COURSE PROJECT I: DATA ANALYSIS FOR AIRLINE DELAYS

1. Objectives

The purpose of this project is to become familiarize with Python to perform basic data analysis. There are many ways to assess the quality of an airline. You will learn how to assess airlines based on historical flight data. In the first part of the project, we will determine which airline is the fastest based on flight time calculations. In the second project, we will develop a very useful predictive model, **Naive-Bayes Classifier** to determine the likelihood of delays and predict whether a flight will be delayed or not.

The objectives of the course projects are to:

- Use Python to read and analyze data;
- Become familiarize with different data structures such as *lists* and *dictionaries*;
- Use conditionals and loops to process data, fix anomalies and perform calculations;
- Perform basic plotting in Python using the package *matplotlib*;
- Apply the concept of conditional probability and *Bayes' Theorem*;
- Develop a simple *Naive-Bayes classifier* model using m-estimates.

Deliverables.

- You should code with Jupyter notebook, and provide detailed problem description, Python code, and output for each exercise listed below. You can work on template provided.
- Your should submit a zip folder named **IE300-ADx-Group$y$-Lastname1-Lastname2-Lastname3.zip**, where x is your session number, y is your group number. The folder should contain a *.ipynb* file that contains all of your work, and a *.pdf* file saved from Jupyter Notebook, both with the same file names as the zip folder. You can also provide any supplementary files if needed in the zip folder.
- Only one member of your group should submit the project on **Gradescope**.
- The deadline for the project is **March 27 (Friday), midnight at 11:59pm.**

2. How do we determine which airline is the "fastest"?

Consider the following scenario. Airline A says they will get you from LGA to DCA in 45 minutes, but took 60 minutes instead. On the same route, Airline B says it will only take 70 minutes but actually took 65 minutes. Which airline do you consider the fastest?

Most people will say Airline A since the trip took 60 minutes as opposed to 65. However, according to the government, Airline A is "late" since it took longer than 15 minutes to arrive. Because of this, most airlines pad their schedules by saying flight times are longer than they usually are. How do we accurately determine which airline is the fastest? An approach using historical data was proposed in [1]. In this case study, we will reproduce some of the results found in the study to determine the fastest airlines.

The dataset *FlightTime.csv* contains 743 observations of flights from ORD to LAX throughout November 2015. There are 10 variables.

Variable Number	Variable Name	Description
1	Flight Date	Date of the flight (mm/dd/yyyy)
2	Carrier	IATA Carrier Identification
3	Flight Number	Number assigned to the flight
4	Origin	Origin airport
5	Destination	Destination airport
6	Departure Time	Time of departure (hhmm)
7	Depature Delay	Difference between scheduled and actual departure time (minutes)
8	Arrival Time	Time of arrival (hhmm)
9	Arrival Delay	Difference between scheduled and actual arrival time (minutes)
10	Flight Time	Difference between actual departure and arrival time (minutes)

Reference: Bureau of Transportation Statistics

There are four types of time we are interested in:

(1) **Flight Time**: The difference between the departure time and arrival time.
(2) **Average Flight Time**: The average time an airline takes to complete the route.
(3) **Target Flight Time**: An estimate of how long a flight should take based on distance and direction of travel. This is calculated based on the spherical distance (great circle distance). There are many ways to compute this but we will use a simplified formula that does not include the complex variables like windspeed, jetstreams, etc. Let l_{ori} and l_{des} is the longitude of the origin and destination respectively, and d be the spherical distance between two points. Assuming a constant velocity and an average time of 20 minutes to runway time, define $TFT = 0.117 * d + .517 * (l_{ori} - l_{des}) + 20$.
(4) **Typical time**: Calculated by the target time plus the average delay associated with the origin and departure airport.

The measure of an airline's performance is determined by the difference between average flight time and typical time which is called **time added**. The lower the time added, the faster the airline is on that particular route.

Exercise 1: Using *FlightTime.csv*, write a Python program to perform the following.

(1) Read the dataset, ignoring any observations without a recorded departure or arrival time. Some times are recorded incorrectly resulting in incorrect flight time. For any flight time less than 230 minutes, delete the observations.

(2) Calculate the number of observations in your dataset.

(3) Calculate the *target flight time* with $d = 1741.16$ mi, $l_{ori} = -87.90°$ and $l_{des} = -118.41°$.

(4) Calculate the *typical time* of this route. You will need to get the average of the departure and arrival delays and add it to the target flight time.

(5) Calculate the time added **for each airline** and determine which airline have the lowest time added. Note, you will need to calculate the average flight time for each airline.

(6) Output the results.

Exercise 2: Using *matplotlib*, create a bar graph for the time added of each airline. Be sure to label your axes and plot.

---

3. Modeling ad Predicting Flight Delays

As any frequent flier knows, flight delays are an inevitable part of commercial air travel. In the previous case study, you were able to determined which airline is the fastest for a particular route. In doing so, you had to take into account departure and arrival delays of each airline. While the cause for delays are largely unpredictable, we can make some insights on which flights are more likely to be delayed without taking weather conditions into consideration.

An obvious predictor of delays is the airline itself. For example, in 2014, only 74.34% of American Airline's flights were on-time while Delta had a 85.21% on-time record [2]. How can we use historical data to make predictions about delays? You may have already learned about several predictive models such as ordinary least-squares. We will study a very popular model in statistics and machine learning called *Naive-Bayes.*

3.1. **Naive-Bayes Classifier.** Recall that *Bayes' Theorem* tells you the probability of an event based on conditions possibly related to that event. That is, given some event A and B,

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)} \tag{1}$$

The Naive-Bayes classifier is based on Bayes' Theorem with the assumption of independence between attributes. Suppose we have a vector $X = (x_1, x_2, ..., x_n)$ representing n **attributes** (independent variables). We wish to compute

$$P(C_k|X) = \frac{P(C_k)P(X|C_k)}{P(X)} \tag{2}$$

where C_k represents possible outcomes (**classes**). If we are trying to decide whether a particular flight is likely to be delayed, the outcome would be from the set $C_k \in \{Y, N\}$, corresponding to the flight being delayed (Y) or not (N). The vector X would contain information about the flight (e.g., airline, origin airport, destination airport) that could give some insight into which class will occur. One approach to picking a class is to pick the value C_k that is most probable, given that we know the values of the attributes. In other words, we pick the C_k that maximizes $P(C_k|X)$. Since the vector X is known, and $P(C_k|X) = P(C_k \cap X)/P(X)$ by the definition of conditional probability, then $P(X)$ will be the same for all possible classes, and we can instead focus on selecting the class that maximizes $P(C_k \cap X)$.

Noting that the attribute vector X typically contains values of several attributes, so we can rewrite $P(C_k \cap X) = P(C_k \cap x_1 \cap x_2 \cap \ldots \cap x_n)$, where event x_i tells us the particular value of attribute i. In practice, the symbol $\cap$ is often replaced with a comma when intersecting many events. By repeatedly applying the definition of conditional probability, we have

$$\begin{aligned} P(C_k \cap X) &= P(C_k)P(X|C_k) \\ &= P(C_k)P(x_1, ..., x_n|C_k) \\ &= P(C_k)P(x_1|C_k)P(x_2, ..., x_n|C_k, x_1) \\ &\;\;\vdots \\ &= P(C_k)P(x_1|C_k)P(x_2|C_k, x_1) \cdots P(x_n|C_k, x_1, x_2, ..., x_{n-1}) \end{aligned} \tag{3}$$

This decomposes $P(C_k \cap X)$ into a product of the probability of outcome C_k with conditional probabilities of each attribute. We assume **conditional independence** between each attributes (hence, the "naive" portion of the name). That is, $P(x_i|C_k, x_j) = P(x_i|C_k)$ where $i \neq j$, which allows us to discard all conditions except C_k in Equation (2). Hence, this equation simplifies to

$$P(C_k|X) \propto P(C_k) \prod_{i=1}^{n} P(x_i|C_k) \tag{4}$$

With this definition, we can define a **classifier** (a decision rule). A common classifier is to pick the most probable class, as discussed above. This is called the **maximum a posterior (MAP) rule** which assigns a label $\hat{y} = C_k$ for some k:

$$\hat{y} = \underset{k}{\operatorname{argmax}}\, P(C_k) \prod_{i=1}^{n} P(x_i|C_k) \tag{5}$$

3.2. **M-Estimates.** The conditional densities $P(x_i|C_k)$ can be defined in different ways (e.g., Normal, Poisson) depending on the problem. If our dataset is small, we often do not know the underlying distributions. An intuitive way to estimate $P(x_i|C_k)$ is to define

$$P(x_i|C_k) = \frac{\hat{n}}{n} \tag{6}$$

where $\hat{n}$ is the number of observations which $C = C_k$ and $X = x_i$ and n is the number of observations $C = C_k$. However, for small data sets we may find that that $\hat{n} = 0$ or $n = 0$, which pose a problem. A way to avoid fix this problem is to use *m-estimates*:

$$P(x_i|C_k) = \frac{\hat{n} + mp}{n + m} \tag{7}$$

where m is called the equivalent sample size and p is the a priori estimate of $P(x_i|C_k)$. A typical choice for p is $p = 1/$ (# of possible values for attribute i), which assumes that all attribute values are equally likely. The idea behind m-estimates is to pretend we have m extra observations with class $C = C_k$, with $m \cdot p$ of them having attribute $X = x_i$. Essentially, m says how confident we are of our prior estimate p since as m increases, we have $P(x_i|C_k) \to p$.

Let us consider an example problem.

Example 1: Consider the following table containing information of delays.

Obs	Origin	Destination	Airline	Delay
1	DFW	ORD	Delta	Y
2	DFW	ORD	Delta	N
3	DFW	ORD	Delta	Y
4	LAX	ORD	Delta	N
5	LAX	ORD	United	Y
6	LAX	LGA	United	N
7	LAX	LGA	United	Y
8	LAX	LGA	Delta	N
9	DFW	LGA	United	N
10	DFW	ORD	United	Y

Table 1. Flight Delay 1

The class is the variable Delay and the attributes are Origin, Destination and Airline. Classify DFW-LGA on Delta using Naive-Bayes with MAP and m-estimates. Assume $m = 3$.

Solution: First, note that DFW-LGA on Delta is not contained in the table. However, we can compute the probability of delay for this flight using Equation (5). Using an equivalent sample size of $m = 3$, we compute the relevant conditional probabilities as follows:

$$
\begin{aligned}
P(DFW|Y) &= \frac{3+3(0.5)}{5+3} = 0.5625 \\
P(DFW|N) &= \frac{2+3(0.5)}{5+3} = 0.4375 \\
P(LGA|Y) &= \frac{1+3(0.5)}{5+3} = 0.3125 \\
P(LGA|N) &= \frac{3+3(0.5)}{5+3} = 0.5625 \\
P(Delta|Y) &= \frac{2+3(0.5)}{5+3} = 0.4375 \\
P(Delta|N) &= \frac{3+3(0.5)}{5+3} = 0.5675
\end{aligned}
$$

For $P(DFW|Y)$, we have five observations where $C_k = Y$, of which three observations have $x_i = DFW$. So, we have that $n = 5$, $\hat{n} = 3$ and $p = 0.5$ since there are only two values of the attribute Origin (DFW and LAX). Since Y and N each occur in five of the ten data points, we compute $P(Y) = P(N) = 0.5$, we apply Equation (5) to obtain

$$
\begin{aligned}
P(Y|DFW, LGA, Delta) &\propto P(Y)P(DFW|Y)P(LGA|Y)P(Delta|Y) \\
&= 0.0385
\end{aligned}
$$

and

$$
\begin{aligned}
P(N|DFW, LGA, Delta) &\propto P(N)P(DFW|N)P(LGA|N)P(Delta|N) \\
&= 0.0698
\end{aligned}
$$

Since $\hat{y} = \text{argmax}\,\{0.0385, 0.0698\} = 0.0698$, we classify DFW-LGA on Delta as N.

Exercise 3: Using the Table 2, will the flight SEA-ATL on Southwest with good weather be delayed? Use m-estimates for the conditional probabilities with $m = 4$. Print your results.

Obs	Origin	Destination	Airline	Weather	Delay
1	SEA	ATL	Southwest	Poor	N
2	SEA	BOS	American	Good	Y
3	SEA	ATL	United	Poor	Y
4	SFO	BOS	Southwest	Poor	Y
5	SFO	ATL	American	Good	N
6	SFO	BOS	United	Poor	Y
7	SFO	BOS	United	Good	N
8	SEA	BOS	Southwest	Poor	Y

Table 2. Flight Delay 2

Let us now use real historical flight data to predict flight delays. The dataset *FlightDelay.csv* contains 2,346 observations of flights departing from JFK or SFO to ATL, LAS or ORD in November 2015. There are 5 variables.

Variable Number	Variable Name	Description
1	Carrier	IATA Carrier Identification
2	Origin	Origin airport
3	Destination	Destination airport
4	Depature Delay	Difference between scheduled and actual departure time (minutes)
5	Arrival Delay	Difference between scheduled and actual arrival time (minutes)

Reference: Bureau of Transportation Statistics

Exercise 4: Using *FlightDelay.csv*, write a Python program to perform the following.

(1) Read the dataset.

(2) Compute the total delay time for each flight. The total delay time is the sum of departure delay and arrival delay time.

(3) Assign any flight with total delay time greater than 15 minutes with "'Y" for being delayed. Otherwise, the flight is not delayed and is assigned "N".

Exercise 5: Using your Python program from the previous exercise, for each of the following flights, compute the probabilities of delay and not delay given the route and carrier and then classify whether the flight is delayed using Naive-Bayes (assume $m = 3$).

(1) JFK-LAS on American Airlines (AA)

(2) JFK-LAS on JetBlue (B6)

(3) SFO-ORD on Virgin Airlines (VX)

(4) SFO-ORD on Southwest Airlines (WN)

Your output should contain $P(Delay|Ori, Dest, Carr), P(Notdelay|Ori, Dest, Carr)$ and a classification of whether the flight is delayed or not.

References

[1] "How We Found The Fastest Flights." *How We Found The Fastest Flights.* 11 Mar. 2015. Web. 1 Aug. 2015.
[2] "The World's Best and Worst On-Time Airlines " *Skift.* 18 Nov. 2014. Web. 5 Aug. 2015.
[3] Mitchell, Tom M. "Machine Learning." New York: McGraw-Hill. 1997.