**CS 5010 Final Project Report**

**Understand, analyze and predict loneliness of young people**

Prepared By: Kia Duan(wd3fg)

Huitong (Jo) Pan (hp4zw)

Teresa Sun (js6sm)

**Introduction**

For this project, our initial motivation was to explore general trends of preferences and feelings across different activities and perspectives for young people. One of our group mates felt lonely from time to time, so we are especially concerned with issue. Our goal was to predict if someone is lonely around us so that we can take care of them. Taking it from there, we were able to narrow down our question of interests to loneliness of young people. First, we would like to figure out if specific groups demonstrate loneliness. Furthermore, we wanted to investigate what factors correlate with loneliness. How can we predict if someone is lonely was our next step. Finally, we wanted to establish a simple application to test loneliness.

**The data**

The dataset is publicly shared on Kaggle, which contains 1010 young people’s responses to 150 questions from a survey conducted in 2013. Here the range for the age group is from 15 to 30. As we all fall into this age group, we are interested in finding out if there’s any trend for young people in general. There are two data files both in csv format. ‘responses.csv’ contains all the responses from the survey. It has 150 columns of which 139 are integer and 11 are categorical variables. They can be split into 8 categories: Music preferences, Movie preferences, Hobbies & interests, Phobias, Health habits, Personality traits, views on life, & opinions, Spending habits, and Demographics. ‘columns.csv’ mainly maps the original (long) variables with their shorted versions. In order to make the dataset ready to use, we first imported the dataset to Python using Pandas and then dropped any missing values. After all the pre processing, we ended up having 674 observations with 150 variables.

**Algorithm**

In this section, we only cover the major functions that are essential to our analysis. Details can be found in the actual codes and comments. Following the data preprocessing steps, we produced two functions to calculate the confidence intervals. The first function “CI” calculates the 95% confidence interval for mean for the group of interest. The second function “twosampCI” returns the 95% confidence interval for difference in sample means for two groups. With these two functions, scipy library was used to calculate the critical values for confidence intervals. These two functions can be applied anytime when considering 95% Confidence Intervals for specific groups or across different groups.

Next, we wrote three functions that conducts hypothesis test. The first function “check\_values\_in\_varibale” returns all unique values of a variable as a list. The second function “divide\_groups” divides the variable of interest into the unique values subsets. The third function “hypothesis\_test” calls the previous functions, using a stats built-in function of scipy library, to return the p-values of the t-test. This function can be used to compare any two numerical data lists, and users can use this function if they are interested in any other variables other than loneliness.

The central piece of our program is a logistic regression model and a loneliness test. The logistic regression model uses 71 independent variables, using K-fold cross validation to get a 70% prediction accuracy for our test set. The function “calculateloneliness” is based on the logistic regression model. Lastly, the function “areyoulonely” takes user input and calls “calculateloneliness”, outputting the loneliness level of the user.

Our program also includes several functions for plotting graphs such as the correlation matrix (**Appendix 1**). Overall, our program has user interaction with the user, taking inputs from the user and producing outputs, their loneliness level, that users might find interesting. We believe that our CI function, hypothesis test function and logistic regression go beyond our basic queries and can be used for other datasets (and deserve some extra credits :))

**Results**

In this section, we present the major findings for the queries we raised at the beginning of the report.

**1. Are there any gender differences in loneliness?**

* We produced 95% Confidence Interval for the mean of loneliness level for male group and female group.
* 95% CI for male is [2.63, 2.90] shown in **Fig.1**.
* 95% CI for female is [2.87, 3.09] shown in **Fig.2.**
* 95% CI for the difference between two groups was also calculated: [-0.39, -0.04].
* Hypothesis test suggested that the difference is statistically significant. So we are able to make the conclusion that young females are generally more lonely than young males.

|  |  |
| --- | --- |
| male.png | female.png |
| **Fig.1** | **Fig.2** |

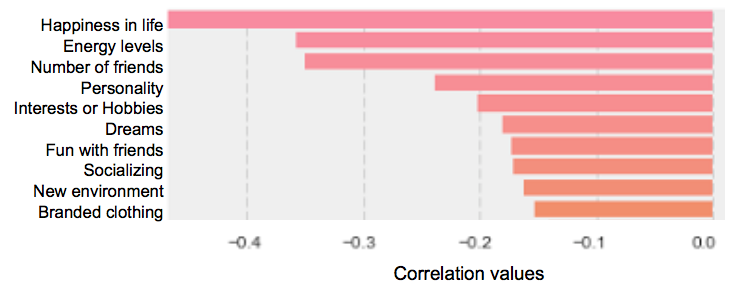
**2. Are people without any siblings more lonely than those who have?**

* We produced 95% Confidence Interval for the mean of loneliness level for people who have siblings and those don’t have siblings.
* 95% CI for single child: [2.72, 3.10] shown in **Fig.3.**
* 95% CI for those have siblings: [2.79, 2.98] shown in **Fig.4**.
* 95% CI for the difference between two groups was also calculated: [-0.19, 0.24].
* Hypothesis test for these two groups generated p-value of 0.81 that is too large.
* Hypothesis test suggested that there’s no statistically significant difference between two groups. Thus we could not conclude that single children are more lonely than people who have siblings.

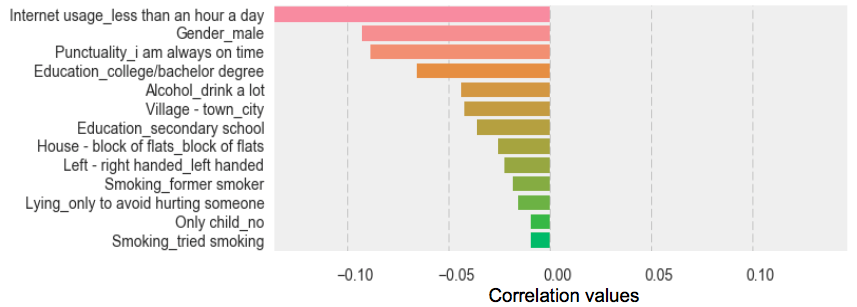
|  |  |
| --- | --- |
| singlechild.png | morechild.png |
| **Fig.3** | **Fig.4** |

**3. What traits are associated with loneliness?**

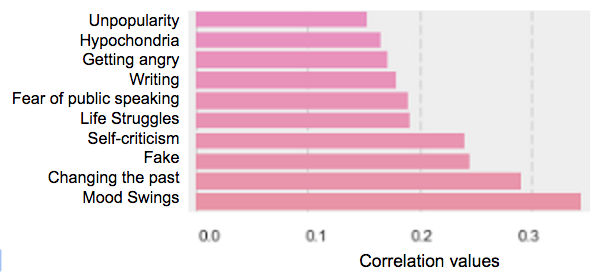
Since numerical variables can’t be directly compared with categorical variables, we did two correlation analyses, one with numerical variables(Appendix 2) and another with categorical variables (Appendix 3). As shown in **Fig.5**, not surprisingly, happiness, energy level and friends are negatively correlated with loneliness level. As illustrated in **Fig.6**, people who rarely use internet also seem to demonstrate low loneliness level. Interestingly, a person who likes branded clothing is also unlikely to be lonely. On the other end of the spectrum, shown in **Fig.7** and **Fig.8**, people who have mood swings and have an education degree of primary school tend to be lonely, probably because the college students are too busy with coursework to feel lonely.



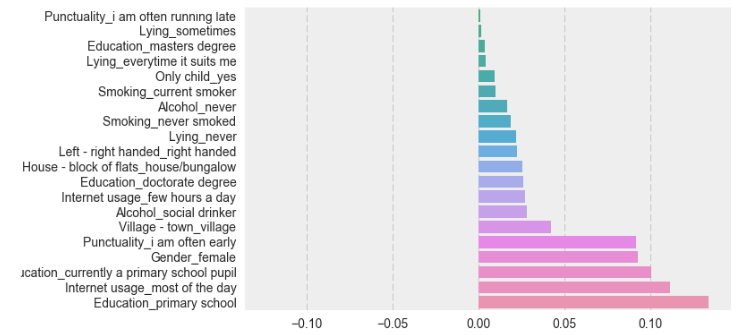
**Fig.5**



**Fig.6**



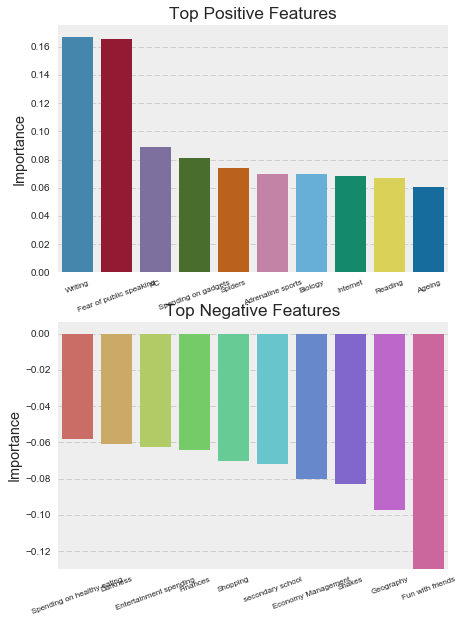
**Fig.7**



**Fig.8**

**4. How to predict a young person’s loneliness?**

We firstly transformed our original loneliness variable to a dummy variable. Loneliness > 3 are considered as lonely. Loneliness<=3 are considered not lonely. Using our logistic regression model with 71 variables, we can predict a person’s loneliness level at an accuracy rate of 72% with (K-fold) test sets. The variables with largest coefficients can be found in **Fig.9**. In short, imagine a person likes writing, fear public speaking, enjoys using PC and spends a lot a gadgets, If he also happens to not like spending time with friends, hates geography and economy, doesn’t have problems with snakes, then chances are that he is feeling pretty lonely.



**Fig.9**

**5. How can these findings be used?**

As mentioned in the algorithm section above, we built a simple loneliness test based on the top 4 positive features and the top 4 negative features in our logistic regression model. The loneliness test takes user input, calculates the user’s loneliness level and displays it. Users can test their own or anyone else’s loneliness level, if interested.

**Testing**

In total, we wrote three classes of unit tests, each consisting two unit tests. The first class tests the “CI” and “twosampCI” function. We tested if these two function return the correct confidence intervals using the test case list1 and list2, as shown in the **sample run1**. The second class tests the “calculateloneliness” and “areyoulonely” functions. We tested if these functions calculated and displayed loneliness correctly, as shown in the **sample run2.** Our last class tests the hypothesis test functions, testing if it takes the correct variable of interest and returns the correct p-values.

**[Sample Run 1]**

import unittest

class confidenceIntervalTestCase(unittest.TestCase):

def test\_CI\_correct(self):

list1 = [1,2,3,4,5,6]

self.assertEqual(CI(pd.Series(list1)),[2.0030527802429905, 4.9969472197570095])

def test\_twosampCI(self):

list1 = [1,2,3,4,5,6]

list2 = [2,3,4,5,6,7]

self.assertEqual(twosampCI(pd.Series(list1),pd.Series(list2)),[-3.1170030603370606, 1.1170030603370606])

if \_\_name\_\_ == '\_\_main\_\_':

unittest.main()

import unittest

class confidenceIntervalTestCase(unittest.TestCase):

def test\_CI\_correct(self):

list1 = [1,2,3,4,5,6]

self.assertEqual(CI(pd.Series(list1)),[2.0030527802429905, 4.9969472197570095])

def test\_twosampCI(self):

list1 = [1,2,3,4,5,6]

list2 = [2,3,4,5,6,7]

self.assertEqual(twosampCI(pd.Series(list1),pd.Series(list2)),[-3.1170030603370606, 1.1170030603370606])

if \_\_name\_\_ == '\_\_main\_\_':

unittest.main()

..

----------------------------------------------------------------------

Ran 2 tests in 0.020s

OK

**[Sample Run 2]**

import unittest

class AreyoulonelyTestCase(unittest.TestCase):

def test\_calculate\_correct(self):

self.assertEqual(calculateloneliness(1,1,1,1,1,1,1,1),0)

def test\_areyoulonely(self):

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\* enter 1 for all questions \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print(areyoulonely())

self.assertEqual(areyoulonely(),(1,1,1,1,1,1,1,1))

if \_\_name\_\_ == '\_\_main\_\_':

unittest.main()

\*\*\*\*\*\*\*\*\*\*\*\*\*\* enter 1 for all questions \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

====== LONELINESS TEST =======

Do you think you are lonely?

1

Doesn't matter what you think.

We will see after 8 questions.

1. Public speaking: Not afraid at all 1-2-3-4-5 Very afraid of (integer)1

2. Poetry writing: Not interested 1-2-3-4-5 Very interested (integer)1

3. Internet: Not interested 1-2-3-4-5 Very interested (integer)1

4. PC Software, Hardware: Not interested 1-2-3-4-5 Very interested (integer)1

5. Socializing: Not interested 1-2-3-4-5 Very interested (integer)1

6. Economy, Management: Not interested 1-2-3-4-5 Very interested (integer)1

7. Cars: Not interested 1-2-3-4-5 Very interested (integer)1

8. I spend a lot of money on partying and socializing.: Strongly disagree 1-2-3-4-5 Strongly agree (integer)1

========== RESULT =========

You must be a joyful person!

You are only-1.61% likely to be a lonely person.

(1, 1, 1, 1, 1, 1, 1, 1)

====== LONELINESS TEST =======

Do you think you are lonely?

1

Doesn't matter what you think.

We will see after 8 questions.

1. Public speaking: Not afraid at all 1-2-3-4-5 Very afraid of (integer)1

2. Poetry writing: Not interested 1-2-3-4-5 Very interested (integer)1

3. Internet: Not interested 1-2-3-4-5 Very interested (integer)1

4. PC Software, Hardware: Not interested 1-2-3-4-5 Very interested (integer)1

5. Socializing: Not interested 1-2-3-4-5 Very interested (integer)1

6. Economy, Management: Not interested 1-2-3-4-5 Very interested (integer)1

7. Cars: Not interested 1-2-3-4-5 Very interested (integer)1

8. I spend a lot of money on partying and socializing.: Strongly disagree 1-2-3-4-5 Strongly agree (integer)1

..

========== RESULT =========

You must be a joyful person!

You are only-1.61% likely to be a lonely person.

========== RESULT =========

You must be a joyful person!

You are only-1.61% likely to be a lonely person.

----------------------------------------------------------------------

Ran 2 tests in 19.027s

OK

**Summary**

In sum, we have provided some insights into our project topic: loneliness of young people. We conclude that there exists gender differences in the loneliness level, and there is no significant difference between the loneliness level of only children and those have siblings. We were also able to develop a real-life application that might be of user's interest. Moving forward, if we have a longer time span, we would like to explore more about the outliers in this dataset. For instance, we would like to identify those who rated a “5” for loneliness and those who rated a “1”for loneliness.

**Appendix 1: Libraries used throughout the program**

# General Libraries:

import pandas as pd

import copy

import matplotlib.pyplot as plt # library for plotting correlation

import numpy as np

import seaborn as sns # library for plotting

import unittest

from scipy.stats import t, norm # library used for confidence interval

from scipy import stats # library used for hypo test

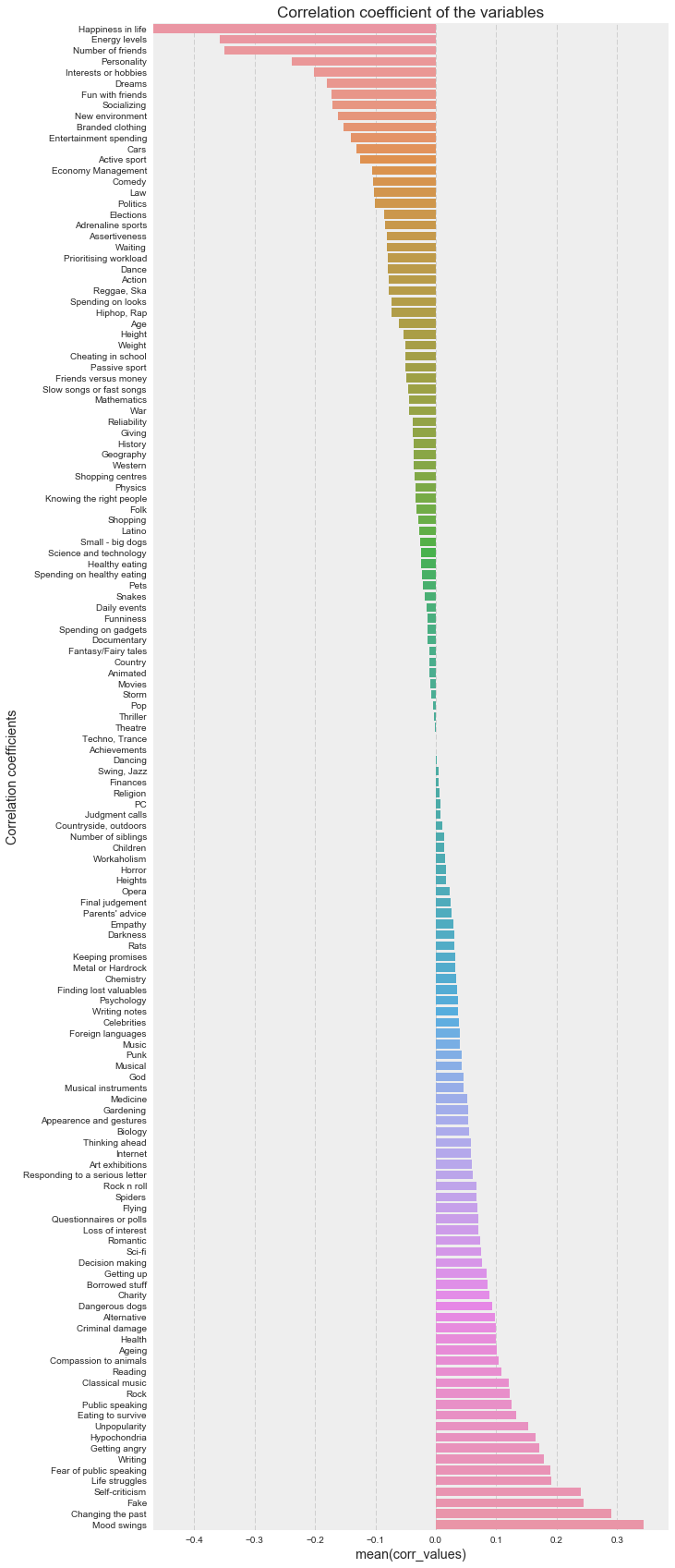
# Libraries for building logistic regression

from sklearn.cross\_validation import KFold, train\_test\_split, cross\_val\_score

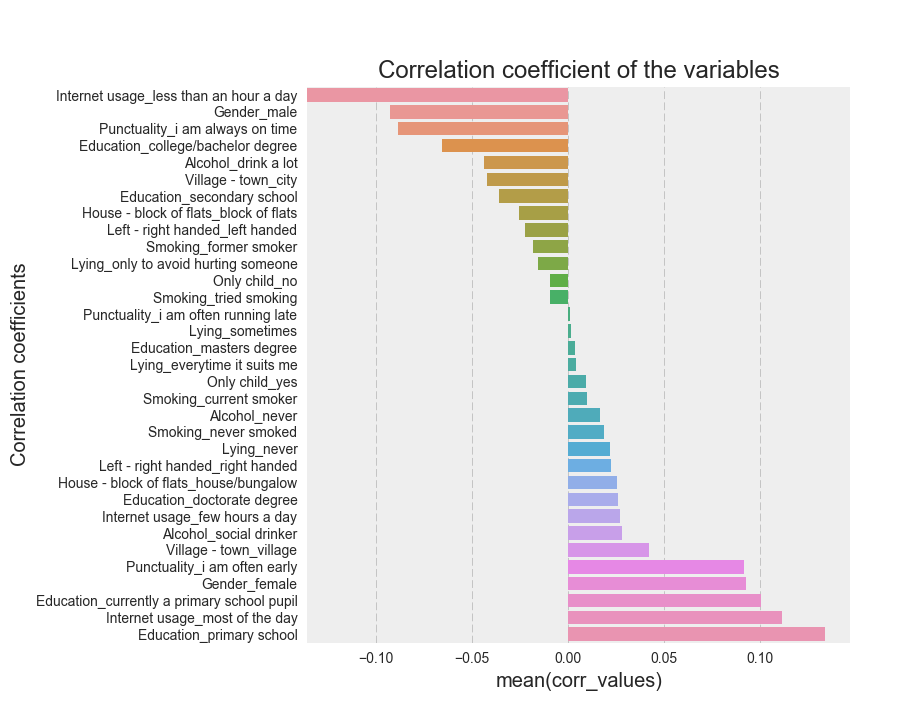
from sklearn.model\_selection import GridSearchCV

from sklearn.linear\_model import LogisticRegression

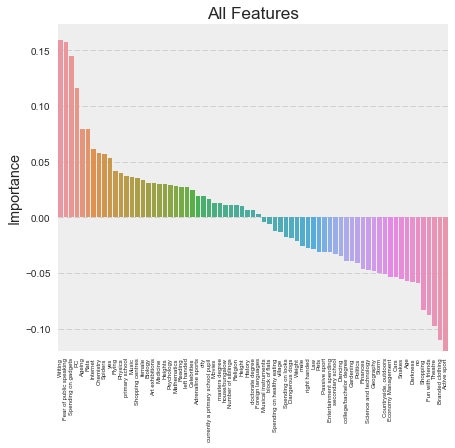
**Appendix 2: Full Correlation Plot with numeric variables**

****

**Appendix 3: Full Correlation Plot with categorical variables**

****

**Appendix 4: Full Plot with all logistic regression’s coefficients**

****