TIM 245 - Data Mining: Midterm

Due: *June 14, 2017*

Instructor: Tyler Munger

Panos Karagiannis
ID: -

Contents

Problem 1: Exploratory Data Analysis and Data Cleaning (50 points) **3**

Problem 2: Predicting Employee Satisfaction (20 points) **9**

Problem 3: Classifying Employee Turnover (20 points) **12**

Problem 4: Brainstorming (10 points) **14**

Problem 1: Exploratory Data Analysis and Data Cleaning (50 points)

The product management team has heard that data quality can be a significant issue when trying to create a good predictive model. Therefore, they have asked you to first assess the collected data and determine if it is suitable for creating an employee satisfaction prediction model.

1. Before you start, the product management team would like a written statement of your process, for Exploratory Data Analysis (EDA). (Hint: The process might include the steps such as compute descriptive statistics or determine threshold for outliers)

2. Then, they would like you to apply your EDA process to the collected data-set and format the results into a well-structured report.

3. Lastly, they would like you to provide them with a set of recommended data pre-processing steps (cleaning, transformation, etc.) for addressing any data quality issues that were discovered during the EDA process.

Answer:

1. Real world data are often **Incomplete**, **Noisy** as well as **Inconsistent**. Our Exploratory Data Analysis will consist of a combination of **visual** and **quantitative** tools to answer important questions for each selected attribute in our dataset. More precisely, for each attribute we will look into descriptive statistics such as the **typical value**, the **spread** for a typical value and whether it affects other attributes (**correlation**). Moreover, we will employ visual tools in order to estimate a good distributional fit for the data but also determine the existence of outliers (**histogram, box plots**). Also, we will use a combination of visual tools and descriptive statistics (**scatter plot, correlation**) to determine the co-linearity of attributes. Finally, we **omit** the attribute `id` since it simply functions as a unique identifier and hence no meaningful insights can be extracted.

Table 1: Pearson Correlation Matrix for all attributes

	id	current_sat	last_eval_sat	n_proj	m_hours	time	promotion	salary
id	1.000000	0.019087	0.004644	-0.003514	0.003326	0.004154	-0.000580	0.020131
current_sat	0.019087	1.000000	0.105021	-0.142970	-0.020048	-0.100866	0.025605	0.047476
last_eval_sat	0.004644	0.105021	1.000000	0.349333	0.339742	0.131591	-0.008684	-0.014793
n_proj	-0.003514	-0.142970	0.349333	1.000000	0.417211	0.196786	-0.006064	-0.004628
m_hours	0.003326	-0.020048	0.339742	0.417211	1.000000	0.127755	-0.003544	-0.004719
time	0.004154	-0.100866	0.131591	0.196786	0.127755	1.000000	0.067433	0.044233
promotion	-0.000580	0.025605	-0.008684	-0.006064	-0.003544	0.067433	1.000000	0.091139
salary	0.020131	0.047476	-0.014793	-0.004628	-0.004719	0.044233	0.091139	1.000000

2. Therefore for each attribute we discover the following information:
`current_satisfaction_score`:

   (a) Typical Value: Calculating the mean for this attribute we get that the average score is 61.28

   (b) Spread: The standard deviation (sample) for this data is 24.86. Since significant outliers exist in this attribute a better measure of dispersion is $mad = 19$

   (c) Distributional fit: Drawing a histogram of the data we see that the distribution looks slightly skewed to the left (i.e. few employees have low satisfaction score)

   (d) Correlation: Referring to $Table1$ we see that `current_satisfaction_score` is most strongly positively correlated with `last_evalualtion_satisfaction_score`, whereas, it is most strongly negatively correlated with `number_projects`.

   (e) Outliers: There exist some outliers which is evident from the magnitude of the measures of dispersion ($sd = 24.86$, $mad = 19$).

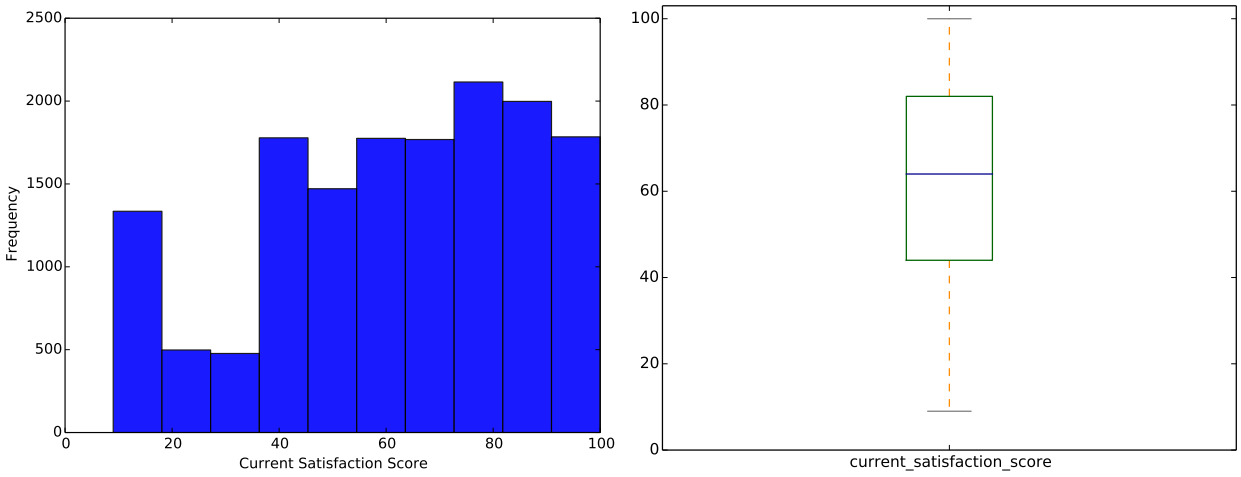

Figure 1: Histograms and Box Plot for `current_satisfaction_score`

`last_evalualtion_satisfaction_score`:

   (a) Typical Value: Calculating the mean for this data the average score is approximately 71.61

   (b) Spread: The standard deviation (sample) for this data is 17.11 and the $mad = 15$.

   (c) distributional fit: Drawing a histogram of the data we see that the distribution looks to be relatively symmetric and centered around the mean, even though a few outliers exist with very small `last_evalualtion_satisfaction_score`.

   (d) Correlation: Referring to $Table1$ we see that `last_evalualtion_satisfaction_score` is most strongly positively correlated with `current_satisfaction_score`, whereas, it is most strongly negatively correlated with `salary`.

(e) Outliers: Looking at the histogram as well as at the standard deviation there does not seem to be significant outliers for this attribute. Here we don't need to look at the *mad* measure since our data are relatively symmetric around the mean.

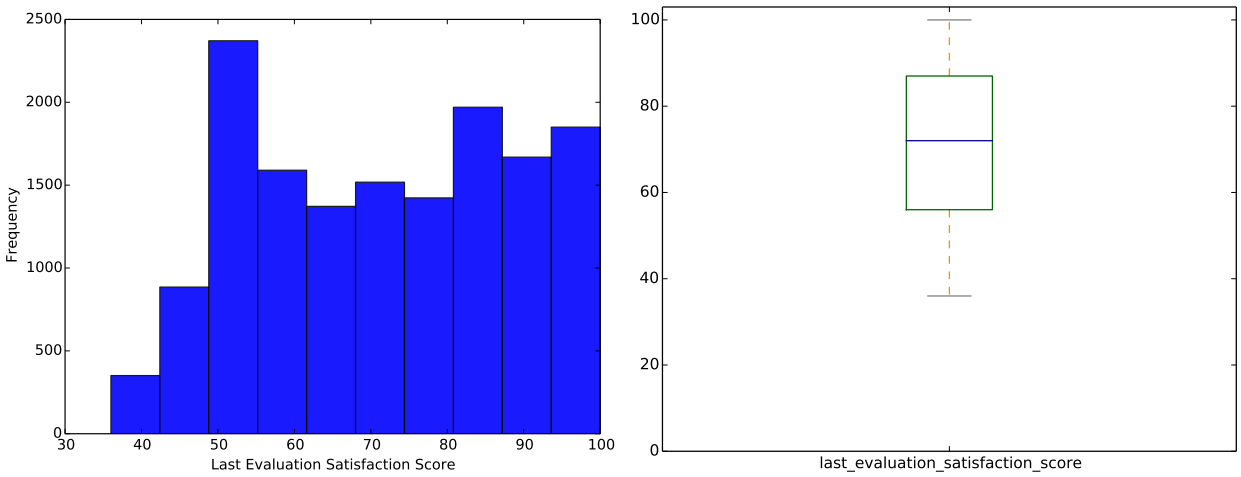

Figure 2: Histograms and Box Plot for `last_evalualtion_satisfaction_score`

`number_projects`:

(a) Typical Value: Calculating the mean for this data we get that the average budget is approximately 3.80

(b) Spread: The standard deviation (sample) for this data is 1.23. Since significant outliers exist in this attribute a better measure of dispersion is $mad = 1$.

(c) Distributional fit: Drawing a histogram of the data we see that the distribution looks to be symmetric around the mean with a few outliers

(d) Correlation: From *Table*1 we see that this attribute is most strongly positively correlated with `average_monthly_hours`, whereas, it is most strongly negatively correlated with `current_satisfaction_score`.

(e) Outliers: As before, looking at the histogram as well as at the standard deviation there does not seem to be significant outliers for this attribute. Here we don't need to look at the *mad* measure since our data are relatively symmetric around the mean.

`average_monthly_hours`:

(a) Typical Value: Calculating the mean for this data we get that the average budget is approximately 201.05

(b) Spread: The standard deviation (sample) for this data is 49.94. Since significant outliers exist in this attribute a better measure of dispersion is $mad = 44$.

(c) Distributional fit: Drawing a histogram of the data we see that the distribution looks to be symmetric around the mean with a few outliers

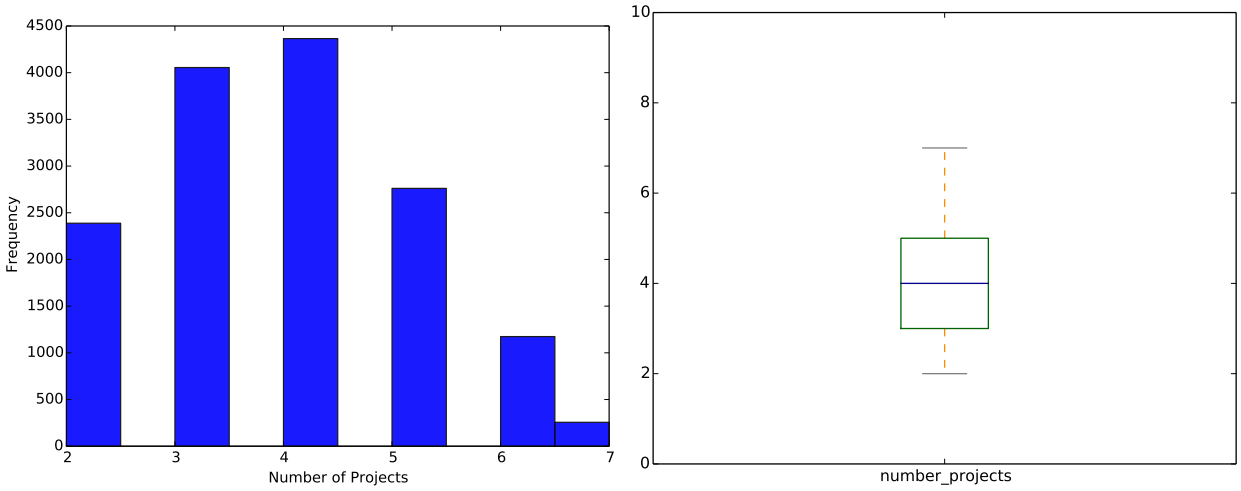

Figure 3: Histograms and Box Plot for `number_projects`

(d) CORRELATION: From *Table*1 we see that this attribute is most strongly positively correlated with `time_spent_at_company`, whereas, it is most strongly negatively correlated with `current_satisfaction_score`.

(e) OUTLIERS: As before, looking at the histogram as well as at the standard deviation there does not seem to be significant outliers for this attribute. Here we don't need to look at the *mad* measure since our data are relatively symmetric around the mean.

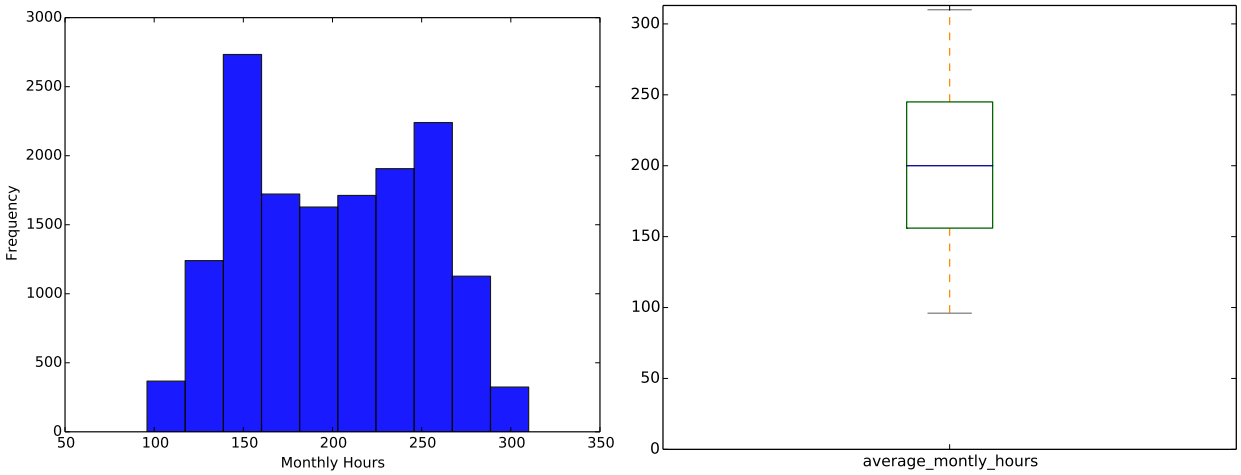

Figure 4: Histograms and Box Plot for `average_monthly_hours`

`time_spent_at_company`:

(a) TYPICAL VALUE: Calculating the mean for this data we get that the average time spent is approximately 3.49

(b) SPREAD: The standard deviation (sample) for this data is 1.46. Since outliers exist

in this attribute another measure of dispersion is $mad = 1$.

(c) DISTRIBUTIONAL FIT: Drawing a histogram of the data we see that the distribution is skewed to the right.

(d) CORRELATION: From *Table*1 we see that this attribute is most strongly positively correlated with `number_projects`, whereas, it is most strongly negatively correlated with `current_satisfaction_score`.

(e) OUTLIERS: Looking at the histogram, as well as, at the standard deviation there seem to be some outliers for this attribute. In particular there are a few datapoints with large `time_spent_at_company`

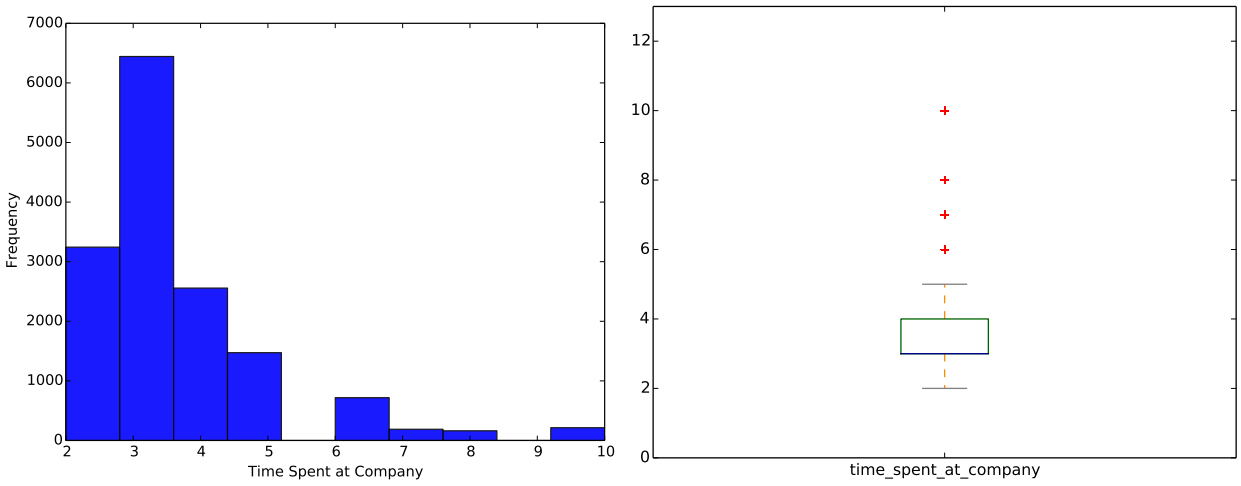

Figure 5: Histograms and Box Plot for `time_spent_at_company`

`promotion_in_last_5_years`:

(a) TYPICAL VALUE: Calculating the mean for this data we get that the average time spent is approximately 0.02

(b) SPREAD: The standard deviation (sample) for this data is 0.14.

(c) DISTRIBUTIONAL FIT: Drawing a histogram of the data we see the majority of the data is clustered in one bin.

(d) CORRELATION: From *Table*1 we see that this attribute is not strongly correlated with any other attributes.

(e) OUTLIERS: Looking at the histogram, as well as, at the standard deviation there seem to be very few datapoints that are not clustered in the bin $[0, 1]$. **Therefore a box plot is omitted**.

`salary`:

(a) TYPICAL VALUE: Calculating the mean for this data we get that the average time spent is approximately $99,214$

(b) SPREAD: The standard deviation (sample) for this data is $41,504$.

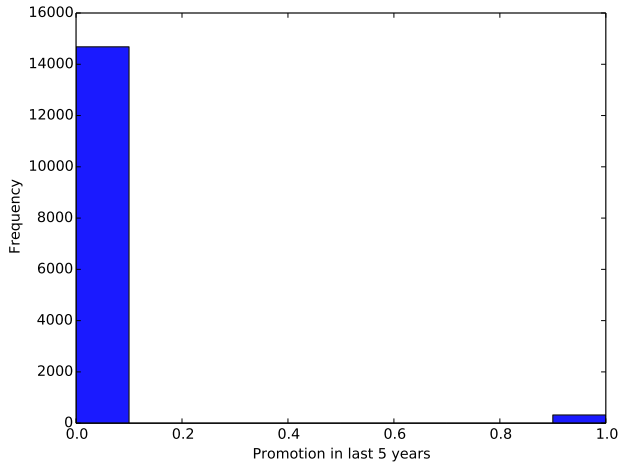

Figure 6: Histograms and Box Plot for `promotion_in_last_5_years`

(c) DISTRIBUTIONAL FIT: Drawing a histogram of the data we see that the distribution is skewed to the right.

(d) CORRELATION: From *Table*1 we see that this attribute is not mostly correlated with any other attributes.

(e) OUTLIERS: Looking at the histogram, as well as, at the standard deviation there seem to be significant outliers for this attribute.

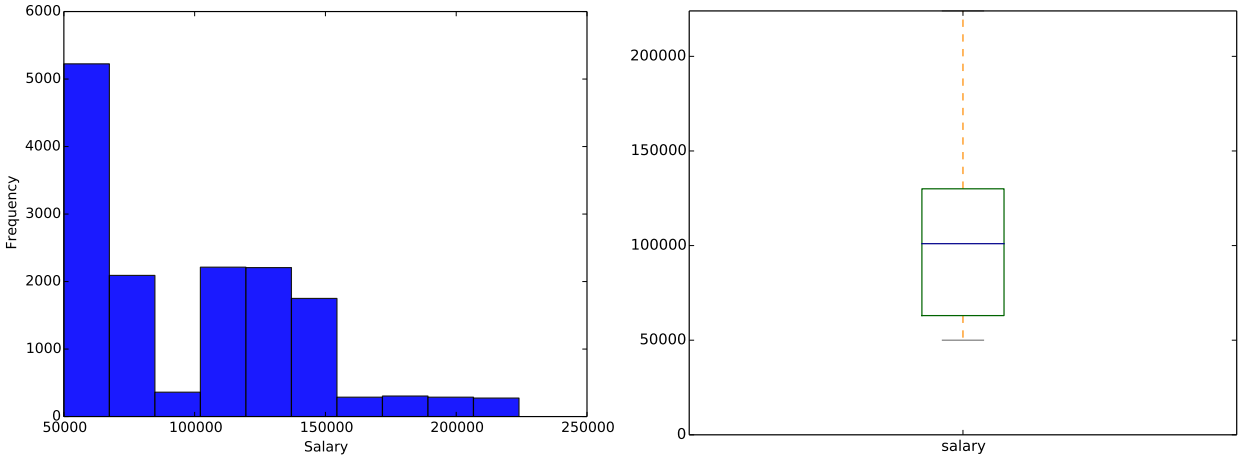

Figure 7: Histograms and Box Plot for `salary`

3. One of the most important pre-processing steps, would be to locate and remove **duplicate and missing** values. Moreover, we need to **remove outlier** values from our attributes, since these values have a large impact on the quality of our results. Lastly, we could also search for **inconsistencies** in the naming conventions of the **department** attribute.

Problem 2: Predicting Employee Satisfaction (20 points)

The product management team has specified the requirement that the model is able to predict the employee's satisfaction score within +/- 25 points of the employee's actual score. You have been asked to perform the following tasks related to constructing the prediction model.

1. First, apply your suggested data pre-processing from Problem 1 to the data-set and encode the nominal "department" attribute as a set of binary indicator attributes (dummy variables).
2. Build and evaluate the following prediction models using the cleaned data-set: linear regression, ridge regression, lasso regression, and elastic net.
3. Compare and contrast the performance of the different prediction models. Does your best model achieve the product managers' target accuracy? Do you think the model's performance is good? Explain why.
4. What is the interpretation of the selected model? What can we say about the relationship between the input attributes and the satisfaction score?
5. The product management team suggested including information about the manager of each employee into the model in order to improve accuracy. Do you think that this would help improve the model?s predictive accuracy? Explain why.

Answer:

1. First we need to explore whether our attributes contain duplicates. Assuming that each employee has a unique `id` then we can query for duplicate values of this attribute, to find that this dataset contains **no duplicate ids**.

   Next performing a brute force search over all the entries of our dataset we determine that there are **no missing values**.

   Based on our analysis in Question 1, we know that there exist **outliers** in our attributes. Therefore for all of our attributes we define $Q_0 = 5\%$ quantile and $Q_1 = 95\%$ quantile and then take the attributes which fall inside these quantiles. Following this process, we only keep the following values:

$$38 \leq \texttt{current\_satisfaction\_score} \leq 86$$
$$53 \leq \texttt{last\_evaluation\_satisfaction\_score} \leq 90$$
$$3 \leq \texttt{number\_projects} \leq 4$$
$$148 \leq \texttt{average\_montly\_hours} \leq 255$$
$$3 \leq \texttt{time\_spent\_at\_company} \leq 4$$
$$0 \leq \texttt{promotion\_in\_last\_5\_years} \leq 1$$
$$60,000 \leq \texttt{salary} \leq 137,000$$

   Finally, before converting our nominal value to numerical we **filter out inconsistencies**. Namely, we cluster:

- {r&d , R&D, rd, RandD} → rd
- {Human_Resources, human_resources, hr} → hr
- {product_mng, product_management} → product_mng
- {IT,it, information_technology } → IT
- {sales, Sales} → sales
- {support, Support} → support

Next we attempt to encode the nominal variable "department" as a set of binary indicators. By processing our csv file, we notice that there are **in total** 12 **distinct values** that this attribute takes. Therefore we encode these categories and hence we create 12 **new binary numerical attributes**.

2. After discarding outliers we still have not merged the attributes together in order to create a single dataframe. We notice, that since we have deleted many values, "merging" the columns back together would result in having many **empty** cells. Therefore, we decide to delete all datapoints that contain at least 1 empty cell. That reduces significantly our datasets's size from 14999 instances to 2200.

   The **reason behind** deleting all instances with at least one empty cell, is that we do not want any outliers in our dataset. More precisely, if an empty cell exists, then it means that its original value had been filtered out by the procedure we followed in the previous step (step 1), and hence, it was an outlier. An **alternative** approach would be to replace every missing value with the mean of that attribute, but that would inevitably distort the data and introduce some bias. This is not necessary here, since the resulting dataset size is adequate to allow our model to "learn" the weights and converge to a meaningful solution.

   We split the cleaned data set into *train* and *test* in the ratio of $3:1$ and then we compute various linear models. We also compute the maximum and minimum difference in order to determine if a model can predict the employees satisfaction score within 25 points. The results are summarized in the following table:

Table 2: MSE, Max/Min Difference for various linear models

	MSE	Maximum Difference	Minimum Difference
Least Squares	209.20	25.70	-43.27
Ridge	209.10	25.06	-48.67
Elastic Net	208.92	24.87	-48.56
Lasso	208.80	24.70	-48.47

3. We see that the simple linear *Least Squares* model has the largest MSE which is logical since it does not penalize for the magnitude of the coefficients, as a result it is much

more likely to overfit the data. *Ridge Regression* performs better than *Least Squares*, nevertheless, its MSE score is worse than both *Elastic Net and Lasso*. It is worth mentioning, that when we compute the coefficients for *Ridge Regression* we get that none of them is 0, thus verifying that **ridge regression does not perform variable selection.** On the contrary, *Lasso* is able to perform variable selection and therefore increase its accuracy achieving the smallest MSE error among all other methods. Also, *Elastic Net* is a hybrid method that uses both *Ridge and Lasso Regression* and manages to perform very similarly to *Lasso*. Therefore, if I were to suggest any model to the management of the company I would recommend **Lasso**, because it has the smallest MSE.

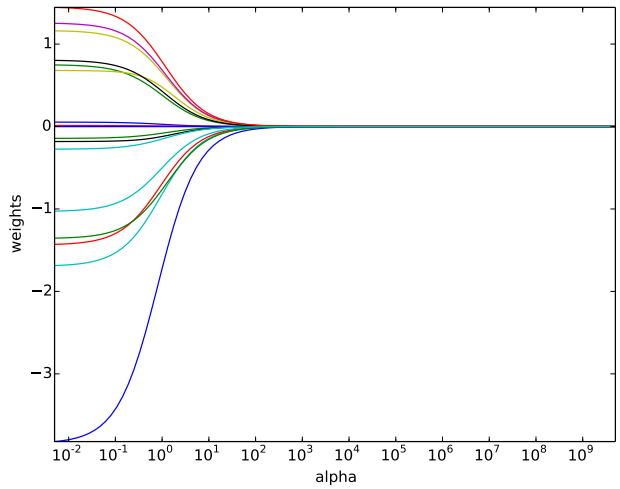

Figure 8: Weights approaching 0 as α increases in the example of Ridge Regression

Even if Lasso has the lowest MSE we see that it still **fails to meet the managers' expectations**. The reason for this performance, is that this model **does not** perform very well given that the MSE is so large. The MSE has the same "units" as the target variable and we know that `current_satisfaction_score` ranges from 22 to 91 (after cleanup), nevertheless, the MSE is $208 \gg 21,91$.

4. Using the *Lasso* model, we get that the $y-$intercept is approximately 7. This means, that the baseline `current_satisfaction_score` is 7.

   The coefficients indicate the strength and type of the relationship. Firstly, we see that not all coefficients 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 some of the attributes with the lowest coefficients are `salary and average_monthly_hours`

which is expected since as we discussed earlier these attributes have a very small correlation with `current_satisfaction_score`. Also, the attributes with relatively high coefficients were the departments that people are working which signifies that the attribute `department` plays a significant role in the taget variable.

5. Even if this information would be useful to have, it is true, that it would introduce sparsity in our data. Typically no more than $5-10$ employees share the same manager, therefore, including the manager of each employee would lead to a much larger number of variables used in the model. As a result our regression models would suffer from the curse of dimensionality and hence they would not perform well.

Problem 3: Classifying Employee Turnover (20 points)

Based on market research, the product management team has identified that Xenefit's customers also want to know which employees are at risk of leaving the company (employee turnover). To develop this feature, they have collected an additional attribute that indicates if the employee left the company or not. You have been asked to perform the following tasks related to creating the employee turnover classification model. What model do you recommend that the product management use to predict employee turnover? Explain why.

1. First, integrate the employee turnover information with the employee satisfaction data-set.
2. The product management team has asked you to explain the differences between Logistic Regression, Support Vector Machines, and K-Nearest Neighbors. In particular, how do these algorithms with respect to computational complexity, performance (underfitting vs overfitting), and interpretability?
3. Perform a set of experiments using the three learning algorithms. Which model performs the best? Explain why.
4. What model do you recommend that the product management use to predict employee turnover? Explain why.

Extra credit:
Create an additional model using a learning algorithm of your choice. Compare and contrast the performance of your model with the Logistic Regression, Support Vector Machine, and K-Nearest Neighbors models.

Answer:

1. We see that in the file `employee_turnover_dataset.csv` contains the attribute `id` which is a unique identifier. Since our original dataset also contains the employees' `id` we can use this attribute to merge the two csv files. It is worth mentioning that the merging of the two files happens **after we have filtered department spelling inconsistencies** from our original csv file.
2. In terms of **complexity**, *SVM and Logistic Regression* models, behave very similarly in this problem. These models minimize a convex loss functions and hence any local minimum is also guaranteed to be a global minimum as well. As far as KNN is concerned, it does not minimize a loss function, yet, the computation of the "distance" between

data points can be very expensive if many features are incorporated into the model.

It has to be mentioned, that all these models have similar complexities for relatively small number of attributes, as it is in the case of $Xenefits$. **Nevertheless**, if the management wanted to incorporate more attributes into the model then the complexities of *Logistic Regression and KNN* would explode, whereas, *SVM* can tackle this problem. In *SVM*, we can proceed by solving the *Dual problem* instead of the *Primal* and using a *Kernel function* to retrieve the weights (a detailed explanation of this fact was submitted in HW2 when comparing Logistic Regression and SVM).

As far as **performance** is concerned, KNN tend to overfit the data more easily than the other two methods. Therefore, in a KNN model we should carefully decide on the parameter K. Similarly in an *SVM* model as we increase the parameter C we decrease the tolerance for misclassification but if we increase too much there is the danger of overfitting the data ($C \gg 0$ is like a hard margin). Also, *Logistic Regression* models can overfit but we can control the size of the coefficient of the model by introducing regularization.

Finally, in terms of **interpretability** the simplest model is KNN since no function minimization is involved. *SVM and Logistic Regression* are more complex models which try to maximize the *margin* and model the class conditional probabilities, respectively. *Logistic Regression* is always non-linear, whereas, in *SVM* a non-linear Kernel function is frequently used. In the following parts we use the **Polynomial Kernel** for our *SVM* model.

3. The results for all three models were obtained through 10-fold cross validation and are summarized below (the attribute `id` is not considered in the model):

Table 3: Classification Models for **unnormalized** data **with** outliers

	Accuracy(%)	F-measure
SVM (C=1)	76.35	0.671
Logistic Regression	77.39	0.746
KNN (K=1)	96.71	0.967
ZeroR	76.19	0.659

In the case where we have not cleaned our dataset, we see that both *SVM and Logistic Regression* perform very poorly, since they outperform the naive *ZeroR* method by only 1%. Nevertheless, the KNN method performs extremely well when comparing it to the baseline *ZeroR* method. This is surprising since the data has not been normalized and hence, we would expect certain attributes that have large magnitude (i.e. `salary`), to dominate the "sum" in the KNN model.

Testing on our **clean** dataset we get the following results:

Table 4: Classification Models for **normalized** data **without** outliers

	Accuracy(%)	F-measure
SVM (C=1)	98.32	0.975
Logistic Regression	98.31	0.975
KNN (K=1)	98.55	0.986
ZeroR	98.31	0.975

Here we see that since the baseline *ZeroR* method performs so well, it is hard for the other models to significantly outperform it. Again, we observe that the KNN model has the best performance in terms of Accuracy and F-measure, even though, marginally. The reason why KNN performs so well must be partially attributed to the "geometry" of the problem. More precisely, it might be true that employees that are "close" (in the vector space) to each other, tend to make the same decision as to leave the company or not.

4. The model that I would propose to the company is KNN. The most important reason is because it seems to be performing very well in the context of this problem as we analyzed in the previous parts of this question. Moreover, since the company does not use many attributes to predict whether an employee will leave or stay in the company, the KNN model is efficient.

Problem 4: Brainstorming (10 points)

Having been impressed by your work on the employee satisfaction and turnover models, the product management team has asked you for new ideas about how data mining can be used to improve Xenefits product offerings in the future. Apply a structured brainstorming process to generate 3-5 possible ideas for improving human resource management using data mining.

For each generated idea, provide a brief description of:

1. What human resource management problem is being addressed, e.g. improve employee engagement.
2. What type of data mining task is involved: classification, prediction, cluster analysis, or association analysis.
3. What data-sets would need to be collected for the data mining task.
4. How could Xenefits use the resulting model (supervised learning) or patterns (unsupervised learning) in their product offering.

   Extra credit:
   Implement the improvements your suggested improvements above and re-evaluate the model. Do the improvements make a difference in the model?s prediction accuracy? Explain why or why not?

Answer:

1. Organizational Effectiveness: Since we would like to assign a group on each employee (and we do not know the true label for each data point), we could use clustering which is an **unsupervised** learning technique. In this case many algorithms could be used based on our domain knowledge. If clusters tend to be globular we could use a fast *K-Means* algorithm, whereas, if clusters tend to me less globular we could use *DB-Scan*

2. Recruitment and availability of skilled labor: Xenefits could apply data mining algorithms for predicting whether a university graduate would be a good fit for the company. Naturally, this falls into the binary classification problem category (supervised learning). The data that the company would have to gather for this task vary, nevertheless, some useful analytics could be the *GPA, interests, specific technical skills* of the prospective employee. By using data mining, the company can look through the data quickly and can also pinpoint the best candidates with a high degree of accuracy. Recruiting skilled employees would enable Xenefits to make better the quality of its products but also explore new ideas about potential products.

3. Leadership development: The company could attempt to predict the leadership development of any given employee from a scale of $1 - 100$. Xenefits could utilize this information in order to create more influential employees that could help advance further the company's products. In order extract such information, Xenefits could use prediction data mining tools such as regression. The data that would need to be available are common factors that lead to leadership enhancement such as: *ability to delegate, confidence, commitment and creativity.*

4. Compensation and benefits: Instead of using standard benefits and compensation plans for all employees, Xenefits could identify what would satisfy its employees the most. Instead of relying on salary surveys or intuition, and alternative approach would be to use data to identify which types and amounts of rewards/benefits have the highest measurable impact on employee productivity. The company could use unsupervised learning clustering methods in order discover groups of employees that seek the same kind of benefits and create specialized plans. In such a way, the company could improve the quality of its products. Finally, in order to perform such an analysis, Xenefits would have to gather the preferences of each employees in terms of desirable compensations and benefits.