant. Moreover, the $y-$ intercept in our model is approximately 2.35 which signifies the baseline happiness i.e. when all other factors are 0.

Extra Credit:

We prove this mathematically. Let $\boldsymbol{Y} \in \mathbb{R}^n, \boldsymbol{X} \in \mathbb{R}^{n\times k}$, $\boldsymbol{\beta} \in \mathbb{R}^k$. Then, in the context of linear regression, we need to find the coefficients $\boldsymbol{\beta}$ that minimize our error function. In this case let the error function be $\mathcal{L} : \mathbb{R}^n \to \mathbb{R}$ with formula:

$$\mathcal{L}(\boldsymbol{\beta}) = \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 = (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^T)(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})$$

Our objective is to minimize the loss w.r.t. $\boldsymbol{\beta}$ hence we need to find $\hat{\boldsymbol{\beta}}$ such that $\mathcal{L}$ is minimum.
Before taking the derivative we expand the expression for the loss:

$$(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^T)(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}) = \tag{1}$$
$$\boldsymbol{Y}^T\boldsymbol{Y} - (\boldsymbol{X}\boldsymbol{\beta})^T\boldsymbol{Y} - \boldsymbol{Y}^T\boldsymbol{X}\boldsymbol{\beta} - (\boldsymbol{X}\boldsymbol{\beta})^T\boldsymbol{X}\boldsymbol{\beta} = \tag{2}$$
$$\boldsymbol{Y}^T\boldsymbol{Y} - 2\boldsymbol{Y}^T\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{\beta}\boldsymbol{X} \tag{3}$$

In order to compute the derivative of (3) we need to show the following results:

Let $f(\boldsymbol{\beta}) = \boldsymbol{c}^T\boldsymbol{\beta}$. Then:

$$f = \sum_{i=1}^{k} c_i\beta_i$$

Hence:

$$\frac{\partial f}{\partial \beta_w} = \sum_{i=1}^{k} c_i \frac{\partial \beta_i}{\partial \beta_w} = c_w$$

So:

$$\nabla_{\boldsymbol{\beta}} f = \boldsymbol{c} \tag{4}$$

Now let the more interesting function $f(\boldsymbol{\beta}) = \boldsymbol{\beta}^T\boldsymbol{A}\boldsymbol{\beta}$, for A k-dimensional square matrix. Then:

$$f = \sum_{i=1}^{k}\sum_{j=1}^{k} \alpha_{ij}\beta_i\beta_j =$$

Differentiating this expression we get:

$$\frac{\partial f}{\partial \beta_w} = \sum_{i=1}^{k}\sum_{j=1}^{k} \alpha_{ij}\frac{\partial(\beta_i\beta_j)}{\partial \beta_w} = \sum_{i=1}^{k}\sum_{j=1}^{k} \alpha_{ij}(\delta_{i,w}\beta_j + \delta_{j,w}\beta_i) =$$
$$\sum_{j=1}^{k} \alpha_{wj}\beta_j + \sum_{i=1}^{k} \alpha_{iw}\beta_i = (\boldsymbol{A}\boldsymbol{\beta})_w + \sum_{i=1}^{k} (\boldsymbol{A})^T_{w,i}\,\beta_i = (\boldsymbol{A}\boldsymbol{\beta})_w + (\boldsymbol{A}^T\boldsymbol{\beta})_w$$

Hence we gey that:

$$\nabla_{\boldsymbol{\beta}} f = (\boldsymbol{A} + \boldsymbol{A}^{\boldsymbol{T}})\boldsymbol{\beta} \tag{5}$$

Finally, we are in place to differentiate (3) using (4), (5). Let $\boldsymbol{Y}^T\boldsymbol{X} = \boldsymbol{c}$ and also let $\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{A}$. Then we get:

$$\nabla_{\boldsymbol{\beta}}\mathcal{L} = \mathbf{0} - 2\boldsymbol{Y}^T\boldsymbol{X} + (\boldsymbol{X}^T\boldsymbol{X} + (\boldsymbol{X}^T\boldsymbol{X})^T)\boldsymbol{\beta} = \tag{6}$$

$$-2\boldsymbol{Y}^T\boldsymbol{X} + 2\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta} = 2(\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{Y}^T\boldsymbol{X}) \tag{7}$$

Equating the derivative with $\mathbf{0}$ and solving the equation we get $\hat{\boldsymbol{\beta}}$:

$$\boldsymbol{X}^T\boldsymbol{X}\hat{\boldsymbol{\beta}} - \boldsymbol{X}^T\boldsymbol{Y} = \mathbf{0} \iff \boldsymbol{X}^T\boldsymbol{X}\hat{\boldsymbol{\beta}} - \boldsymbol{X}^T\boldsymbol{Y} = \mathbf{0} \iff$$

$$\iff \boldsymbol{X}^T\boldsymbol{X}\hat{\boldsymbol{\beta}} = \boldsymbol{X}^T\boldsymbol{Y} \iff \hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}$$

To verify these results experimentally is straighforward. We use `X<-data.matrix(data[0:5])` and then set the first column to be 1's to define $\boldsymbol{X}$. Then, do `Y<-data.matrix(data[6])` to set table $\boldsymbol{Y}$. To compute the transpose we do `X_T<-t(X)`, whereas the inverse of the product is computed using `solve( X_T %*% X)` . Putting everything together we get:

```
b_hat= solve( X_T %*% X) %*% X_T %*% Y
```

In the following classification problems, we filter out the costumer id since it explodes our feature space and does not contribute to our classification procedure.

Problem 1: Classification

How does the ZeroR model work? How is the ZeroR useful when creating a baseline?

Answer:

The ZeroR model is one of the simplest ways to perform classification. In this classifier, we completely ignore the attributes and we simply always predict the majority class. Due to its simplicity and ease of implementation it is often used as a benchmark for other classification methods.

Using 10-fold cross validation in our example, we see that by using ZeroR we classify correctly 73.463% percent of the instances. Moreover, the F measure is 0.622

Problem 2: Classification

Compare and contrast the performance of the three models. Do the Naive Bayes and Decision Tree models significantly outperform the ZeroR model? Explain why or why not.

Answer:

When we compute the *Naive Bayes* we get a bit worse accuracy but better F-measure results than the ZeroR method (Accuracy=71.9296%, $F = 0.735$). Therefore, we see that

Naive Bayes does not significantly outperform our baseline. One reason for this result, might be the independence assumptions that Naive Bayes makes to compute the posterior distribution. It is very likely that the attributes of our datasets are highly correlated therefore Naive Bayes performs poorly.

When we use *Decision Trees* we get that the accuracy is 77.9781%, whereas, the F measure is 0.772. As a result we see that this method outperforms both *ZeroR* as well as *Naive Bayes*. A possible reason for this result, might be the fact that Decision Trees do not make any independence assumptions (as in Naive Bayes) and also do not ignore the attributes (as in ZeroR).

Problem 3: Classification

How does the performance of the K-NN model compare to the baseline models from Experiment 1?

Answer:
Using the *K-NN* method with $K = 10$ we see it surpasses all methods used in previous questions in terms of both accuracy and F measure, except the *Decision Trees*. More precisely, the accuracy we get for *K-NN* is 77.5806% whereas the F-measure is 0.767.

Problem 4: Classification

Should the input attributes be normalized? Explain why or why not.

Answer:
The attributes should definitely be normalized. For instance, the `Total Charges` are in the range of $[0, 10^3]$ whereas `tenure` ranges from $[0, 10^2]$. Since we used the Euclidean distance in our model, the term differences of the `Total Charges` would dominate the sum. As a result, the "effect" of the `tenure` in our model would be diminished.

Problem 5: Classification

Experiment with the following values for K=3,10,50,100. What effect does changing K have on the model performance? What is the optimal value for K? Explain why.

Answer:
We see that as we increase the value of K our results keep getting more accurate up to a point and then start getting more inaccurate. This fact is attributed to overfitting our test data. Below follow the measures we get as we increase K:

1. $K = 1$: Accuracy=71.4894% , F-measure=0.717
2. $K = 10$: Accuracy=77.5806% , F-measure=0.767
3. $K = 50$: Accuracy=79.3554% , F-measure=0.791

Problem 5: Classification continued on next page...

4. $K = 100$: Accuracy=79.327% , F-measure=0.790

As we increase K, both the accuracy as well as the F-measure get better, nevertheless, when K reaches 100 we have **overfitted** the data therefore our results become worse. From the above list, the optimal value of K is 50 (given these results, above we could also do a binary search to find the optimal K in logarithmic time).

Problem 6: Classification

Compare and contrast the linear models (Logistic Regression and SVM) to the baseline and the non-linear models from the previous experiments?

Answer:
Both models *Logistic Regression* and SVM as discriminative methods therefore they do not make any underlying assumptions for the distribution of the data points and also minimize a loss function in order to find the optimal weights. This means that we would expect them to outperform the previously used methods. Running these tests in $Weka$ we see the following results:

1. *Logistic*: Accuracy=80.2215%, F-measure=0.809
2. SVM: Accuracy=80.2641%, F-measure=0.810

Problem 7: Classification

Which model performs better: logistic regression or SVM? If there is a difference, explain why.

Answer:
Both models behave very similarly, but we see that the SVM model slightly outperforms *Logistic Regression*. Both these methods minimize convex loss functions and hence any local minimum is also guaranteed to be a global minimum as well. The reason that the SVM has a bit better accuracy is because it uses the *hinge* loss which punishes misclassification, whereas, the logarithmic loss (used in Logistic Regression) leads at better probability estimation at the cost of accuracy.

Problem 8: Classification

Compare and contrast the Random Forest model with the results from Experiments 1, 2, and 3.

Answer:
Running the *Random Forest* model in Weka we get the following results: Accuracy=77.4790% and F-measure= 0.763. The idea behind a Random Forest is the use of many Decision Trees to classify a given vector. Therefore, this model is very similar to Decision Trees and does not make any assumptions about the underlying distribution of the data points as in Naive Bayes. We see that the *Random Forest* outperforms *ZeroR* and *Naive* Bayes, nevertheless,

has very similar results to *Decision Trees*.

Moreover, comparing *KNN* with *Random Forest* depends on the value of the parameter K that we choose for our *KNN* model. For $K < 10$, we observe that *Random Forest* outperforms *KNN*, whereas, the opposite happens when $10 \leq K \leq 100$.

Finally, we see that both *Logistic Regression* as well as *SVM* outperform *Random Forest*. This might be due to the power of the methods used in Problem 3 in minimizing a loss function to find the optimal weights.

Problem 9: Classification

What is the difference between a Random Forest and the decision tree model from Experiment 1. Would you expect the Random Forest model to outperform the decision tree model? Explain why.

Answer:
We would expect that the Random Forest would outperform the Decision Trees since in the former model we operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes. Nevertheless, this is not verified by our experimental results. As we observe, both models perform very similarly in both their Accuracy as well as their F-measure. This observation might be because during the *Random Forest* model we generate many *Decision Trees* which only "mislead" us and introduce noise to the final decision.

Problem 10: Classification

What model do you recommend that the telecommunications company use to predict churn? Explain why.

Answer:
The model I would suggest to the telecommunications company to use in order to predict churn would be *SVM*. There are two basic reasons as to why I would suggest the SVM model against any other model considered in this assignement.

Firstly, as we saw experimentally the SVM outperforms most of the models that we used, in both accuracy and F-measure. This is mainly because it is not forced to make any assumptions about how the data were distributed or generated. Solving a loss function to minimize the margin is a powerful technique that inevitably yields good results.

Secondly, the true power of $SVMs$ becomes apparent when the telecommunications company would like to incorporate much more features (attributes) into their model. Then instead of solving the *Primal Problem* we can equivalently solve the *Dual Problem*.

For example let the dimension of our attributes be D. Then solving the Primal problem would be much more inefficient in the case where $D \gg N$. This is obvious from the formulas of the two equivalent problems:

Primal:

$$\begin{aligned} \underset{w,b}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{w}\| + \frac{C}{M}\sum_{i=0}^{M}\xi_i \\ \text{subject to} \quad & t_i(<\mathbf{w}, x_i> +b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{aligned}$$

In the Primal we see that we optimize over $\mathbf{w}$ which is a D-dimensional vector. Therefore if the feature space is large, then it would imply large complexity in solving the above problem. On the contrary, by manipulating the Lagrangian we obtain the formula for the Dual:

Dual:

$$\begin{aligned} \underset{\alpha}{\text{minimize}} \quad & \alpha^T H\alpha - \alpha^T e \\ \text{subject to} \quad & \alpha^T \mathbf{t} = 0 \\ & 0 \leq \alpha \leq \frac{C}{M} \end{aligned}$$

Which is a minimization over the N-dimensional vector α. This means that now our problem only depends on the number of data points. Moreover in order to retrieve $\mathbf{w}$ we can calculate:

$$\mathbf{w} = \sum_{i=0}^{M} \alpha_i t_i x_i$$

But that still would very inefficient since we would have to store and maintain a D-dimensional vector. Instead we can use kernel functions to overcome this barrier. If we let K be the kernel associated with the RKHS we can say that:

$$<\mathbf{w}, x> = \sum_{i=0}^{M} \alpha_i t_i K(x_i, x)$$

To further illustrate why the Dual is often more efficient, consider the example of classification of emails. Then let $\mathbf{x}_i$ be the binary vector containing information on whether an email contains word i. Then if we attempt to incorporate bigrams and trigrams in in our vector $\mathbf{x}_i$ we see that the dimension D grows exponentially. Therefore, representing richer and richer concepts in our vector explodes the complexity in the Primal space.

As a result, by using $SVMs$ the telecommunications company would be able to incorporate more and more attributes, or even combinations of attributes without facing any optimization problems.

Problem 11: Classification

What are the limitations of the recommended model? How do you recommend that it be used?

Answer:
Even though $SVMs$ have many limitations, I think that the most significant one is that of overfitting. In general is it true that kernel models are more sensitive than other models in terms of overfitting. Therefore, the telecommunications company would have to be careful in their choice of parameters of the model.

A possible use of this prediction model would be to create specific favorable policies for customers who are about to churn. That way, the telecommunications company would be able to maintain more customers and as a result increase its profits.

Appendix

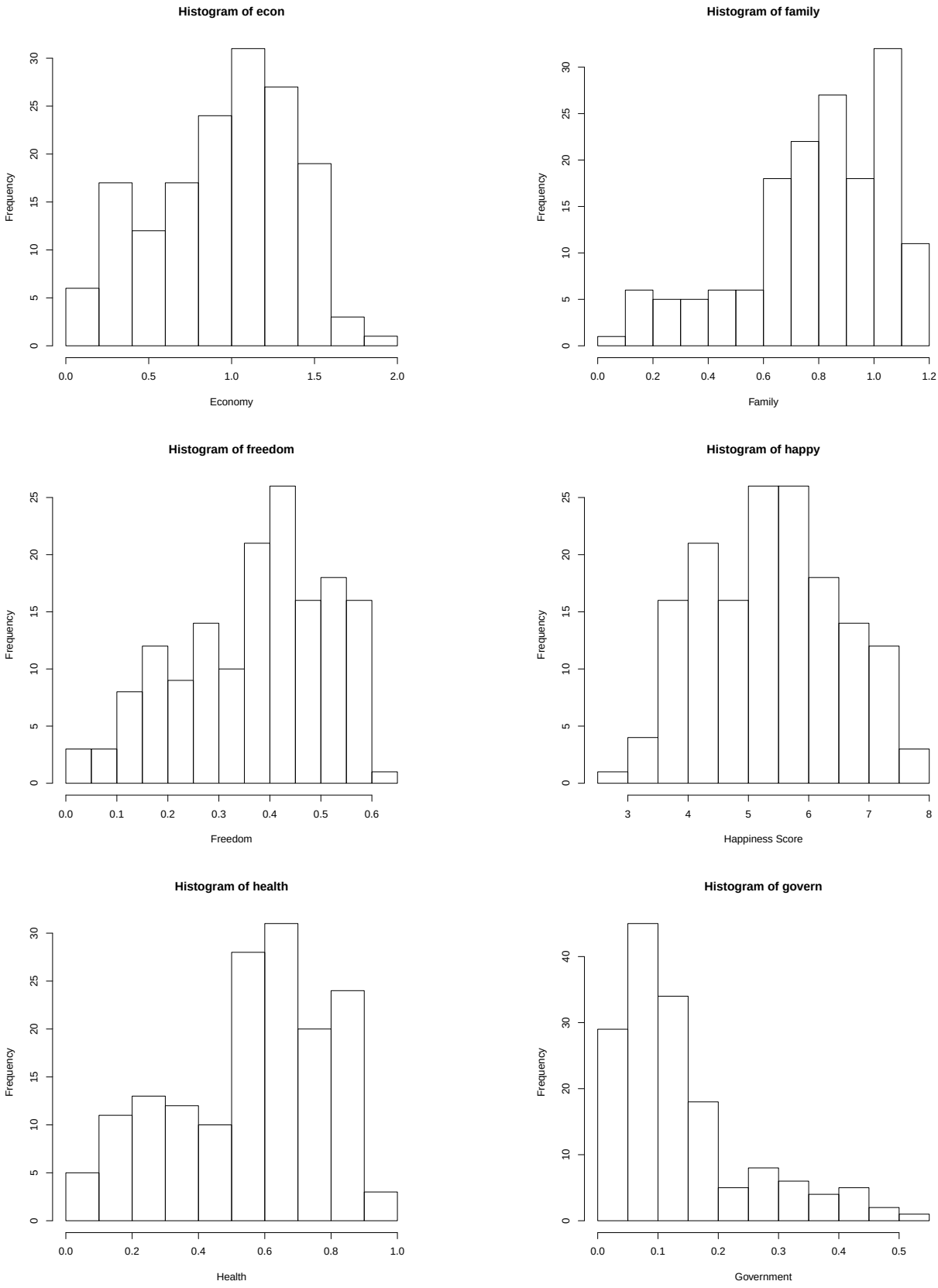
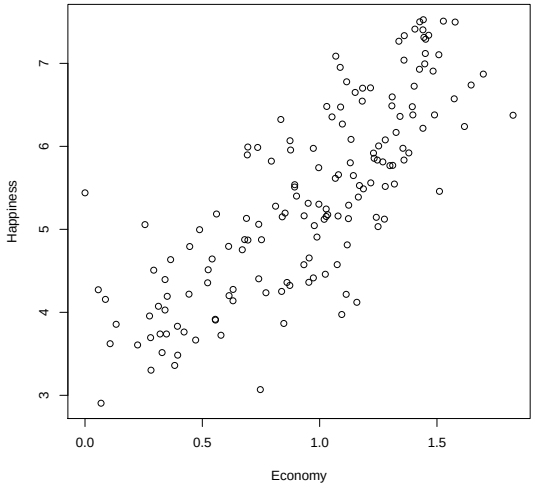
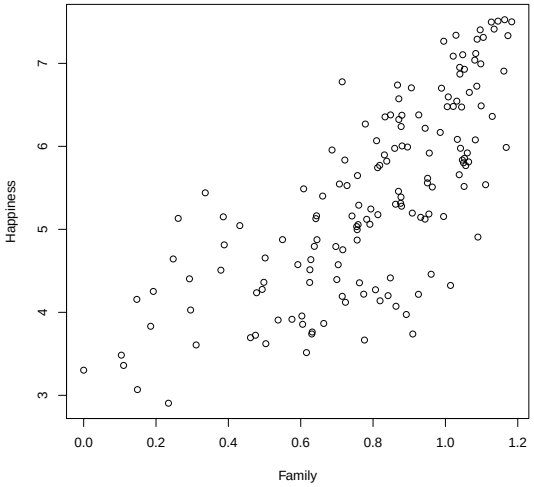
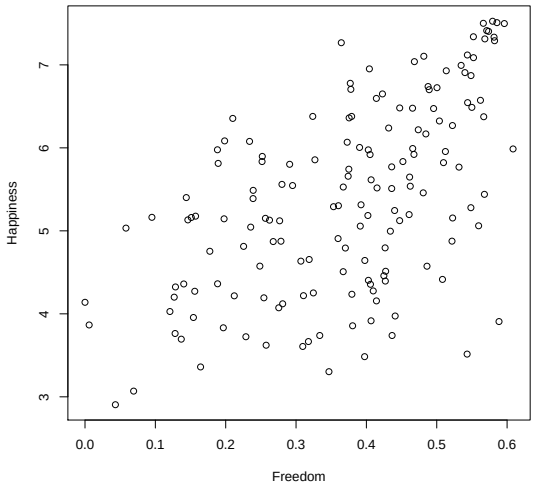
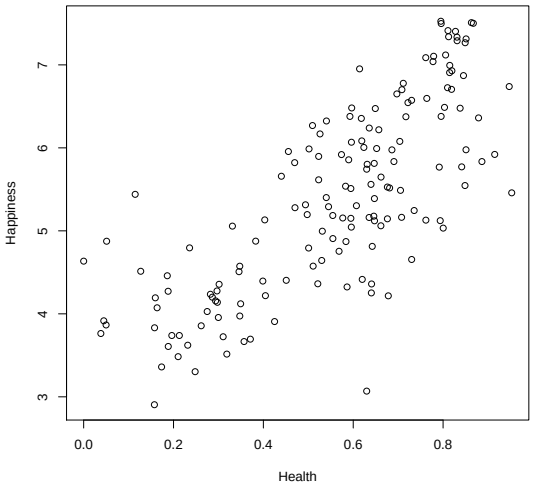


Figure 1: Histograms for all the given variables









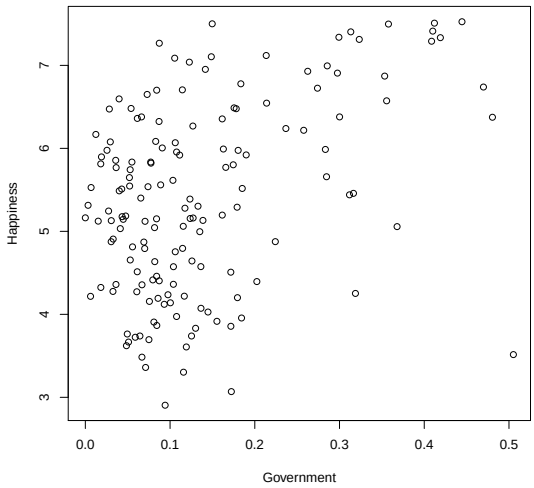


Figure 2: Scatter Plots