Problem Set 1: Titanic Survivor Data

1 – A Simple Heuristic

```
import numpy as np

import pandas as pd

import statsmodels.api as sm

def simple_heuristic(file_path):
    '''
    In this exercise, we will perform some rudimentary practices similar to those of
    an actual data scientist.

    Part of a data scientist's job is to use her or his intuition and insight to
    write algorithms and heuristics. A data scientist also creates mathematical models
    to make predictions based on some attributes from the data that they are examining.

    We would like for you to take your knowledge and intuition about the Titanic
    and its passengers' attributes to predict whether or not the passengers survived
    or perished. You can read more about the Titanic and specifics about this dataset at:
    http://en.wikipedia.org/wiki/RMS_Titanic
    http://www.kaggle.com/c/titanic-gettingStarted

    In this exercise and the following ones, you are given a list of Titantic passengers
    and their associated information. More information about the data can be seen at the
    link below:
    http://www.kaggle.com/c/titanic-gettingStarted/data.

    For this exercise, you need to write a simple heuristic that will use
    the passengers' gender to predict if that person survived the Titanic disaster.

    You prediction should be 78% accurate or higher.

    Here's a simple heuristic to start off:
      1) If the passenger is female, your heuristic should assume that the
      passenger survived.
      2) If the passenger is male, you heuristic should
      assume that the passenger did not survive.

    You can access the gender of a passenger via passenger['Sex'].
```

```
If the passenger is male, passenger['Sex'] will return a string "male".
If the passenger is female, passenger['Sex'] will return a string "female".

Write your prediction back into the "predictions" dictionary. The
key of the dictionary should be the passenger's id (which can be accessed
via passenger["PassengerId"]) and the associated value should be 1 if the
passenger survied or 0 otherwise.

For example, if a passenger is predicted to have survived:
passenger_id = passenger['PassengerId']
predictions[passenger_id] = 1

And if a passenger is predicted to have perished in the disaster:
passenger_id = passenger['PassengerId']
predictions[passenger_id] = 0

You can also look at the Titantic data that you will be working with
at the link below:
https://www.dropbox.com/s/r5f9aos8p9ri9sa/titanic_data.csv
'''
predictions = {}

df = pd.read_csv(file_path)

for passenger_index, passenger in df.iterrows():

    passenger_id = passenger['PassengerId']

    # Your code here:
    # For example, let's assume that if the passenger
    # is a male, then the passenger survived.
    if passenger['Sex'] == 'female':

        predictions[passenger_id] = 1

    else:

        predictions[passenger_id] = 0

return predictions
```

```
Your heuristic is 78.68% accurate. Is it 78% or better?
```

2 - A More Complex Heuristic:

```
import numpy
import pandas
import statsmodels.api as sm

def complex_heuristic(file_path):
'''
    You are given a list of Titantic passengers and their associated
    information. More information about the data can be seen at the link below:
    http://www.kaggle.com/c/titanic-gettingStarted/data

    For this exercise, you need to write a more sophisticated algorithm
    that will use the passengers' gender and their socioeconomical class and age
    to predict if they survived the Titanic diaster.

    You prediction should be 79% accurate or higher.

    Here's the algorithm, predict the passenger survived if:
    1) If the passenger is female or
    2) if his/her socioeconomic status is high AND if the passenger is under 18

    Otherwise, your algorithm should predict that the passenger perished in the disaster.

    Or more specifically in terms of coding:
    female or (high status and under 18)

    You can access the gender of a passenger via passenger['Sex'].
    If the passenger is male, passenger['Sex'] will return a string "male".
    If the passenger is female, passenger['Sex'] will return a string "female".

    You can access the socioeconomic status of a passenger via passenger['Pclass']:
    High socioeconomic status -- passenger['Pclass'] is 1
    Medium socioeconomic status -- passenger['Pclass'] is 2
    Low socioeconomic status -- passenger['Pclass'] is 3

    You can access the age of a passenger via passenger['Age'].

    Write your prediction back into the "predictions" dictionary. The
    key of the dictionary should be the Passenger's id (which can be accessed
    via passenger["PassengerId"]) and the associated value should be 1 if the
    passenger survived or 0 otherwise.

    For example, if a passenger is predicted to have survived:
    passenger_id = passenger['PassengerId']
```

predictions[passenger_id] = 1

And if a passenger is predicted to have perished in the disaster:
passenger_id = passenger['PassengerId']
predictions[passenger_id] = 0

You can also look at the Titantic data that you will be working with
at the link below:
https://www.dropbox.com/s/r5f9aos8p9ri9sa/titanic_data.csv
'''

```
predictions = {}
 df = pandas.read_csv(file_path)
 for passenger_index, passenger in df.iterrows():
            passenger_id = passenger['PassengerId']

      # your code here
      # for example, assuming that passengers who are male
      # and older than 18 surived:

          if passenger['Sex'] == 'female' or passenger['Age'] < 18 and passenger['Pclass'] == 1:
                     predictions[passenger_id] = 1
          else:
                     predictions[passenger_id] = 0
 return predictions
```

```
Your heuristic is 79.12% accurate. Is it 79% or better?
```

3 - Your Custom Heuristic

```
import numpy
import pandas
import statsmodels.api as sm

def custom_heuristic(file_path):
```

'''

You are given a list of Titantic passengers and their associated
information. More information about the data can be seen at the link below:
http://www.kaggle.com/c/titanic-gettingStarted/data

For this exercise, you need to write a custom heuristic that will take
in some combination of the passenger's attributes and predict if the passenger
survived the Titanic diaster.

Can your custom heuristic beat 80% accuracy?

The available attributes are:

Pclass	Passenger Class (1 = 1st; 2 = 2nd; 3 = 3rd)
Name	Name
Sex	Sex
Age	Age
SibSp	Number of Siblings/Spouses Aboard
Parch	Number of Parents/Children Aboard
Ticket	Ticket Number
Fare	Passenger Fare
Cabin	Cabin
Embarked	Port of Embarkation (C = Cherbourg; Q = Queenstown; S = Southampton)

SPECIAL NOTES:
Pclass is a proxy for socioeconomic status (SES)
1st ~ Upper; 2nd ~ Middle; 3rd ~ Lower

Age is in years; fractional if age less than one
If the age is estimated, it is in the form xx.5

With respect to the family relation variables (i.e. SibSp and Parch) some relations were ignored. The following are the definitions used for SibSp and Parch.

Sibling: brother, sister, stepbrother, or stepsister of passenger aboard Titanic
Spouse: husband or wife of passenger aboard Titanic (mistresses and fiancees ignored)
Parent: mother or father of passenger aboard Titanic
Child: son, daughter, stepson, or stepdaughter of passenger aboard Titanic

Write your prediction back into the "predictions" dictionary. The key of the dictionary should be the passenger's id (which can be accessed via passenger["PassengerId"]) and the associating value should be 1 if the passenger survvied or 0 otherwise.

For example, if a passenger is predicted to have survived:

```
passenger_id = passenger['PassengerId']
predictions[passenger_id] = 1
```

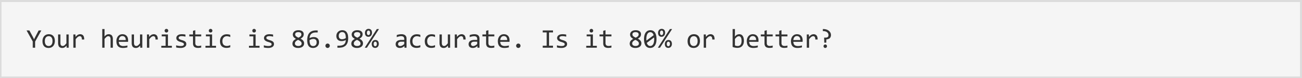
And if a passenger is predicted to have perished in the disaster:

```
passenger_id = passenger['PassengerId']
predictions[passenger_id] = 0
```

You can also look at the Titantic data that you will be working with at the link below:

```
    https://www.dropbox.com/s/r5f9aos8p9ri9sa/titanic_data.csv
    '''

predictions = {}
df = pandas.read_csv(file_path)
for passenger_index, passenger in df.iterrows():
 # your code here
        passenger_id = passenger['PassengerId']
        if (passenger['Sex'] == 'female' or passenger['Pclass' != 3]) or (passenger['Age'] < 18):
            predictions[passenger_id] = 1
        else:
            predictions[passenger_id] = 0
 return predictions
```



```
Your heuristic is 86.98% accurate. Is it 80% or better?
```