Udacity

Intro to Machine Learning

Project

IDENTIFY FRAUD FROM ENRON EMAIL

---

by Omoju Miller

July 12, 2016

IDENTIFY FRAUD FROM ENRON DATA: INTRODUCTION

PROJECT OVERVIEW

In early 2000s, the Enron corporation of Houston, Texas was considered one of the most profitable energy companies in the United States. Fortune magazine named Enron one of the most innovative companies in the US for six years in a row, 1996 - 2001. However, by late 2001, the Enron corporation filed for bankruptcy after it was revealed that the company overstated its profits and defrauded it share holders of a considerable amount of wealth. Several person were later indicted and convicted of fraud in the Enron case.

The goal of this project is to use applied machine learning, based on released Enron data—Enron emails and financial accounts—to identify persons of interest in the Enron debacle.

PROBLEM STATEMENT

The problem that we are interested in solving is building an algorithm that can help us identify persons who might be of interest with regards to fraud at Enron. These persons are predominantly Enron employees and consultants who worked for the corporation.

The dataset that we have available for this task was collected over two weeks in 2002 by the Federal Energy Regulatory Commission (FERC) during its investigation into the case. The dataset contains over 600,000 emails of 158 high-level Enron employees as well as the financial records of stock payments, salary and so forth of those employees.

To identify persons of interest that might have been party to fraud at Enron we suggest the following strategy:

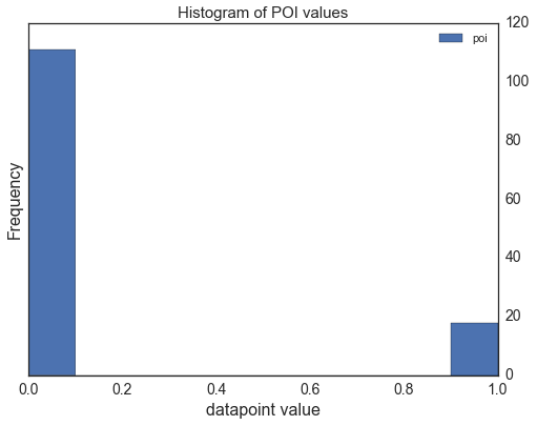
(a) Explore the dataset to ensure its integrity and understand the context.

(b) Identify features that may be used. If possible, engineer features that might provide greater discrimination.

(c) With the understanding that this a "classification" task, explore a couple of classifiers that might be well suited for the problem at hand.

(d) Once classifiers have been identified, tune them for optimality.

The Enron dataset while it contains a robust amount of emails, contains data for only about 150 people. An interesting aspect of the Enron data is that the class labels for our classification is heavily unbalanced at a ratio of around 6:1 in favor of negative examples as can be seen in figure 0.1a.



(a)

Figure 0.1: **Person of Interest plot.** *By visualizing this feature which acts as the label for our classifiers, we can see that the dataset is extremely skewed in favor of non-persons of interest.*

For this reason, using "Accuracy" as a performance metric leads to misleading information. A more apt measure of the performance of a learner should take into account the results of a confusion matrix and calculate "precision" and "recall," noting that precision is more a measure of a classifiers exactness.

DATA EXPLORATION

The Enron dataset used in the project was created by the Udacity team. It was done by combining the Enron email and financial data. The data is stored in a python dictionary where each key-value pair in the dictionary corresponds to one person. For example shows the corresponding key-value data point of Enron executive Pai Lou, the only executive to reap and keep substantial wealth from Enron. Table 0.1 shows a list of persons of interest that were generated by the Udacity team.

```
data_dict['PAI LOU L']
{'bonus': 1000000,
 'deferral_payments': 'NaN',
 'deferred_income': 'NaN',
 'director_fees': 'NaN',
 'email_address': 'lou.pai@Enron.com',
 'exercised_stock_options': 15364167,
 'expenses': 32047,
 'fraction_from_poi': 0,
 'fraction_to_poi': 0,
 'from_messages': 'NaN',
 'from_poi_to_this_person': 'NaN',
 'from_this_person_to_poi': 'NaN',
 'loan_advances': 'NaN',
 'long_term_incentive': 'NaN',
 'other': 1829457,
 'poi': False,
 'restricted_stock': 8453763,
 'restricted_stock_deferred': 'NaN',
 'salary': 261879,
 'shared_receipt_with_poi': 'NaN',
 'to_messages': 'NaN',
 'total_payments': 3123383,
 'total_stock_value': 23817930}
```

- financial features:
  `salary`, `deferral_payments`, `total_payments`, `loan_advances`, `bonus`, `restricted_stock_deferred`, `deferred_income`, `total_stock_value`, `expenses`, `exercised_stock_options`, `other`, `long_term_incentive`, `restricted_stock`, `director_fees`

- email features:
  `to_messages, email_address, from_poi_to_this_person, from_messages, from_this_person_to_poi, shared_receipt_with_poi`
- POI label:
  `poi`

PERSONS OF INTEREST
BELDEN TIMOTHY N
BOWEN JR RAYMOND M
CALGER CHRISTOPHER F
CAUSEY RICHARD A
COLWELL WESLEY
DELAINEY DAVID W
FASTOW ANDREW S
GLISAN JR BEN F
HANNON KEVIN P
HIRKO JOSEPH
KOENIG MARK E
KOPPER MICHAEL J
LAY KENNETH L
RICE KENNETH D
RIEKER PAULA H
SHELBY REX
SKILLING JEFFREY K
YEAGER F SCOTT

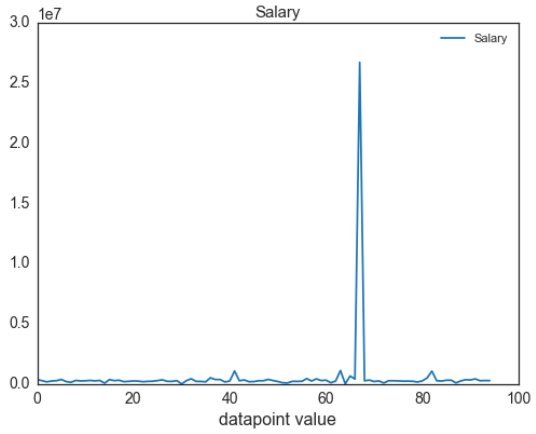
Table 0.1: Persons of interest in the dataset

While the original Enron dataset contained a robust amount of emails, this pared down one contains data for only about 150 people. Furthermore, the dataset is extremely skewed as can be seen from figure 0.1a.

OUTLIER DETECTION

As part of the data exploration process, we were careful to analyze the data for potential outliers. One of the first tasked we did was visualize the salary of Enron executives 0.2a. From that visualization, it was clear that there was an outlier in the dataset. We found a data-point that was completely outside of a reasonable range as can be seen from figures 0.2a and 0.3a.

A closer look revealed that the data-point was an input error which included the TOTAL of all salaries as its own row. The data-point was removed leaving the dataset in a more realistic state as can be seen from figure 0.3b.



(a)

Figure 0.2: **Plot of Enron employee salaries.** *From this visualization, we can see there is a huge spike to the right of data-point 60, this spike corresponds with an outlier of value $26,704,229.00.*

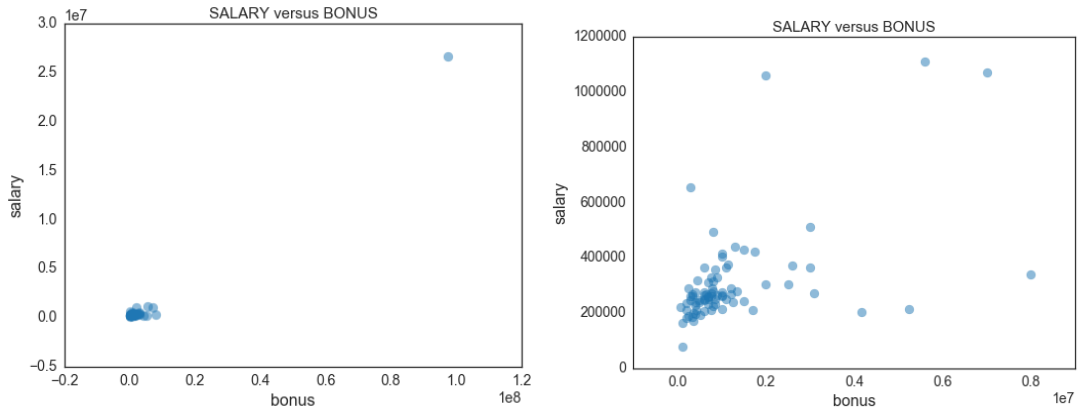
EXPLORATORY VISUALIZATION

One of the areas we concentrated on was that of compensation. We were very interested in the compensation packages as represented in the financial data of the executives. We moved ahead with a working hypothesis, that if fraud was indeed occurring at Enron, then more than likely, the money was probably going to be funneled out through paid bonuses and stocks. A more rigorous hypothesis would then correlate stock options granted and exercised by the Enron executives with the sales of the shares on the open market. However, such an investigation, is outside the scope of this project.

To zoom in on the compensation, we focused on three features 'bonus,', 'exercised_stock_options' and 'restricted_stock'. Table 0.2 shows some financial data for some of the highest compensated employees. It comes as no surprise that we see that the sitting CEO at the time of the collapsce, Ken Lay, had the second highest bonus paid, that wasn't

NAME	SALARY	BONUS
ALLEN PHILLIP K	$201,955.00	4,175,000
BELDEN TIMOTHY N	$213,999.00	5,249,999
SKILLING JEFFREY K	$1,111,258.00	5,600,000
LAY KENNETH L	$1,072,321.00	7,000,000
LAVORATO JOHN J	$339,288.00	8,000,000
NAME	SALARY	EXERCISED STOCK OPTIONS
FREVERT MARK A	$1,060,932.00	10,433,518
PAI LOU L	$261,879.00	15,364,167
SKILLING JEFFREY K	$1,111,258.00	19,250,000
RICE KENNETH D	$420,636.00	19,794,175
LAY KENNETH L	$1,072,321.00	34,348,384
NAME	SALARY	RESTRICTED STOCK
YEAGER F SCOTT	$158,403.00	3,576,206
IZZO LAWRENCE L	$85,274.00	3,654,808
BAXTER JOHN C	$267,102.00	3,942,714
KEAN STEVEN J	$404,338.00	4,131,594
FREVERT MARK A	$1,060,932.00	4,188,667
SKILLING JEFFREY K	$1,111,258.00	6,843,672
PAI LOU L	$261,879.00	8,453,763
WHITE JR THOMAS E	$317,543.00	13,847,074
LAY KENNETH L	$1,072,321.00	14,761,694

Table 0.2: Financial data on some of Enron's top paid employees.



(a) (b)

Figure 0.3: **TOTAL insertion.** *Figure (a) Bonus versus salary with outlier present. Figure (b) Same dataset but without the outlier. By visualizing these two features in a scatter plot we were able to clearly detect the existence of an outlier. In addition, once the data has been cleaned of outliers, we can see that majority of data-points cluster in a close range, with a few data-points spread outside bonus range of $2 million. From descriptive statistics, we learned that the 75% of bonus amount was $1 million.*

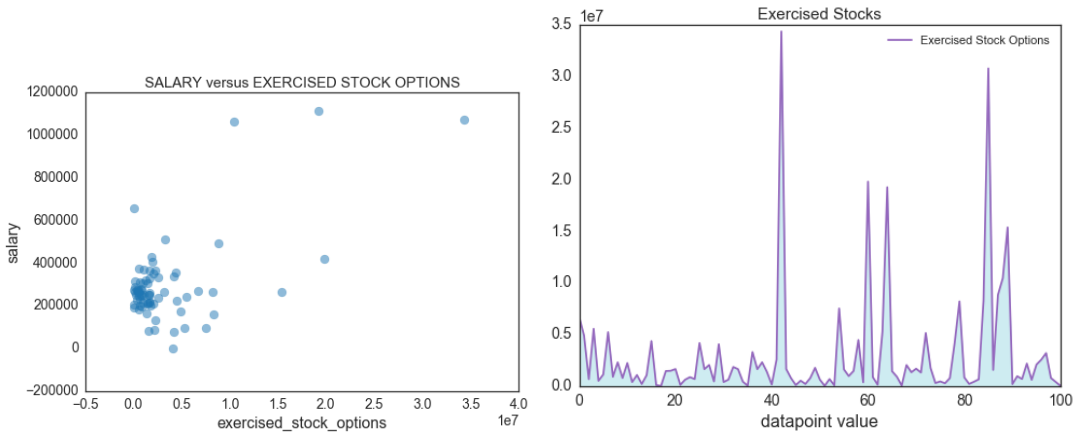
surprising. What was surprising was the name "John Lavorato." He was the employee that got the highest bonus. So who was John Lavorato? He was the former head of Enron's trading operations.

Another interesting character that showed up in the financial data was "Pai Lou." From table 0.2, one can see that he had some of the largest exercised stock options. Apart from the past CEOs and chairmen, he was *the* employee that sold the most Enron stock. He also got some of the largest shares of restricted stock. Its interesting that these men aren't on the POI list in table 0.1. I firmly believe they are probably on the co-conspirators unindicted list.

To gain insights into the relationship between salary and exercised stock options, we took that subset of data and ran a KMeans clustering algorithm on it to see how the data clustered; we focused on three clustered as can be seen from figure 0.4a. The clusters generated matched our intuition about the executives, mostly all the five executives with the highest exercised stock options were clustered as one, as can be seen in the visualization in figure 0.4a by the markers colored blue. It is important to note we were not able to infer a relationship between salary and exercised stock options as we had thought.

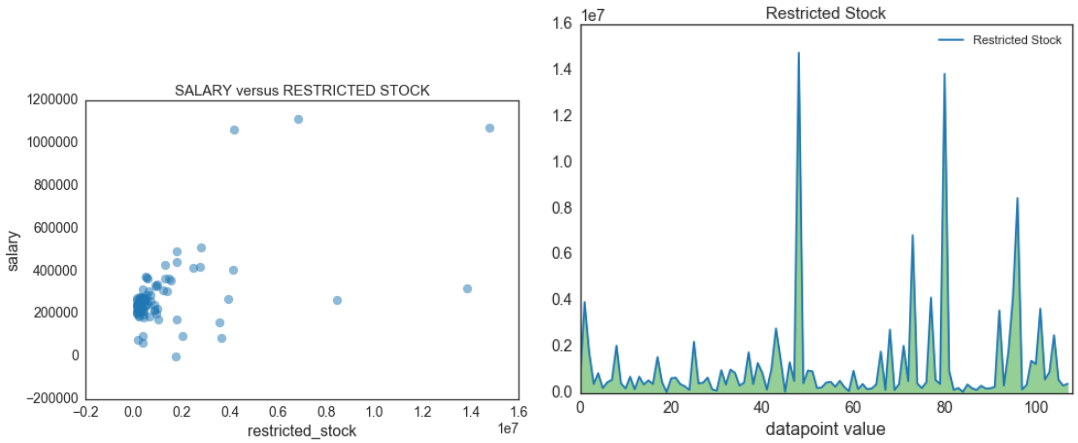
We proceeded on, with a hypothesis that more than likely, the executives with the highest salaries, probably also had the highest restricted stock. Just as with exercised stock options, it wasn't the case. From figure 0.5a one can see that there is a data-point with a salary close to $200,000, but with over 8 million shares, when most of the

other folks around that salary bracket had less than a million shares. Its important to note that for both our visualizations, the KMeans clustering algorithm, clustered the data-points along both the exercised stock options and restricted stocks.



(a) (b)

Figure 0.4: **A closer look at salary and exercised stock options.** *Figure (a) A scatter plot of salary versus exercised stock options. Figure (b) A plot of exercised stock options. From these two plots one can see that there is a small subset of senior executives who exercised a significant amount of Enron shares.*



(a) (b)

Figure 0.5: **A closer look at salary and restricted stock.** *A scatter plot of salary versus restricted stock. Figure (b) A plot of restricted stock.*

ALGORITHMS AND TECHNIQUES

For the problem of identifying fraud from the Enron financial and email data, we experimented with four different classifiers, three ensemble methods and on regression:

(a) A RandomForestClassifier
We selected this learner because it is considered one of the best off-the-shelf learning algorithm, and requires almost no tuning.

(b) An ExtraTreesClassifier
Its another ensemble learner like Random Forest, with one caveat. When creating a branch in a tree, Random Forest chooses the most discriminant value, whereas ExtrRandTrees split point is arbitrarily. This helps to increase the bias slightly and lower the variance even more.

(c) A Gradient Boosted Trees (XGBoost) Classifier
From reading a lot of posts on Kaggle and perusing the past winners of KDD competitions all touting the robustness of this classifier, we wanted to see how it would perform against an unbalanced dataset like the one we have for this problem.

(d) A LogisticRegression (Robust, with Lp = L1) Classifier
Its computational simpler than the ensemble methods. Furthermore, it has been used in real life to predict adverse risk events that have relatively small chances of occurring like credit card fraud and so on. The dataset composition of those events are similar to the Enron dataset, where the labels are heavily skewed towards one class.

We selected the robust LogisticRegression based on the fact that it handles outliers better. While the data has been cleansed of "outliers," there are a few executives whose data-points fall outside of the mean by several standard deviations.

BENCHMARK

For this specific problem, we were unable to acquire benchmarks that we could test the performance of our learners against. However in the notes accompanying the project description, we were tasked to shoot for a precision score of over 0.3. From our four learners, and the initial trials, we have already superseded this number, as can be seen from table 0.3.

IDENTIFY FRAUD FROM ENRON DATA: METHODOLOGY

DATA PREPROCESSING

Significant data preprocessing wasn't necessary for this problem. Once we ridded our dataset of outliers, we decided to play with a few features to get a feel of how they would behave with the algorithms. We tested out our four learners with `salary` and `bonus` features before we scaled them with a minimax-scaler afterwards. We found no difference, so we decided to forgo scaling.

FEATURE ENGINEERING

We suspected that there was interdependence of features so we engineered a feature, the "fraction of emails to and from poi." We figured this metric could give us another window into discriminating the data.

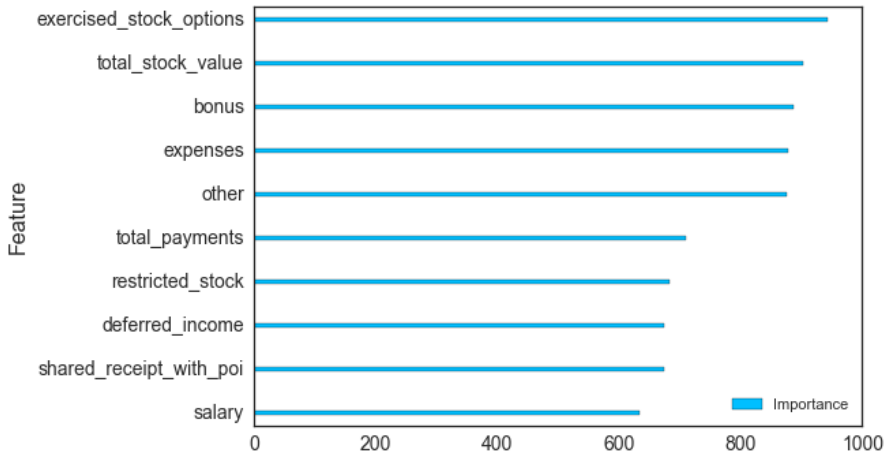
FEATURE SELECTION

Our approach to feature selection was based on intuition as well as running the features through feature selection algorithms to guide us in our decision of features to use. In the case of the latter we selected a RandomForest classifier as well as a Gradient Boosted Trees—XGBoost—classifier to implement feature selection.

For the RandomForest algorithm, we used a baseline classifier, we fitted it to our data and used its "`feature_importances_`" to obtain a score on the strength of that feature. To obtain reliable scores, we ran the algorithm a 1000x and averaged the scores of the features to obtain the top n features as visualized in figure 0.6a.

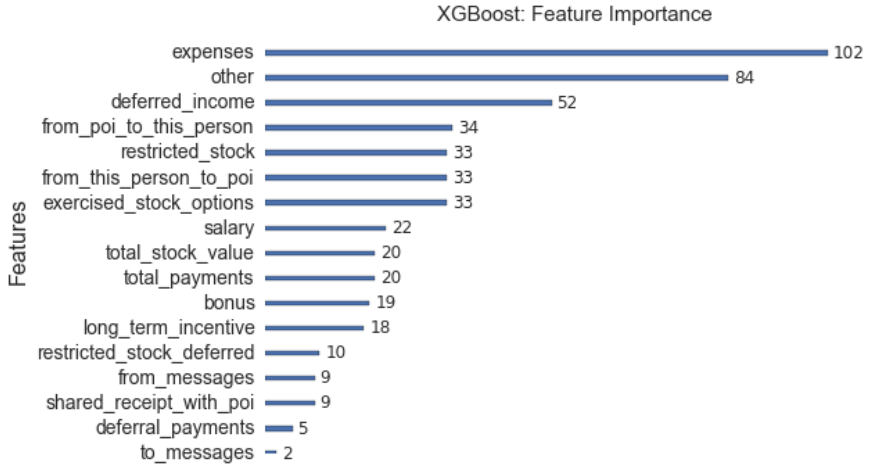
For the XGBoost classifier, we took a different approach to obtaining reliable data. We re-sampled the dataset and grew it a 1000x using sklearn's `StratifiedShuffleSplit` function with a thousand folds. We then fed this more expansive dataset to an `XGBoost` classifier and applied its `plot_importance` function to discover the important features as can be seen from figure 0.7a.

We used these features selected by the algorithms along side our domain guided features on our four baseline learners, the results can be see in table 0.3. From this experiment, we decided to go with our ad-hoc generated features because the performance was superior when we considered the precision score of all four learners. It is important to note that the other features were not woeful in their



(a)

Figure 0.6: **Feature importance as discovered by RandomForest.**



(a)

Figure 0.7: **Feature importance as discovered by XGboost.**

Table 0.3: Scores

Result of Training with **RandomForest** generated features			
	Metrics		
	Precision	Recall	F1
ExtraTreesClassifier	0.387	0.144	0.210
LogisticRegression	0.416	0.170	0.241
RandomForestClassifier	0.370	0.140	0.203
XGBClassifier	0.441	0.200	0.275
Result of training with **XGBoost** generated features			
	Metrics		
	Precision	Recall	F1
ExtraTreesClassifier	0.454	0.174	0.252
LogisticRegression	0.293	0.200	0.237
RandomForestClassifier	0.379	0.131	0.194
XGBClassifier	0.416	0.196	0.266
Result of training with **Ad-hoc** features			
	Metrics		
	Precision	Recall	F1
ExtraTreesClassifier	0.449	0.140	0.213
LogisticRegression	0.499	0.169	0.252
RandomForestClassifier	0.404	0.146	0.215
XGBClassifier	0.422	0.299	0.350

performance, and overall, these features with baseline classifiers had already put us over the threshold of the benchmark precision score—0.3—we were to surpass.

For the final algorithm, we ended up going with our intuition and selecting features that we had explored in the data analysis. These features are as follows:

```
poi
bonus
exercised_stock_options
restricted_stock
fraction_from_poi
fraction_to_poi
from_poi_to_this_person
from_this_person_to_poi
salary
```

IMPLEMENTATION

We implemented the four learning algorithms. For each of the learners we implemented the baseline algorithm using 10 fold cross validation to get an accuracy score. These scores proved to be misleading. We went ahead and used a stratified shuffle split cross validation with a thousand folds and calculated the precision, recall and F_1 scores respectively.

From our trials we generated the following scores as can be seen in table 0.3. From these trials, we decided to go with the *Logistic Regression* because it gave us the highest precision score when training on our ad-hoc features. Furthermore, the regression also had the least computational time of 0.938s, whereas the Random Forest and the ExtraTrees both computed in 18.280s, and the XGBoost computed in around 10.925s.

REFINEMENT & MODEL EVALUATION AND VALIDATION

To improve on the performance of the algorithm, we decided to reduce the number of features that were used for training. Through some trial and error, we realized if we held the feature list to the first three features, our performance improve dramatically as can be seen in table 0.4.

```
bonus
exercised_stock_options
restricted_stock
```

Once we achieved this result, we moved forward with tuning the hyper-parameters of the robust logistic regression. We tuned the C

Table 0.4: Result of training with reduced features

	Metrics		
	Precision	Recall	F1
ExtraTreesClassifier	0.490	0.181	0.264
LogisticRegression	0.630	0.202	0.306
RandomForestClassifier	0.483	0.190	0.272
XGBClassifier	0.505	0.332	0.400
Results of tuning			
LogisticRegression	0.73010	0.18800	0.29901

constant as well as the tolerance through scklearn's (GridSearchCV) using a stratified shuffle split with a 1000 folds. The algorithm selected the best parameters as follows: C=0.125, tol=0.0001. We fitted this new classifier and achieved a precision score of 0.73.

IDENTIFY FRAUD FROM ENRON DATA: CONCLUSION

FREE-FORM VISUALIZATION

In this section, you will need to provide some form of visualization that emphasizes an important quality about the project. It is much more free-form, but should reasonably support a significant result or characteristic about the problem that you want to discuss. Questions to ask yourself when writing this section:

- Have you visualized a relevant or important quality about the problem, dataset, input data, or results?
- Is the visualization thoroughly analyzed and discussed?
- If a plot is provided, are the axes, title, and datum clearly defined?

REFLECTION

In this section, you will summarize the entire end-to-end problem solution and discuss one or two particular aspects of the project you found interesting or difficult. You are expected to reflect on the project as a whole to show that you have a firm understanding of the entire process employed in your work. Questions to ask yourself when writing this section:

- Have you thoroughly summarized the entire process you used for this project?
- Were there any interesting aspects of the project?
- Were there any difficult aspects of the project?
- Does the final model and solution fit your expectations for the problem, and should it be used in a general setting to solve these types of problems?

IMPROVEMENT

In this section, you will need to provide discussion as to how one aspect of the implementation you designed could be improved. As an example, consider ways your implementation can be made more general, and what would need to be modified. You do not need to make this improvement, but the potential solutions resulting from

these changes are considered and compared/contrasted to your current solution. Questions to ask yourself when writing this section:

- Are there further improvements that could be made on the algorithms or techniques you used in this project?
- Were there algorithms or techniques you researched that you did not know how to implement, but would consider using if you knew how?
- If you used your final solution as the new benchmark, do you think an even better solution exists?