**Capstone Proposal**

Krittisak Chaiyakul
Nov 7th, 2017

**Proposal**

1. **Domain Background**

Federal Aviation Administration is the United States Department of Transportation. They provide information about the transportation, including flights. Their website shows the real-time flight delay information, which consists of flight delay time. However, the information covers only major airports in the United States.

In this project, we will cover all airports across the United States. We can apply machine learning to learn on previous the flight delay information and predict the expected delay time of the flights. It will be a lot useful if we the likely arrival time of our flights.

2. **Problem Statement**

In this project, we will be using k-nearest neighbor regression to predict the flight delay time by training with the previous flight delay information. We measure the performance of the model by predicting the delay time in minutes on testing set and calculating the error rate. Therefore, our goal is to find the optimal model to minimize the error rate.

3. **Datasets and Inputs**

The dataset 'flight.csv' is derived from https://www.kaggle.com/usdot/flight-delays. Row represents each flight information and column represents each feature, one of which we need to predict, 'ARRIVAL_DELAY.' Instead of using every features, we will use only 12 features, which are guaranteed.

- 'MONTH', 'DAY', 'DAY_OF_WEEK', 'FLIGHT_NUMBER', 'TAIL_NUMBER', 'SCHEDULED_DEPARTURE', 'SCHEDULED_TIME', 'DISTANCE', and 'SCHEDULED_ARRIVAL' - Numeric
- 'AIRLINE', 'ORIGIN_AIRPORT', and 'DESTINATION_AIRPORT' - String

However, we will discard a few flights if they are canceled. This dataset is related to the problem because the features consist of the characteristic of the flights and are necessary for predicting the delay time.

## 4. Solution Statement

Firstly, we will split the dataset into training and testing set. Then we train our model on training set to get the best k value for lowest error rate. Finally, we test the final model with testing set and calculate the error rate of the model.

## 5. Benchmark Model

Our benchmark model will be the average delay time of all flights in the same 'MONTH', 'DAY', and 'DAY_OF_WEEK' columns. We try to get the better predicted results with our proposed model.

## 6. Evaluation Metrics

We use mean absolute deviation (MAD) as our metric.

$$mad = \frac{1}{N}\sum_{i=1}^{N}|out_i - pred_i|$$

*where* $N$ *is the total number of predicting flights,*
$out_i$ *is the previous recorded flight delay*
$pred_i$ *is the predicted delay time of flight* $i$ *from our model*

We calculate the mad value of predicted delay from both benchmark model and our solution model. In other words, the mad value is the average error delay of the predicted time in minutes.

## 7. Project Design

7.1. Setting the Environment
- *python3* - programming language for the project
- *scikit-learn* - open source for k-nearest neighbor regression
- *pandas* - open source for storing the data

7.2. Exploring the Data
- Plot the delay time by date and airline.

- Explore the distribution and relationship of delay time with the features.

7.3. Preparing the Data
- Keep ‘ARRIVAL_DELAY’ column as outputs
- Keep ‘MONTH’, ‘DAY’, ‘DAY_OF_WEEK’, ‘FLIGHT_NUMBER’, ‘TAIL_NUMBER’, ‘SCHEDULED_DEPARTURE’, ‘SCHEDULED_TIME’, ‘DISTANCE’, ‘SCHEDULED_ARRIVAL’, ‘AIRLINE’, ‘ORIGIN_AIRPORT’, and ‘DESTINATION_AIRPORT’ columns as input features.
- Delete all rows such that the flights are canceled.
- Split dataset into training/testing set as 0.8/0.2 proportion to the size of data.

7.4. Developing a Benchmark Model
- Create a 3-d matrix of size 12x31x7 to store the average delay time.
- Calculate the average delay time of all flights in the same ‘MONTH’, ‘DAY’, and ‘DAY_OF_WEEK’ columns and store values in the matrix.

7.5. Developing a Solution Model
- Use *KNeighborsRegressor* from *scikit-learn* as our model.
- Use *GridSearchCV* from *scikit-learn* to choose the best parameter k for our model on training set.

7.6. Evaluating Model Performance
- Predict the delay time on testing set with our both models and calculate the performance with evaluation metric.
- Compare the mean absolute deviation of both models.