# ACCIS 6703 Intro to Data Mining Project Assignment (Spring 2021)

Project Part I (total 50 points), has **three** parts:

* I(a), 18 points
* I(b), 25 points
* I(c), 17 points

Refer to blackboard for Due Date information.

Important Notes (please read before starting the projects):

1. Student is required to finish the project independently. For details, refer to the syllabus.
2. All the homework & project are required to be written in python code and submitted in Jupyter Notebook (.ipynb) with Outputs/Results (“Run all” and “save”). If Outputs/Results are not shown, I will try to run the code from my end. If there is an error, any code/comments before the error will be graded. However, any code/comments after the error will be scored started from the basis of zero with the instructor’s best judgment for any merit, and the scores will be final without further negotiation[[1]](#footnote-1).
3. The submission can be one consolidated .ipynb (together with other files) or multiple .ipynb(s) as you see fit.
4. Do not alter the original input data. Students may not get full points if the original input data is altered outside Python/Jupyter Notebook. For example, if the inputs are provided as five csv files, the requirement is to read in all the input files in Python/Jupyter Notebook as it is. (Partial) Points will be deducted if the five files were processed outside Python/Jupyter Notebook such as Excel before read in the files.
5. If you would like to work on project part II for bonus points, you must submit part I as final and send me an email requesting the part II. Students are not allowed to submit part II without submit part I as final.
6. The project assignment may require some levels of **research effort**, including online article/code (main source), research paper as well as textbook research.
7. Updates to the project will be published in Blackboard, please pay attention to blackboard announcement. Updates/ Blackboard announcement will be mostly for clarification purpose.
8. If there is any question, please email me.

## Project Part (a) Data Analysis: Triangular Arbitrage for Cryptocurrency (18 points)

**Introduction:**

A cryptocurrency is a digital or virtual currency that uses cryptography for security. The “crypto” in cryptocurrencies refers to complicated cryptography which allows for a particular digital token to be generated, stored, and transacted securely and, typically, anonymously. Alongside this important “crypto” feature of these currencies is a common commitment to decentralization. Many cryptocurrencies are decentralized systems based on “blockchain” technology, a distributed ledger enforced by a disparate network of computers.

The first blockchain-based cryptocurrency was “Bitcoin”, which still remains the most popular and most valuable. Today, there are thousands of alternate cryptocurrencies with various functions or specifications. Some of these are clones of Bitcoin while others are forks, or new cryptocurrencies that split off from an already existing one.

For more about cryptocurrency, please refer the [Investopedia](https://www.investopedia.com/terms/c/cryptocurrency.asp).

**Triangular arbitrage of cryptocurrency:**

Consider a collection of three (3) cryptocurrencies {BTC, ETH, LTC[[2]](#footnote-2)} along with USD (US dollars). We would like to test for triangular arbitrage opportunities. Triangular arbitrage opportunities means one should be able to take in any initial capital (in any of the currencies), and return the resulting amount after a cycle (i.e. A->B->C->A). For instance, one can start with a given amount of USD, exchange for BTC, and then ETH, and then back to USD. There are several more combinations (e.g. start with BTC/ETH/LTC instead).

At any point in time, we observe the bitcoin price in terms of USD. This price is denoted by BTC-USD. You can find it here [BTC to USD](https://www.investing.com/crypto/bitcoin/btc-usd-historical-data?cid=1031044). At the same time, one can derive the value of BTC-USD implied by ETH. Here’s how it works: Observe both ETH-USD and ETH-BTC, then their ratio should represent BTC-USD. Here’s a real example: On March 10, 2019, at the same exchange, Brittrex, this is what we observe. As seen below, the derived BTC-USD (computed by division) is different from the observed BTC-USD.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ETH-USD | 135.27 |  | derived BTC-USD | 3915.6487 |
| ETH-BTC | 0.034546 |  | observed BTC-USD | 3915.3000 |

You can find more prices here: [ETH to BTC](https://www.investing.com/crypto/ethereum/eth-btc-historical-data?cid=1031045); [ETH to USD](https://www.investing.com/crypto/ethereum/eth-usd-historical-data?cid=1031047); [BTC to USD](https://www.investing.com/crypto/bitcoin/btc-usd-historical-data?cid=1031044)

**Project tasks**

(i) (6 points) Using the available historical data provided (the included csv files[[3]](#footnote-3), date range: 3/10/2018--3/10/2019), plot the time series (line charts) of the two BTC-USD prices (derived and observed) on the ***same***figure over this period. Note to use the “Price” column in each “csv” data instead of “Open”, “High” “Low” columns. Also note the date column is in reverse order, you **may or may not** need to sort it.

(ii) (6 points) Compute the spread (i.e. difference between the two time series) and plot its time series.

Compute the mean and standard deviation during this period (3/10/2018--3/10/2019). Also, show the histogram of the daily spreads to see its distribution.

(iii) (6 points) Now repeat parts (i)-(ii) for Litecoin (LTC). That is, replace ETH above with LTC. You’ll need LTC-USD, LTC-BTC, and BTC-USD.

Note: you may find the relevant prices using these links: [link1](https://www.investing.com/crypto/currency-pairs?c1=191&c2=189&exchange=1022), [link2](https://www.investing.com/crypto/currency-pairs?c1=191&c2=12&exchange=1022), [link3](https://www.investing.com/crypto/currency-pairs?c1=189&c2=12&exchange=1022)(The links are provided for information purpose. All the input data has been provided as csv files)

If you think further, you can replace LTC with other coins, and also BTC with another major crypto or [Stablecoin](https://www.investopedia.com/terms/s/stablecoin.asp). There are truly numerous combinations for triangular arbitrage in the crypto market! (Note: not your task)

Hint1: Your output should be similar or identical to the figures in “I(a) Sample Output” folder. Feel free to create your own plot titles and labels as you see fit.

Hint2: One of the input column has string dollar amount value similar to “*4,000*”. You will need to convert to numeric value *4000* before plot charts. You may consider [*replace*](https://www.w3schools.com/python/ref_string_replace.asp) or other functions to remove the comma in “*4,000*”.

## Project Part (b) Logistic Modeling (25 points)

Logistic regression, also called a logit model, is used to model dichotomous outcome variables. In the logit model the log odds of the outcome is modeled as a linear combination of the predictor variables. Here we are going to use what we learned from the class (mainly Lecture 05, and also Lecture 06, 07) to help a researcher build a logistic regression model.

The researcher is interested in how variables, such as GRE (Graduate Record Exam scores), GPA (grade point average) and prestige of the undergraduate institution, effect admission into graduate school. The response variable, admit/don’t admit, is a binary variable. There are three predictor variables: GRE, GPA scores and “rank”. We will treat the variables GRE and GPA as continuous. The variable “rank” takes on the values 1 through 4. Institutions with a rank of 1 have the highest prestige, while those with a rank of 4 have the lowest. Use all the three variables to build logistic model.

Using the data “Graduate\_School\_Admission.csv” provided.

**Project tasks**

(i) (3 points) Mimic what we learned in class, provide at least three example **codes** of Exploratory Data Analysis (“EDA”). “df.describe()” could serve as one example of EDA since it provides the count, mean, standard deviation and other information for the numeric variables. Run the codes and output results.

(*optional*) provide brief interpretation of the EDA results

(ii) (6 points) Split the data into training data and testing data. (splitting ratio is 4:1, in other word, testing sample size is 20%. And for consistency of the model results, set “random\_state = 0”)

(iii) (8 points) Show/output confusion matrix and the accuracy score, precision score, recall score, F1 score. Also provide definition of F1 score. (F1 score was not taught in class, self-research required).

(iv) (4 points) Based on results from step (3), provide your interpretation of how the model works (student can either praise the model or criticize the model, and provide your rationale).

(v) (4 points) (self-researching required) Using model make prediction: What are the estimated log-odds of graduate school admission for a student with a GPA of 3.2 and a GRE score of 670 who attended a rank 1 school? How about a student who attended a rank 2 school, but who had a GPA of 3.7 and GRE of 750? (Hint: there are many ways to make prediction using logistic model. One way to do it is make the new data the same as the X\_test format, and use similar code as the “logistic\_regression.predict(X\_test)” .

Note: for this Part I(b) logistic regression model, no feature scaling is needed.

## Project Part (c)

There are total of 4 options in this section, choose one of them as part c of your project. (If you do multiple of the following options, you will receive the one with the highest score.)

(FYI, option 3 and 4 involves coding and self researching, and thus considered harder. Thus they come with extra bonus points.)

### Option 1: Documentation of Machine Learning model code (17 points)

For part c (option 1), you were provided an Exploratory Data Analysis (“EDA”) code that is used to prepare the data/analysis for building models (“Project Part c Machine Learning EDA.ipynb”). Your task is to document the code clearly, including both high level what the code do as well as provide enough details about the code itself. The final goal of the document is aimed to help a Python beginner, who has no or little previous python/coding experience, to understand the logistic regression model code. Do assume the personnel have logistic regression model and other modeling experience, and the help he/she needs from you is purely on the Python code side.

Your document will consist of three parts:

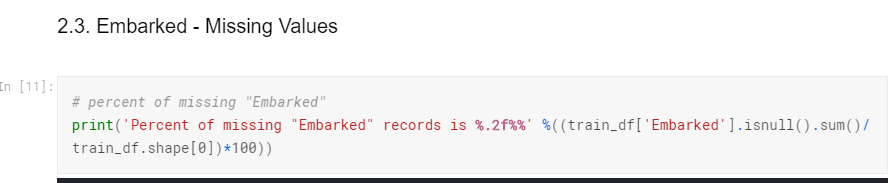
1. (7 points) In a separate document (recommended in Microsoft Word/pdf), provide insights related to the following topics/section to help the reader grasp the high level picture. Each section listed below will normally require one to two paragraphs of narrative. Feel free to write more whenever you see fit.

* Overview of the model/code
* Data Cleaning and Analytics

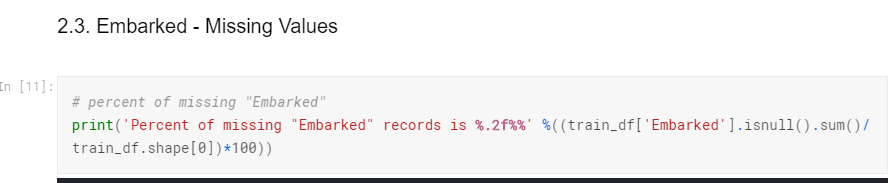
1. (10 points) In supplement to the above document, provide enough details to help the reader to understand the code within the Jupyter Notebook. You will NOT need to provide explanation to each single line of code, BUT enough details should be provided to the key code based on your understanding. Again, the ultimate goal of the document is to help a Python beginner, who has no or little previous python/coding experience, to understand clearly the logistic regression model code.

Your notation of the code can come in as either one of the following two formats:

* As comments using # inside the coding area, for example:



* Or as “Markdown” narrative outside coding area, for example:



1. (optional) Recommend additional performance tests for the logistic models.

Note: Ignore any warning signs for this project.

Your answer will be scored on how clearly your documentation is and how much help your documentation can offer the Python beginner to understand clearly the logistic regression model code.

### Option 2: Documentation of Association Rules model code (17 points)

For part c (option 2), you were provided an association rules (including Exploratory Data Analysis (“EDA”)) code file that is used to prepare the data/analysis for building models (“Project Part c Association Rules.ipynb”). Your task is to document the code clearly, including both high level what the code do as well as provide enough details about the code itself. The final goal of the document is aimed to help a Python beginner, who has no or little previous python/coding experience, to understand the logistic regression model code. Do assume the personnel have logistic regression model and other modeling experience, and the help he/she needs from you is purely on the Python code side.

Your document will consist of three parts:

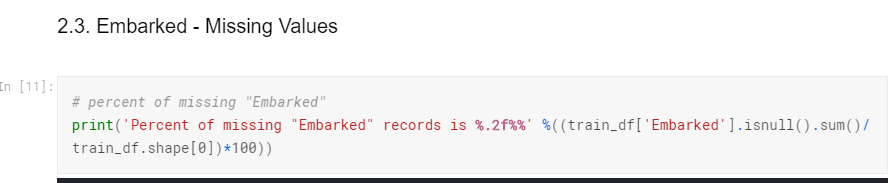
1. (7 points) In a separate document (recommended in Microsoft Word/pdf), provide insights related to the following topics/section to help the reader grasp the high level picture. Each section listed below will normally require one to two paragraphs of narrative. Feel free to write more whenever you see fit.

* Overview of the code (outline what the code is doing)
* Data Exploration Analysis/Clearning
* Explain the association rule used in the file (pretend your audience have never heard of association rule before)

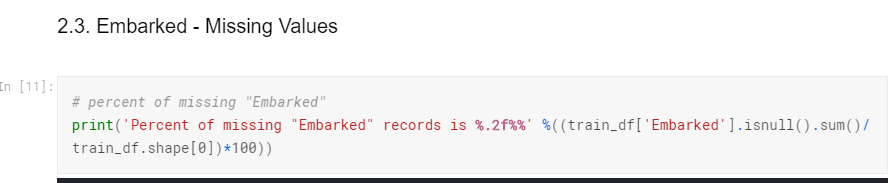
1. (10 points) In supplement to the above document, provide enough details to help the reader to understand the code within the Jupyter Notebook. You do NOT need to provide explanation to each single line of code, BUT enough details should be provided to the key code based on your understanding. Again, the ultimate goal of the document is to help a Python beginner, who has no or little previous python/coding experience, to understand clearly the logistic regression model code.

Your notation of the code can come in as either one of the following two formats:

* As comments using # inside the coding area, for example:



* Or as “Markdown” narrative outside coding area, for example:



Hint: there are 3-4 places where the code is already explained, then you do not need to add additional comments (you are welcome to add more but not required).

### Option 3: Web Scraping (40 points)

This option of project part c has two tasks. If you choose this option, you need to finish both options to receive credit (including the bonus credit)

**Option 3 Task 1 Instructions:**

1. Install wget on your machine
2. wget the [www.example.com](http://www.example.com) website; view the downloaded index.html file in a web browser
3. wget (without recursion) [www.utsa.edu](http://www.utsa.edu); view the downloaded index.html file in a web browser (if you get a robot.txt file instead of an index.html file, you were detected as a webscraper and blocked; just wget another site with some graphics and more interesting stuff than the example.com site)
4. Compare and contrast the quality/completeness of the two web pages you just scraped. Explain differences you see (i.e. why.)

Turn In\*:One file named: Wget\_YOURLASTNAME.ext, where ‘ext’ is one of the approved file types (.pdf, .doc, .docx). The file must contain screen prints of both rendered web pages (partial is fine for utsa.edu) and your written answer to #4 above.

**Option 3 Task 2 Instructions:**

1. Install the Python package “Requests: HTTP for humans” (<http://docs.python-requests.org/en/master/)>
2. Install the Python package “Beautiful Soup” (<https://www.crummy.com/software/BeautifulSoup/)>
3. Write a Python script that will scrape a Craig’s List web page for items for sale of a type you specify from a city you select, other than San Antonio. Here are the expectations:
   * You may hard-code the item type and city.
   * You may not knowingly pick the same city and item as another student.
   * You must scrape the date the item was posted to Craig’s list, its location, the full description of the item, and its price.
   * All items for sale on the page must be included in your output, even if some of the attributes are not provided (i.e. an item for sale does not include a location).
   * The script must write all scraped data in a nicely formatted CSV file, having a descriptive header row and properly delimited data/columns.

Turn In\*: (1) Your Python script, well commented, named Webscraper\_YOURLASTNAME.py. (2) Your output CSV file, named Output\_YOURLASTNAME.csv

1. This is to ensure that the Python code can be run without error, as the case in daily business world. [↑](#footnote-ref-1)
2. BTC: Bitcoin; ETH: Ethereum; LTC: Litecoin are three top ranked cryptocurrencies by market caps [↑](#footnote-ref-2)
3. The five csv files are downloaded using the above links and “Download Data” option. See picture below.  [↑](#footnote-ref-3)