Capstone Report

Udacity Machine Learning Engineer Nanodegree

DonorsChoose.org Application Screening

Aaron Penne

2018-06-25

Table of Contents

[1 Definition 3](#_Toc517894562)

[1.1 Project Overview 3](#_Toc517894563)

[1.2 Problem Statement 3](#_Toc517894564)

[1.3 Metrics 4](#_Toc517894565)

[2 Analysis 4](#_Toc517894566)

[2.1 Data Overview 4](#_Toc517894567)

[2.2 Data Exploration 4](#_Toc517894568)

[2.2.1 “Resources” Dataset 4](#_Toc517894569)

[2.2.2 “Training” and “Test” Datasets 5](#_Toc517894570)

[2.3 Exploratory Visualization 9](#_Toc517894571)

[2.4 Algorithms and Techniques 10](#_Toc517894572)

[2.5 Benchmark 10](#_Toc517894573)

[3 Methodology 11](#_Toc517894574)

[3.1 Data Preprocessing 11](#_Toc517894575)

[3.1.1 Basic Data Reading and Cleaning 11](#_Toc517894576)

[3.1.2 Feature Engineering 11](#_Toc517894577)

[3.1.3 Handling Text 11](#_Toc517894578)

[3.2 Implementation 12](#_Toc517894579)

[3.2.1 Model Selection and Data Cross Validation 12](#_Toc517894580)

[3.3 Refinement 13](#_Toc517894581)

[4 Results 13](#_Toc517894582)

[4.1 Model Evaluation and Validation 13](#_Toc517894583)

[4.2 Justification 13](#_Toc517894584)

[5 Conclusion 14](#_Toc517894585)

[5.1 Free-Form Visualization 14](#_Toc517894586)

[5.2 Reflection 14](#_Toc517894587)

[5.3 Improvement 14](#_Toc517894588)

[6 References 15](#_Toc517894589)

# Definition

(approx. 1-2 pages)

## Project Overview

Teachers in low income schools often find the school is unable to provide their students with adequate resources. This leads some teachers to purchase items with their own money to meet the students’ needs. To help solve this issue, Charles Best created the website DonorsChoose.org to allow philanthropic individuals the opportunity to selectively donate to various teacher projects. Teachers submit their applications to DonorsChoose.org, and if they are accepted then the project goes public and people may donate to it.

This charity has served as a Kickstarter for small public-school needs and provides a direct way to make an impact on young people’s lives. DonorsChoose.org has received top industry awards every year since 2005 and has fulfilled over 600,000 classroom projects. This impact can be made even greater by leveraging advanced technologies to actively utilize the datasets on hand.

DonorsChoose.org implements a screening process to ensure a high quality level of projects are listed on their site. This means hundreds of applications need to be read in detail by volunteers. This takes a lot of time and resources for the organization, and also adds delay to the teachers waiting for approval. The problem to be solved is a classification one, should the DonorsChoose.org accept or reject a given proposal. By automating a large segment of this process, applications that have an obvious classification can be accepted/rejected, while the more nuanced applications can be read more in depth by a volunteer.

## Problem Statement

The dataset is compiled from historical project applications that were submitted to DonorsChoose.org. These are presented on Kaggle (<https://www.kaggle.com/c/donorschoose-application-screening>) in the form of comma separated variable files (CSV). These will be used as inputs to create a predictive classifier. In order to solve this problem, several steps must be taken: The data must be imported, cleaned, examined, and joined to other inputs. Various algorithms must be selected and cross validated to identify which model fits best, perhaps combining models into a customized ensemble. The chosen algorithm(s) must be selected and trained on the historical data. Finally, testing data will be run through the model(s) to create predictions which will be scored by Kaggle.

There are three data files available: training application data, test application data, and data on the resources requested. The application data contains some metadata about the application and the essays. The resources data includes one line item per resource requested, and may have multiple line items for a single application. Initially it seems that there will be predictive information in the resources categorical data and numerical data. It also is possible that the categorical and numerical data in the application metadata will be valuable for prediction. The essays and other text information will likely contain the most information gain, as there will be a deep and wide breadth of words used by the nationwide applicants.

These initial observations lead me to believe a two pronged approach will be needed from a high level perspective. One will be model(s) based on the categorical and numerical data. The other will be model(s) based on the text essays.

## Metrics

Cross validation in section XYZ will be performed <cross validation diagram?>

The confusion matrix shown in the visualization section is used to calculate precision and recall. The confusion matrix shows the number of correct and incorrect predictions with the following numbers: True positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). These numbers are eponymous counts of, for example, the number times a prediction was positive when the actual label was positive. These are valuable metrics, but it is difficult to identify superior models by comparing 4 values for each model. TP, TN, FP, and FN are typically combined into two metrics called precision and recall, defined as

While these are useful in their own right, two metrics are still difficult to use when comparing multiple models. Therefore, they are combined into a single metric known as the F1 score, defined as the harmonic mean of precision and recall

Another way to use the TP and FP values is to use a receiver operating characteristic (ROC) curve. This metric originated in radar detection to score a radar operator’s ability to correctly identify a target. Nowadays it is used to characterize algorithmic classification instead of human radar operators, but the principle is the same. ROC curves plot the FP rate (FPR) against the TP rate (TPR) for all decision thresholds.

AUC ROC https://developers.google.com/machine-learning/crash-course/glossary#AUC

Accuracy

In this section, you will need to clearly define the metrics or calculations you will use to measure performance of a model or result in your project. These calculations and metrics should be justified based on the characteristics of the problem and problem domain. Questions to ask yourself when writing this section:

* Are the metrics you’ve chosen to measure the performance of your models clearly discussed and defined?
* Have you provided reasonable justification for the metrics chosen based on the problem and solution?

# Analysis

(approx. 2-4 pages)

## Data Overview

Each request contains metadata such as the school’s state, class grade, teacher ID, categories, submission datetime, etc. The primary content of each request is essays filled out by the teacher, with 2-4 essays per request. Another resources dataset is provided that details the description, quantity, and price of each item in the applications. This resources dataset is tied back to the applications dataset using a unique id for each application.

The training set has 182k records and the testing set has 78k records. There are two class labels in this dataset, “accepted” and “rejected”. An interesting break in the data is that proposals before a certain date had 4 questions to be answered. The later proposals had 2 questions. It will take some experimentation to determine how to properly handle this discrepancy, more information on this is in Section 7

Many of the attributes are numerical in nature. These will be the simplest to model. Many attributes are also categorical, meaning they can be converted to numerical values and modelled in similar ways. The bulk of the potentially interesting data, however, is the essays written by the teachers. These provide both a challenge and an opportunity, as there is quite a bit of information encoded in these texts.

## Data Exploration

### “Resources” Dataset

Each proposal has a list of items requested. Each proposal may request multiple item types, and multiples of any given item. The quantity requested and price of each item is given. We will want to map the requests to a single proposal, so some summarizing will be necessary. There are many features possible here, it makes sense to anchor the desired features to answerable questions.

* Do cheap or expensive proposals tend to get approved?
  + Highest/Lowest price of items (max/min of price)
  + Total cost of request (sum of price \* quantity for each proposal)
* Do proposals with few or many items tend to get approved?
  + Total number of items requested (sum of quantity)
  + Total number of unique items requested (count of items per proposal)
* Does the type of item requested affect approval?
  + Aggregation of all words from item descriptions (string join of description)

The possibility of information being gleaned from the list of descriptions means that text analysis will need to be used. This was obvious from the data descriptions but here it becomes apparent.

### “Training” and “Test” Datasets

The training and test datasets have the same features except the training data has class labels identified.

Notes on first pass of features

* id - unique id of the project application
  + Used to tie resources.csv data in
* teacher\_id - id of the teacher submitting the application
  + Counts of submissions per teacher captured in other feature, teacher\_number\_of\_previously\_posted\_projects
  + Will be disregarded
* teacher\_prefix - title of the teacher's name (Ms., Mr., etc.)
  + Categorical data with few unique values, likely useful
  + Needs to be encoded
* school\_state - US state of the teacher's school
  + Categorical data with ~50 unique values, likely useful
  + Needs to be encoded
* project\_submitted\_datetime - application submission timestamp
  + Needs to be split into multiple columns
    - Year
    - Month
    - Day of year
    - Day of week
    - Hour of day
* project\_grade\_category - school grade levels (PreK-2, 3-5, 6-8, and 9-12)
  + Categorical data with few unique values, likely useful
  + Needs to be encoded
* project\_subject\_categories - category of the project (e.g., "Music & The Arts")
  + Categorical data with few unique values, likely useful
  + Several records have 2 entries, needs to be split into 2 new columns
  + Needs to be encoded
* project\_subject\_subcategories - sub-category of the project (e.g., "Visual Arts")
  + Categorical data with few unique values, likely useful
  + Several records have 2 entries, needs to be split into 2 new columns
  + Needs to be encoded
* project\_title - title of the project
  + Needs to be normalized
* project\_essay\_1 - project\_essay\_4
  + Essays will need to be combined, see section XYZ #FIXME
* project\_resource\_summary - summary of the resources needed for the project
  + Needs to be augmented with resources.csv data
* teacher\_number\_of\_previously\_posted\_projects - number of previously posted applications by the submitting teacher
  + Useful as is
* project\_is\_approved - whether DonorsChoose proposal was accepted (0="rejected", 1="accepted"); train.csv only
  + Class labels

Note: Prior to May 17, 2016, the prompts for the essays were as follows:

* project\_essay\_1: "Introduce us to your classroom"
* project\_essay\_2: "Tell us more about your students"
* project\_essay\_3: "Describe how your students will use the materials you're requesting"
* project\_essay\_4: "Close by sharing why your project will make a difference"

The target variable is contained in project\_is\_approved, with binary classes: rejected=0 and approved=1.

Looking at the training data it becomes apparent that two approaches should be used to solve this classification problem:

1. **Classification based on categorical features**
2. **Classification based on text information**

The questions and corresponding features that come to mind are:

* **Do the categorical features independently have an affect on approval?**
  + ['teacher\_prefix', 'school\_state', 'project\_grade\_category', 'project\_subject\_categories', 'project\_subject\_subcategories']
  + Acceptance rate of each feature
  + Number of submissions of each feature
* **Does the date of submission affect approval?**
  + Approval rates by year, month, day of year, day of week, hour of day
* **Do new or experienced submitters get approved more?**
  + Acceptance rate of teacher\_number\_of\_previously\_posted\_projects
* **Does the title affect approval?**
  + Number of words in project\_title
  + Sentiment of project\_title
* **The big one: Do the essays affect approval?**
  + Length of each essay
    - Number of characters
    - Number of words
    - Word density
  + After cleaning essay text\* evaluate the following:
    - SVM linear classification
    - Topic models of each essay
    - doc2vec models of each essay
* **Does the resource summary affect approval?**
  + Closely linked with aggregated description field from resources table

\*Cleaning text covered in section XYZ #FIXME

Essays are an interesting chunk of this dataset and potentially have a wealth of information in them. Every application has questions 1 and 2, but the counts of questions 3 and 4 are significantly lower. Typically if there are large amounts of missing data in an attribute, it would be thrown out. However, I believe that removing project\_essay\_3 and project\_essay\_4 from consideration would throw away useful data. According to the data description:

Note: Prior to May 17, 2016, the prompts for the essays were as follows:

* project\_essay\_1: "Introduce us to your classroom"
* project\_essay\_2: "Tell us more about your students"
* project\_essay\_3: "Describe how your students will use the materials you're requesting"
* project\_essay\_4: "Close by sharing why your project will make a difference"

Starting on May 17, 2016, the number of essays was reduced from 4 to 2, and the prompts for the first 2 essays were changed to the following:

* project\_essay\_1: "Describe your students: What makes your students special? Specific details about their background, your neighborhood, and your school are all helpful."
* project\_essay\_2: "About your project: How will these materials make a difference in your students' learning and improve their school lives?"

To me, the newer question 1 appears to be a combination of the old questions 1 and 2, while the newer question 2 appears to be a combination of the old questions 3 and 4. The old questions 1 and 2 ask about the classroom and students, while the new question 2 asks about the students and the school. The old questions 3 and 4 ask about how materials will be used and how they will make a difference, while the new question 2 addresses the same topics.

As such I will combine the old question 1 and 2 into question 1, and the old question 3 and 4 into question 2.

Example from dataset:

Metadata:

|  |  |  |
| --- | --- | --- |
| **Feature Name** | **Value** | **Type** |
| id | p226941 | String (key) |
| teacher\_id | 103cc1667cf9361bf1c58c8425e76e95 | String (key) |
| teacher\_prefix | Mrs. | String (Categorical) |
| school\_state | CA | String (Categorical) |
| project\_submitted\_datetime | 9/5/16 19:28 | Datetime |
| project\_grade\_category | Grades PreK-2 | String (Categorical) |
| project\_subject\_categories | Literacy & Language, Math & Science | String (Categorical) |
| project\_subject\_subcategories | Literacy, Mathematics | String (Categorical) |
| project\_title | Technology Boost! | String |
| project\_essay\_1 | <See Essay 1 below> | String |
| project\_essay\_2 | <See Essay 2 below> | String |
| project\_essay\_3 | Null | String |
| project\_essay\_4 | Null | String |
| project\_resource\_summary | My students need a projector and tablets to allow them to access the same technology most others take for granted. | String |
| teacher\_number\_of\_previously\_posted\_projects | 1 | Integer |
| project\_is\_approved | 1 | Integer |

Essay 1:

My children come to school everyday with the same expectations as any other children; they are eager to learn, excited about new discoveries, and want to feel like they have a place in this world. My challenge is to meet their expectations regardless of the fact that many of them face extremely difficult economic situations at home. \r\n\r\nAll children deserve access to educational tools, regardless of their family's economic standing! Most of my students come from working class families that are trying their best, but are unable to provide what many take for granted. The only exposure most of my students have to technology is at school, and our aging classroom equipment is barely limping along.

Essay 2:

With our new iPads, my students will be able to access educational programs, games and websites to enhance their learning, especially in math and literacy. My children love sharing their work with each other as well as seeing real-life examples of classroom concepts, and our new LCD projector will give us this opportunity all day long!\r\nThis new equipment will give my wonderful students the technology boost they deserve and need to be on an equal playing field with other children their age. It will also help them develop the skills necessary in third grade when they are expected to be computer literate to complete the state's on-line standardized tests!

The full essays from this record are included above for reference. Note that these are stored as a single string, with escaped characters such as “\n” present. This means some string cleaning will be needed when reading this data in. More on the approach to processing this data can be seen in section XYZ.

The project\_subject\_categories and project\_subject\_subcategories features are categorical but many records contain multiple categories. For example, a project\_subject\_categories value may be a single entry such as “Applied Learning”, or it may have two entries such as “Music & The Arts, Health & Sports”. It will likely be a good idea to split these cases into multiple features. Using a string split it is possible to verify that the maximum number of categories in either feature is two. This anomaly initially points to splitting these features into 4 features. It is also intuitive that the subjects will give a good result if aggregated into a single feature.

In this section, you will be expected to analyze the data you are using for the problem. This data can either be in the form of a dataset (or datasets), input data (or input files), or even an environment. The type of data should be thoroughly described and, if possible, have basic statistics and information presented (such as discussion of input features or defining characteristics about the input or environment). Any abnormalities or interesting qualities about the data that may need to be addressed have been identified (such as features that need to be transformed or the possibility of outliers). Questions to ask yourself when writing this section:

* If a dataset is present for this problem, have you thoroughly discussed certain features about the dataset? Has a data sample been provided to the reader?
* If a dataset is present for this problem, are statistics about the dataset calculated and reported? Have any relevant results from this calculation been discussed?
* If a dataset is ***not*** present for this problem, has discussion been made about the input space or input data for your problem?
* Are there any abnormalities or characteristics about the input space or dataset that need to be addressed? (categorical variables, missing values, outliers, etc.)

## Exploratory Visualization

Correlation matrix

Choropleth of states? Bar chart probably. For rate of acceptance of state and sex.

Population comparison of state? # of applications from states to refine rate and get statistical significance

Scatterplot pairplot

Violin plot of each feature?

Histogram of each feature for sure

In this section, you will need to provide some form of visualization that summarizes or extracts a relevant characteristic or feature about the data. The visualization should adequately support the data being used. Discuss why this visualization was chosen and how it is relevant. Questions to ask yourself when writing this section:

* Have you visualized a relevant characteristic or feature about the dataset or input data?
* Is the visualization thoroughly analyzed and discussed?
* If a plot is provided, are the axes, title, and datum clearly defined?

## Algorithms and Techniques

Before beginning investigations, the data will be pre-processed. The resource data will need to be mapped to the proposal data. An interesting problem to deal with in this dataset is that before a certain date the proposals included 4 questions. After that date the newer proposals included 2 questions. This means that something needs to be done with the NaN values present in the newer data. There are a few options, and each will be tested to see if a positive or negative impact on the score occurred. The common thing to do when an attribute has a large percentage of missing data is to simply delete that attribute. This seems to be an erroneous approach in this case, as much data will be lost. Reading the proposal prompt questions shows that the old prompts #1 and #2 are very similar to the new prompt #1. Also, the old prompts for #3 and #4 are very similar to the new prompt #2. This leads to another option of melting the old #1 and #2 questions into #1, and old #3 and #4 into #2. Alternatively, all of the text can be combined into a single text field then processed.

Initial investigations will include the rate of acceptance of different attributes (% accepted per state for example) and correlation of various attributes. This stage will also involve visualizing the data to determine which variables provide information and which ones only contribute noise.

The first pass of this project will be an attempt to classify the proposals based solely on metadata and resources requested. The essays will be ignored. This will allow for a quick prototype of a classifier without diving into the complexities of text. The second pass will only use the essays as the primary inputs. NLP techniques such as topic modelling will be used to compare the essays of accepted and rejected proposals. Finally, an ensemble method combining these models will be used to make the final prediction.

In this section, you will need to discuss the algorithms and techniques you intend to use for solving the problem. You should justify the use of each one based on the characteristics of the problem and the problem domain. Questions to ask yourself when writing this section:

* Are the algorithms you will use, including any default variables/parameters in the project clearly defined?
* Are the techniques to be used thoroughly discussed and justified?
* Is it made clear how the input data or datasets will be handled by the algorithms and techniques chosen?

## Benchmark

Random selection will be the first model to compare against, weighted with the same class distribution as the training set. The training set is imbalanced and has a significantly larger number of accepted proposals, so that percentage will be used for the random selection. This model is to be a naïve baseline to get a sense of where zero performance lies.

From the training data it can be shown that the class balance is 84% approved and 16% denied. This means the baseline random model needs to be programmed to randomly approve applications 84% of the time.

AUC score

F1 score results

In this section, you will need to provide a clearly defined benchmark result or threshold for comparing across performances obtained by your solution. The reasoning behind the benchmark (in the case where it is not an established result) should be discussed. Questions to ask yourself when writing this section:

* Has some result or value been provided that acts as a benchmark for measuring performance?
* Is it clear how this result or value was obtained (whether by data or by hypothesis)?

# Methodology

(approx. 3-5 pages)

## Data Preprocessing

### Basic Data Reading and Cleaning

### Feature Engineering

### Handling Text

The resources dataset had some text data which may be useful. It contained one description for each record which was akin to a description that would exist on a purchase order, such as this example:

Dixon Ticonderoga Wood-Cased #2 HB Pencils, Box of 96, Yellow (13872)

For applications that had multiple resources requested, the descriptions were aggregated into a single record, separated by a comma. The new feature collecting all descriptions is title res\_descriptions. This would allow the possibility of predicting on this row although not much information was expected to be gained from this feature.

The applications dataset contained six text features that had potential for natural language processing (NLP). They are project\_title, project\_essay\_1-4, project\_resource\_summary, and res\_descriptions.

According to the explanation in section XYZ, records with four essays were aggregated into two. The first and second essays were combined into essay\_1 while the third and fourth were combined into essay\_2. If the record had 2 essays they were mapped directly, the first to essay\_1 and the second to essay\_2. The essays are also aggregated into a single feature, essay\_agg, along with the project\_title to make a single blob of the entire submitted essay text. It is possible for this aggregated essay to yield good separation between records and possibly have high correlation with the class labels.

After the pivoted resources dataset is joined to the applications dataset it is possible to separate out the text data and perform NLP actions on it. First all NaN or null values in the text data are filled with ‘’, meaning an empty string. Then all escaped characters such as “\n” or “\r” are removed. Next all surrounding whitespace is stripped from each value of each feature. The project\_subject\_categories and project\_subject\_subcategories features are split into 4 new features: subject\_a, subject\_b, subject\_c, and subject\_d. Perhaps these features will yield more information if concatenated into a single feature, so they are joined in a new feature subject\_agg.

Word count

Subjectivity, polarity, and sentiment are computed on the non-normalized text. These produce numerical results which have meaningful ranges. The results are joined to the numerical data. The TextBlob library is able to perform these calculations directly using pretrained vocabularies and the VADER method.

Text normalization is a critical component of the NLP approach. The normalization section takes the longest time to run, even longer than training models. For each text value, the following steps are taken. All characters are converted to lowercase. All stopwords (according to the corpus XYZ) are removed, as well as any punctuation. Words are stemmed, split into single word members of a list, then joined with a single space between all words. Stemming is a process where words are reduced to their base form, allowing for more universal matching and less unique variants of a given word. For example, the words “reading”, “reader”, “readers”, and “read” may all be reduced to the root “read”. Some stemmed words are not actual words themselves (such as “forgiv”) but that is irrelevant as far as models are concerned. These normalized values are created in new features and given the suffix “\_norm” to preserve the original text features in addition to the new normalized features. This allows flexibility when evaluating models.

In this section, all of your preprocessing steps will need to be clearly documented, if any were necessary. From the previous section, any of the abnormalities or characteristics that you identified about the dataset will be addressed and corrected here. Questions to ask yourself when writing this section:

* If the algorithms chosen require preprocessing steps like feature selection or feature transformations, have they been properly documented?
* Based on the ***Data Exploration*** section, if there were abnormalities or characteristics that needed to be addressed, have they been properly corrected?
* If no preprocessing is needed, has it been made clear why?

## Implementation

### Model Selection and Data Cross Validation

Kfold cross val

Techniques

Looped through multiple classifiers

Looped through multiple parameters/settings/hyperparameters?

Some scores

In this section, the process for which metrics, algorithms, and techniques that you implemented for the given data will need to be clearly documented. It should be abundantly clear how the implementation was carried out, and discussion should be made regarding any complications that occurred during this process. Questions to ask yourself when writing this section:

* Is it made clear how the algorithms and techniques were implemented with the given datasets or input data?
* Were there any complications with the original metrics or techniques that required changing prior to acquiring a solution?
* Was there any part of the coding process (e.g., writing complicated functions) that should be documented?

## Refinement

In this section, you will need to discuss the process of improvement you made upon the algorithms and techniques you used in your implementation. For example, adjusting parameters for certain models to acquire improved solutions would fall under the refinement category. Your initial and final solutions should be reported, as well as any significant intermediate results as necessary. Questions to ask yourself when writing this section:

* Has an initial solution been found and clearly reported?
* Is the process of improvement clearly documented, such as what techniques were used?
* Are intermediate and final solutions clearly reported as the process is improved?

# Results

(approx. 2-3 pages)

## Model Evaluation and Validation

In this section, the final model and any supporting qualities should be evaluated in detail. It should be clear how the final model was derived and why this model was chosen. In addition, some type of analysis should be used to validate the robustness of this model and its solution, such as manipulating the input data or environment to see how the model’s solution is affected (this is called sensitivity analysis). Questions to ask yourself when writing this section:

* Is the final model reasonable and aligning with solution expectations? Are the final parameters of the model appropriate?
* Has the final model been tested with various inputs to evaluate whether the model generalizes well to unseen data?
* Is the model robust enough for the problem? Do small perturbations (changes) in training data or the input space greatly affect the results?
* Can results found from the model be trusted?

## Justification

In this section, your model’s final solution and its results should be compared to the benchmark you established earlier in the project using some type of statistical analysis. You should also justify whether these results and the solution are significant enough to have solved the problem posed in the project. Questions to ask yourself when writing this section:

* Are the final results found stronger than the benchmark result reported earlier?
* Have you thoroughly analyzed and discussed the final solution?
* Is the final solution significant enough to have solved the problem?

# Conclusion

(approx. 1-2 pages)

## Free-Form Visualization

Dataisbeautiful level crapola

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?

**Before submitting, ask yourself. . .**

* Does the project report you’ve written follow a well-organized structure similar to that of the project template?
* Is each section (particularly **Analysis** and **Methodology**) written in a clear, concise and specific fashion? Are there any ambiguous terms or phrases that need clarification?
* Would the intended audience of your project be able to understand your analysis, methods, and results?
* Have you properly proof-read your project report to assure there are minimal grammatical and spelling mistakes?
* Are all the resources used for this project correctly cited and referenced?
* Is the code that implements your solution easily readable and properly commented?
* Does the code execute without error and produce results similar to those reported?

# References

M. Fatourechi, R. K. Ward, S. G. Mason, J. Huggins, A. Schlögl and G. E. Birch, "Comparison of Evaluation Metrics in Classification Applications with Imbalanced Datasets," *2008 Seventh International Conference on Machine Learning and Applications*, San Diego, CA, 2008, pp. 777-782.

Zweig MH, Campbell G (1993) Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine. Clinical Chemistry 39:561-577. [PubMed](http://www.ncbi.nlm.nih.gov/pubmed/8472349)