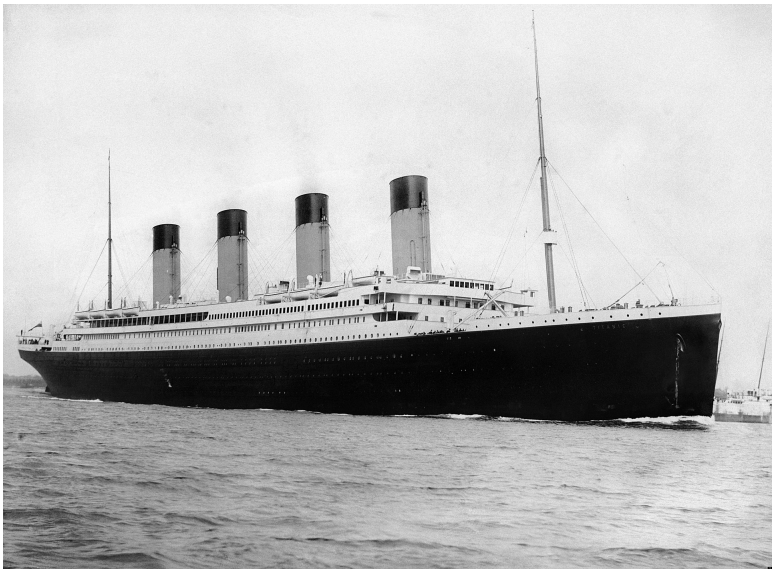
Lab 2: Gathering and Cleaning Data



In this inaugural lab we will be working with the Titanic data set. The Titanic was a passenger ship that (in)famously sank on 15 April 1912 with an estimated 1,500 people lost in the wreck. The Titanic dataset is a classic, canonical data set in the field of data science and machine learning. It is often used for demonstrating classification algorithms. This dataset contains demographic information about passengers who were on board the Titanic as well as whether they survived or not. We will focus on loading the data, building an understanding of what is in the data set, and cleaning the data for later use.

1 Reading in the data

1. First read the data into your python development environment using pandas. You'll need to import pandas and then apply the '.read_csv' method.

2. Second, there will be times when built in methods like read_csv will fail, particularly when the data you are working with has a more complex structure (we will deal with at least one data set like that this term). We will take this opportunity to practice reading in data from scratch on a simple data set. There are many ways to do this. In this lab, we will read the data into a dictionary where the keys are variable names and the values are lists. To do this, add to the following code:

```
data = {}    #empty dictionary

with open('titanic.csv') as file:    #open connection to the file
    for line in file:    #loop over lines
        ...
```

Each line will be read in as a string – you will need to use the .split method to parse this string in various ways. Once you have pulled out an individual data point append it to the list associated with the relevant variable in your dictionary. Make sure to watch out for lines that cause problems. Finally, turn this into a pandas data frame.

3. Answer the following questions in your writeup:

   - What do the rows and columns of the data frames you have read in mean? What kind of information is contained in each?
   - Use the .info and .describe methods on the data frame that comes from using .read_csv and on the data frame that comes from the scratch built approach. What kind of information do these two methods generate? What differences do you observe between the results for the two data frames?
   - For each variable in the data describe what the variable measures and any relevant information such as the meanings of individual codes, the units, etc.

2 Representing the data

1. Variables can either be represented as numerical (integer or float) or categorical. Consider the Pclass, SibSp, Parch, Fare, and Cabin variables. Count the number of unique values for each. Decide the best way to represent each – as integer, float, or categorical?

2. Convert the following variables to categorical variables: Pclass, Sex, Cabin, and Embarked. You may find the .astype method to be useful.

3. Plot a bar chart of the Survived variable. You may find the Seaborn countplot function to be useful for this.

4. Answer the following questions in your writeup:

   - Regarding Pclass, SibSp, Parch, Fare, and Cabin variables, why did you choose integer, float, or categorical for each?
   - Describe the results of plotting the Survived variable – what do you learn about your data set? How does it compare to the historical survival rate aboard the Titanic?
   - Identify a variable which would make a good label/target/response/dependent variable for a supervised machine learning problem – would that variable be suitable for classification or regression?

3 Missing values

1. The variable Age appears to have a substantial number of missing values. We will practice filling them in. First, create three new columns in your data set called Age_fill_mean, Age_fill_median, Age_fill_mode by copying the original Age variable. Fill the missing values in Age_fill_mean with the mean Age in the data, in Age_fill_median with the median Age in the data, and in Age_fill_mode with the modal Age in the data. You may find the .fillna method from pandas, the .mean and .median methods from numpy, and the .mode method from scipy.stats to be useful here.

2. Next let's practice using some very simple supervised machine learning to fill in the missing values, specifically the k-nearest neighbors (kNN) algorithm. To do this:

   - Create another new column in your data set called Age_fill_knn;
   - Import KNeighborsRegressor from sklearn.neighbors;
   - Divide your data into two data sets, one called data_drop_age_na in which there are no missing values for Age and a second called data_age_na that contains all the rows with missing values for Age – you may find the .isna and .notna methods to be useful;
   - Instantiate and fit a kNN model using data_drop_age_na where n_neighbors is set to 5, the features used are SibSp, Parch, and Fare, and the response is Age. Then use this fitted model to predict in the data_age_na data frame;
   - Finally, use these predicted values to fill in the missing values in the original, full data set in the Age_fill_knn column.

3. Answer the following questions in your writeup:

   - How many passengers in this data set are missing age information?
   - Describe what you did in applying the kNN algorithm to fill missing values.
   - Use density plots to compare the results of these different methods of filling missing values. For example, import the seaborn library and use kdeplot. How do you interpret the resulting plots? What do the x and y axes mean? What do they tell us about the effects of these different methods of filling in missing values for Age?
   - Consider the Cabin variable. Why would this variable not work well with the methods we applied to fill in missing values for Age? Identify at least one reason that has to do with interpretation rather than with code.

4 Submission Instructions

- Write up the answers to the questions in a short 2 page word document. Add footnotes identifying which sentence addresses which questions. Use complete sentences – your goal is to explain your dataset to the audience, what you've done to it, and what you've

learned about it. Accordingly, you may seek to emulate some of the sections of the whale paper describing their data. Include the plots you've generated. Convert this to pdf and submit it. Submit your .ipynb file as well.

- The grading rubric for this assignment will be available in Canvas.
- NO OTHER SUBMISSION TYPES WILL BE ACCEPTED.
- **Late policy**: 5% of total points deducted per day for three days – after that no submissions allowed.