Matt Sutherland

mwsuther@syr.edu

M.S in Applied Data Science

Syracuse University School of Information Studies

Project Portfolio Milestone

November 2022

Table of Contents

[Introduction 2](#_Toc120556879)

[Project # 1 – Process Improvement (Stock Trading) 2](#_Toc120556880)

[Project # 2 – 311 Non-Emergency Services (City of San Diego) 9](#_Toc120556881)

[Project # 3 – 2020 Presidential Election Twitter Sentiment Analysis 15](#_Toc120556882)

[Project # 4 - “Bridging the Gap” – Learning to Practice 20](#_Toc120556883)

[Project # 5 - My “CI/CD” Pipeline for - Continuous Improvement / Continuous Development 25](#_Toc120556884)

[Works Cited 27](#_Toc120556885)

# Introduction

A graduate of Syracuse University’s School of Information Studies (iSchool) Applied Data Science Master’s Degree - shall be able to use data for gaining insight into real world problems and act as a technical leading voice when solutioning.

The iSchool’s Data Science degree is an interdisciplinary program with a broad range of learning opportunities across a similarly broad range of data science disciplines. Students that complete the program will be able to:

* Collect, and store data by leveraging applicable technologies and an understanding of data structures and data properties.
* Create actionable insights often combining variables in unique ways.
* Apply visualizations and predictive models that support the discovery and execution of actionable insights.
* Use programming languages to collect, clean, modify, and perform simple and complex statistical calculations / modelling (i.e., Python and R).
* Communicate insights gained via visualizations and analytics to a broad spectrum of audiences (i.e., technical, and non-technical business stakeholders).
* Apply ethics in development and with the use of data and predictive models.
  + Make every effort to collect and apply data while reducing bias and promoting fairness, transparency, and privacy.

# Project # 1 – Process Improvement (Stock Trading)

*Class: MBC 638 Data Analysis and Decision Making*

*Date: August 2022*

*GitHub / Google Collab URL:* <https://github.com/sudz4/syracuse-data-science/blob/c5d096d5ca7f79572f16ae82a820601122427972/proj_1.ipynb>

|  |  |
| --- | --- |
| Project Name | Stock Investment Process Improvement |
| Project Overview | The goal of this project is to create a trading model that will enable me to buy up to 10 stocks on \***01JUN22** while keeping the initial investment to $25,000 or less, with the objective of selling the shares on **12SEP22\*** for the current market price. |
| Problem | The problem is that I do not have an effective trading model that will enable me to buy up to 10 stocks while keeping the initial investment to $25,000 or less.  *All while earning a profit to at least offset some tuition costs.* **(The ultimate objective, however, is ALL tuition costs covered for the current quarter).** \*M*odels are simulations; no actual trades were made to support this project.* |
| Tools & Techniques | The metrics that will be used to measure the success of the project are as follows:  -The number of stocks that can be purchased with an initial investment of $25,000 or less.  -The success rate of the trading model (the percentage of stocks that are sold for a profit).  -The amount of money made from the sale of the stocks and the potential longevity and scalability of the model. **A research team shall be assembled to work on the project.** |
| Insights Gained | Stock market data is abundant both for discrete and continuous data, text data sources.  Combining variables and stack ranking those variables from filtered conditions provides more valuable trading models / strategies. |

**Learning Objectives / Outcomes:**

**1. Data Collection**

Data Sources: IEX Cloud, Yahoo Finance

IEX Cloud has a paid API that provides data points that are used directly. Also, there are requirements where data points must be created by performing calculations on the data.

Logo

Description automatically generated with medium confidenceIcon

Description automatically generatedA picture containing text, sign, clipart, pointing

Description automatically generated

3rd Party Data Sources

A picture containing icon

Description automatically generatedA picture containing logo

Description automatically generated

**2. Actionable Insights**

Price to Earnings Ratio (P/E) ratio alone yielded a loss when uniquely applied.

Price to Earnings Ratio (P/E) as an aggregate percentile “high quality momentum” (HQM) score returned $5,873 from a $25,000 initial portfolio size.

|  |  |  |  |
| --- | --- | --- | --- |
| **Trading Strategy** | **Logo** | **$ Result** | **Outstanding bursar bill for final class at Syracuse = $5,406.00**  **{$ Result} >= $5,406?** |
| Robinhood Tech Bro | Logo  Description automatically generated | **+$1,650** | **FAIL** |
| P/E Ratio Guy | I made a few alternate sub Mascots/Logos and I hope you idiots like them :  r/wallstreetbets | **-$4752** | **FAIL** |
| Market Cap Gal | Cathie Wood: Founder and CEO of Ark Invest | #44 | Cointelegraph Top 100 | **+$24** | **FAIL** |
| Market Cap Gal’s Cousin | Here is Kim Jong WSB kid for all the memes this week : r/wallstreetbets | **+$331** | **FAIL** |
| Peregrine Falcon | A picture containing bird, bird of prey  Description automatically generated | **+$5873** | **SUCCESS!** |
| Otto the Data Scientist | Logo  Description automatically generated | **-$1800** | **FAIL** |

**3. Visualizations / Predictive Models**

Chart, line chart

Description automatically generated

(P/E Ratio)

The P/E ratio is a measure of the price paid for a share relative to the annual net income or profit earned by the firm per share. It is a market capitalization-weighted measure and is calculated as follows:

P/E Ratio = Price per share / Earnings per share (EPS)

The EPS used in the calculation can be reported EPS or diluted EPS.

P/E ratios are used by analysts and investors to determine the relative value of a company's shares in an absolute sense and relative to shares of other companies.

P/E ratios can be calculated for a single company or for a group of companies (index).

Instead of using one P/E ratio data point, an average of P/E ratios is calculated across time.

"""

High Quality Momentum (HQM)

-Create an array of time periods

-Create string for each time period

-Use string interpolation

-Calculate momentum percentiles

"""

time\_prds = [

'two\_year',

'one\_year',

'six\_month',

'three\_month’]

The best performing model is aptly named “The Peregrine Falcon”. The model is based on stock momentum. The **peregrine falcon** (and the cheetah) has the highest kinetic energy (aka “momentum”) in the animal kingdom.

Chart, line chart

Description automatically generatedA picture containing bird, bird of prey

Description automatically generated

The Peregrine Falcon uses an average of simple returns for the following stock ticker symbols.

['AJG', 'ALB', 'CEG', 'ENPH', 'GPC', 'MCK', 'NLSN', 'ON', 'PFG', 'PWR']

**4. Programming Language**

Python with the following libraries installed.

import pandas as pd

pd.set\_option('display.max\_rows', 200)

import datetime as dt

import matplotlib.pyplot as plt

import pandas\_datareader as web

import math

from statistics import mean

from scipy import stats

import json

import yfinance as yf

import requests

import matplotlib.pyplot as plt

import openai

Two important user defined functions are also created. And API keys are imported.

from m\_udfs import batcher

from m\_udfs import robot\_analyst

from m\_config import IEX\_API\_KEY

from m\_config import OPENAI\_API\_KEY

**5. Communication / Visualizations & Analytics**

The Python program automates excel formatting. This hides all the math and complexity and provides a document that can be shared with a trader (or someone that shall execute trades). The excel document output is an order sheet, with exact details on the number of shares to buy for selected ticker symbols based on the portfolio model.

A screenshot of a computer

Description automatically generated

**6. Development / Evaluation Ethics**

The data available for stock analysis is abundant. Making sense of the data is challenging because there are an infinite number of data points that can affect stock price. The key focus of this project is to learn and understand these data points. The Peregrine Falcon model might perform well for a given time frame, and we learned about the value of aggregating the same data point over different time periods. It is not clear whether the same model would perform well across time periods or in the future.

**7. Lessons Learned / Skills Developed**

Data frame cleaning, slicing, concatenating, merging etc. skills are tested as part of this project. To further add complexity, a real-time API is used for ingesting professional grade stock trading data. The API programming skills have been developed over years starting with simple projects around Twitter sentiment analysis during the 2020 Presidential election.

I learned an immense amount about stocks and the veiled mystique of algorithmic trading. I hesitate to dive deep into this area for a month or two. It took me 124 hours of development to complete the final programs (one for each trading strategy). There are 20+ other programs that never made it to the final six (6). Part of every programming session involved studying the stock market. I needed to develop requirements for the programs. I had to learn about stock signals and what to look for, which calculations to perform and strategies to test.

RSI Index\* vs Adjusted Close Price for ServiceNOW ($NOW)

Chart

Description automatically generated

What is an RSI index? These questions were constantly coming up and it was cumbersome navigating to google and invostopedia.com, stackoverflow.com, youtube.com, a kindle textbook, and ten other tabs.

In the summer 2022 I started automating work related (pre-sales engineer) tasks that now save me hours. With that same theme in mind, I created a virtual assistant to help me with stock terms.

# rsi = robot\_analyst()

# print(rsi)

"""

The relative strength index (RSI) is a momentum indicator that measures the magnitude of recent price changes to evaluate overbought or oversold conditions in the price of a stock or other asset. The RSI is displayed as an oscillator and the indicator value ranges from 0 to 100.

The RSI is calculated using the following formula:

RSI = 100 - (100 / (1 + RS))

where RS = Average Gain / Average Loss

The RSI is considered overbought when the indicator value is above 70 and oversold when the indicator value is below 30.

"""

The **robot\_analyst()** function enables me to stay in my programming flow and not navigate away from vs code. I run the function with the search term inserted and then copy the output into a markdown cell or just comment it out like above.

"""

Codex robot\_analyst() user defined function.

This function is used during development to help the programmer

learn about and understand technical stock terms.

"""

import openai

from m\_config import OPENAI\_API\_KEY

def robot\_analyst():

openai.api\_key = OPENAI\_API\_KEY

answer\_data = openai.Completion.create(

model="text-davinci-002",

prompt=input('Question: -> '),

temperature=0.3,

max\_tokens=150,

top\_p=1,

frequency\_penalty=0,

presence\_penalty=0

)

answer = answer\_data['choices']

for ans in answer:

final\_answer = ans['text']

print(final\_answer)

The API is available free up to a certain point. I became better and limiting pings on APIs from accessing stock data. I apply that skill similarly to the OpenAI endpoint I must hit as part of a function I wrote that hooks into the OpenAI model and outputs a description of a stock term etc. I can dial up the output by increasing the “max\_tokens” (an argument input to OpenAI’s create() function). For example, if I want more elaborate responses to the inputs.

**8. Further Development Plans?**

This project is geared towards an “Intelligent Automation (IA)” function. Many of course have heard of AI (Artificial Intelligence). The intelligent automation concept brings unique and actionable data to the end user, but the end user ultimately makes the decision and executes the trades.

When programming skills and stock market knowledge further develop, the objective shall be to automate trade execution. To expand on this more complex algo-trading type strategy the objective is to create an ETF (Exchange-Traded Fund) and execute trades automatically and dynamically for personal use and hopefully positive returns.

Further development shall add complexity to these buy/hold strategies. With simulated execution stocks are bought on 01JUN2022 and sold on 12SEP2022 for the project submitted as part of the final for MBC 638 – Data Analysis and Decision Making.

# Project # 2 – 311 Non-Emergency Services (City of San Diego)

*Class: IST 736 – Text Mining*

*Date: September 2021*

*GitHub / Google Collab URL:* <https://github.com/sudz4/syracuse-data-science/blob/c5d096d5ca7f79572f16ae82a820601122427972/proj_2.ipynb>

|  |  |
| --- | --- |
| Project Name | Digital City: 311 Non-Emergency Services Request Analysis (San Diego) |
| Project Overview | 1. Map 311 requests for the city of San Diego. 2. Perform data analysis. 3. Develop ideas for more efficient 311 request fulfillment. |
| Problem | Non-Emergency request processes are rarely explicitly addressed by State/Local Governments. San Diego and some other cities make a concerted effort to address 311 requests. Even when a process is implemented, there are dynamic components that can be made more efficient. |
| Tools & Techniques | City service groups should aim to serve their constituents as best they can and cities like San Diego are at the forefront of promoting digital services to reach this objective. The data collected from the Get It Done San Diego app is publicly available and therefore readily available to analyze.  There are or course many other avenues to understand more about the needs of the people of San Diego. Looking at multiple different sources of San Diego related data creates a more complete picture of what is important to the city and its constituents and visitors. Twitter is another great resource to combine with the 311 data. Twitter similarly opens its data to the world (upon approval of a developer account) and is an incredible resource for topic modelling and sentiment analysis. |
| Insights Gained | The map displays all the most recent 500 311 requests with icons. The icons when clicked display the name of the request type. In the example below, one of the more common requests is highlighted, “Graffiti Removal”. The filter in the top right of the map enables the viewer to quickly filter between requests based on how long the request has been open. The icons are also intuitively color coordinated with the green icons representing new cases and the dark red icons representing requests that have been open and unresolved for over 20 days. |

**Learning Objectives / Outcomes:**

1. **Data Collection**

Data Sources:

Graphical user interface, text, application

Description automatically generated

Digital services are becoming pervasive across the United States in cities like San Diego. One of the services San Diego offers to its constituents is the Get It Done San Diego mobile app. The mobile app enables citizens and visitors to submit non-emergency type requests, commonly referred to as 311-requests.

Graphical user interface, text, application, email, website

Description automatically generated

1. **Actionable Insights**

An image of a lion (San Diego Zoo theme) is used as a “mask” to deliver the word cloud output. The word cloud displays top words used in the tweets represented by size. The “mask” picks up the shape of the lion and pulls in the colors from the lion image.

A picture containing big cat, lion, mammal, brown

Description automatically generatedA picture containing text, newspaper

Description automatically generated

**3. Visualizations / Predictive Models**

Latitude and longitude data for requests can be analyzed and then plotted on a map. It is important to specify that latitude and longitude does not equal zero. Requests that are missing data are plotted at what cartographers affectionately have labeled “null island”. Null island is located at 0,0 and located in the Gulf of Guinea off the west coast of Africa. Requests with no location data will be filtered out of the data set.

A picture containing text

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

Toggleable layers are also added to filter the data based on request age. Request age is determined by how long a request has been open without having been closed / remediated. From a business use case perspective, the city does not want to have request fulfillment pending for months. The point of deploying a mobile application to intake requests is to respond to issues more quickly.

Background pattern, map

Description automatically generated

1. **Programming Language**

Python mainly using the **sckit-learn** library using topic modelling from **LatentDirichletAllocation**.

Python is also used to connect to the Twitter API.

**5. Communication / Visualizations & Analytics**

When compared with the other types of requests that were collected via the Get It Done San Diego app, Graffiti removal represents the third highest percentage of requests. The highest percentage of requests is represented as “Missed Collection”. It will be important for the city of San Diego to focus on these high-volume requests from a business efficiency and continual improvement perspective. Also, the requests indicated in the map view as “more than 20 days old” is an attribute that should be targeted for improvement.

Chart, bar chart

Description automatically generated

**6. Development / Evaluation Ethics**

In 2020 Twitter was less censored and it seemed easier to get real sentiment analysis data. There is not good available data to understand the demographic of people that submit 311 requests. The persona submitting 311 issues / requests is an important component that is not well understood. For example, do senior citizens not submit 311 requests via the app because of technology barriers? On the contrary, are more 311 requests submitted by senior citizens because they have more expendable time? The answers to these two-example hypothesis are not clear but are worth exploring if possible and as available data points evolve.

**7. Lessons Learned / Skills Developed**

Sentiment analysis and API skills were developed while creating this project.

I learned that all my lost data went to “null” island if my syntax was off or (with expected behavior) if location data is missing for the data entry.

Sentiment analysis of Tweets

Graphical user interface, text, application, email

Description automatically generated

Based on the nature of service requests, they generally tend to carry a negative sentiment label. Therefore, filtering tweets by negative sentiment and relevant topics allow for a targeted prediction of service requests.

**8. Further Development Plans?**

Future development will focus on proactively creating cases from Twitter and other related automation.

**San Diego related topics from Twitter**

Text

Description automatically generated

**Metadata in images (if submitted as an attachment to 311 service request)**

There is a step when submitting a 311 issue where the user usually submits a picture. The location of the picture can be pulled from image meta-data. This skips the location step and speeds up request submission while promoting automation.

For testing purposes, the image is of my Italian sportscar (fiat 500 turbo). In a real scenario, a citizen or constituent would take a picture of graffiti, an encampment, a damaged street sign, etc. and attach the picture to the 311-service request. To automate a step in this process the location data can be captured from photo metadata (or the users location shared via an iOS or android app.

|  |  |
| --- | --- |
| **Metadata** | **Image** |
| - Image width: 3024 pixels  - Image height: 4032 pixels  - Image orientation: Horizontal (normal)  - Bits/pixel: 24  - Pixel format: YCbCr  - Creation date: 2020-12-12 17:01:32  - Latitude: 32.79673888888889  - Longitude: -117.2315  - Altitude: 8.3 meters  - Camera focal: 1.8  - Camera exposure: 1/14  - Camera brightness: -2.47  - City: %G  - Camera model: iPhone 11  - Camera manufacturer: Apple  - Compression: JPEG (Baseline)  - ISO speed rating: 500  - EXIF version: 0231  - Date-time original: 2020-12-12 17:01:32  - Date-time digitized: 2020-12-12 17:01:32  - Shutter speed: 3.77  - Aperture: 1.7  - Exposure bias: 0.0156  - Focal length: 4.25  - Flashpix version: 0100  - Focal length in 35mm film: 26  - Producer: 14.2  - Comment: JPEG quality: 93% (approximate)  - Format version: JFIF 1.01  - MIME type: image/jpeg  - Endianness: Big endian | A black car parked on a street  Description automatically generated with low confidence |

# Project # 3 – 2020 Presidential Election Twitter Sentiment Analysis

*Class: IST 664 – Natural Language Processing*

*Date: 2020*

|  |  |
| --- | --- |
| Project Name | Presidential Election (2020) Sentiment Analysis (w/Twitter) |
| Project Overview | How are people feeling about the 2020 Presidential Election Candidates, Donald Trump, and Joe Biden?  The programs covered as part of this project focus on sentiment analysis. Specifically, the focus is around sentiment for tweets relating to the 2020 U.S. Presidential Election. |
| Problem | Information overload from the media. What are “the people” saying, which candidate do people like more in 280 characters or less? |
| Tools & Techniques | After connecting to the Twitter API, the first program tests to see if tweets containing a keyword or hashtag for “Joe” pulls tweets from the Twitter listener. Ten Tweets were queried to start while keeping in mind that there are limits to the number of times a user with a Twitter developer account can ping the Twitter API. |
| Insights Gained | Natural Language Processing (NLP) is broken up into two parts. The first part is natural language like English, French, and Chinese. The last letter in the NLP acronym, processing, refers to how a computer will process that natural language with NLP techniques such as sentiment analysis, email spam detection, topic modeling, text generation, and more. |

**Learning Objectives / Outcomes:**

1. **Data Collection**

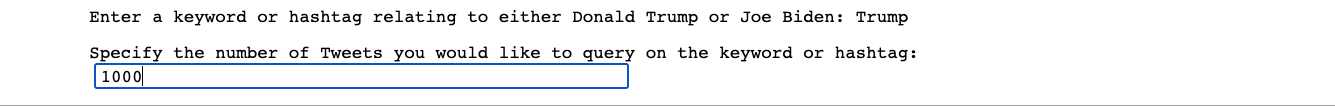
Data Sources:

Twitter can be a great way to see how people are thinking and feeling about a subject. We want to create a repeatable process to check real time sentiment for either Donald Trump or Joe Biden as represented by their most recent mentions in Tweets. The program below enables the user to query for a keyword and then specify the number of tweets to query. The results are then serviced up in a visualization that is easy to interpret and understand.

It is easy to see how the program works. Enter a simple input and then specify the number of tweets to query. The API defaults to most recent tweets but there is an argument to pass that specifies a start and end date.

A screenshot of a cell phone

Description automatically generated



Pardon the departure from formal syntax. But it was cool working with Twitter data during this time-period. I felt like I could easily get a good pulse on the volatile poll swings and uncertainty going into the 2020 elections. Not that I cared much at the time or was politically invested either. I was mostly invested with looking at the outputs from my new Twitter programs.

1. **Actionable Insights**

Sentiment analysis for mentions of Joe Biden on Twitter (Most Recent 1000 Tweets \*2020).

A screenshot of a cell phone

Description automatically generated

1. **Visualizations / Predictive Models**

The sentiment from the Twitter population as it relates to the two presidential candidates can now be reproduced on command. Next, we want to look at Trump and Biden’s tweet sentiment. A limitation to note from the Twitter API is that extracting tweets is limited to 200 at a time. In this case and to remain “real time” and up to date, this program queries the most recent 200 Tweets from the candidates’ respective Twitter handles.

A screenshot of a social media post

Description automatically generated

1. **Programming Language**

I have connected to the Twitter API with R in 2019-2020 but this project was all programmed in python.

1. **Communication / Visualizations & Analytics**

Joe Biden’s tweets are put into a data frame for easier parsing an analysis. The issue right now is that symbols like ‘@’ and ‘RT’ are going to skew some of the data analysis. We need to apply some regular expressions to remove these symbols.

A screenshot of a social media post

Description automatically generated

1. **Development / Evaluation Ethics**

I was dissuaded from exploring the API more as some of the censorship ensued. It felt to me like the source data was corrupted. The great thing about the Twitter API is that it was built by developers and with the developer in mind. The data points are easily accessible.

Last thing on corrupt source data. I became very comfortable with APIs because of learning how to interact with the Twitter API. Again, departure from formal tone, it was extremely cool that I could query text from programs and feel like I bypass any echo-chamber or algorithm that my username might be (unintentionally) wrapped up in.

I don’t have any conspiracy theories on this one. It makes sense to me that my username could get caught in an echo-chamber, often exactly as the user would hope. As an example, my feed has many python, and data science people, some people I followed years ago when I was really into R. As the user I want those feeds coming to me so I can discover a new python related twitter account to follow, or a new sports reporter that follows my favorite team, etc.

In sum, as a python programmer with limited but some understanding of how to read API documentation (Twitter has great documentation), it’s easy to get creative with precise data collection.

1. **Lessons Learned / Skills Developed**

Regular expressions are used to filter out bi-grams or just by matching strings. This skill is adapted here to make the data output more accurate. For example, a viral tweet from either candidate is often “re-tweeted” with the annotation automatically added “RT”. Using regular expressions and 10-20 iterations of trial and error a solution presented itself.

It is good to know that it’s possible to filter the “RT”. I had to question now how I felt about a “re-tweet”. In other words, is a “RT” less valuable than a unique tweet? A lot of people do a “RT” and add their own message. Others just “RT” with no comments. With the robust Twitter API endpoints, it’s possible to slice the data output a lot of ways.

1. **Further Development Plans?**

I want to say I would dive back into developing from the Twitter API, but it is less interesting for me now.

I may however, incorporate Twitter sentiment analysis as a variable component in stock portfolio trading models. I’d use sentiment data from (twitter.com/api) and (stocktwits.com/api) and take an average (using percentile scores vs other stocks). Then combine that variable with really any other combination of variables or sets of variables. Almost lastly, I’d back test those models on historical data and determine whether anything interesting surfaces. And finally, test, retest, test again anything that might seem lucrative before quitting day job.

# Project # 4 - “Bridging the Gap” – Learning to Practice

*Class: N/A with influence from MBC 638 – Data Analysis and Decision Making*

*Date: September 2022*

*GitHub / Google Collab URL:* <https://github.com/sudz4/syracuse-data-science/blob/6c13adbd2a0f9ecde87f6bae579ab786479ffdac/proj_4.ipynb>

|  |  |
| --- | --- |
| Project Name | N/A – Programs developed while researching potential process improvement final project ideas. |
| ~~Project~~ (Research) Overview | Business use case at work, justifying headcount for pre-sales engineers to promote increased cycle time and work output. |
| Problem | Business stakeholders / leadership require business justification(s) to increase headcount. |
| Tools & Techniques | Queuing Theory math - I could not find a function anywhere in “open-source land”, so I wrote the math out literally as code.   * Then I created my own functions to interact with it. |
| Insights Gained | First principles. |

**Learning Objectives / Outcomes:**

1. **Data Collection**

Queuing theory is applied to several hypothesis related to improving cycle time. A function is used to gain an understanding of a stochastic work intake queuing process.

**Agner Krarup Erlang**

The concept of an expanded view of queuing theory (as a modification of “Little’s Law”, and the coining of the modern term is attributed to Agner Krarup Erlang (1 January 1878 – 3 February 1929). Agner was a Danish mathematician, statistician, and engineer, who invented the fields of traffic engineering and queueing theory.

1. **Actionable Insights**

Create a function that simulates (mostly) any pre-sales task queueing scenario.

* Indicator to measure and improve = "Average Cycle Time" (CT)

tq: waiting time in the queue (represented as days)

tp: effective processing time (also represented as days)

CVa: is a deterministic distribution of arrivals

CVp: is a stochastic distribution or processing time

**CT = tq + tp**

**Assumptions:**

* The scenario assumes that SOWs or similar tasks are task completion by batch.
* SOWs or similar tasks are requested in a continuous flow with a constant interval time.
* A constant flow of tasks may be realistic.
* Focus process improvement in other areas such as windows to intake or view SOW requests.
* Use an SLA timer starting when the SOW state is moved from "Received -> In Progress".
* Processing time also as a function of requirements ascertained from the Client.
* Discovery calls increase time LOE in the short term but decrease LOE when developing a
* solution for a SOW, DEMO, etc.

In a deterministic scenario (CVp = CVa = 0) because there is no queuing time is zero (0)

for both solutions.

But we bring variability in with count and rate of arrivals, rate of processing and time between

starting new processing, quality issues, errors, etc.

1. **Visualizations / Predictive Models**

**Pre-Sales Engineering Relative Work Queues**

**Diagram

Description automatically generated**

**Deterministic distribution of arrivals**

Text

Description automatically generated

**>>> Output**

Graphical user interface, text, application

Description automatically generated

1. **Programming Language**

Python

1. **Communication / Visualizations & Analytics**

A picture containing chart

Description automatically generated

1. **Development / Evaluation Ethics**

The basic evaluation of the model makes assumptions about a Pre-Sales Engineers skill level. Further development would focus on a scoring system of Pre-Sales Engineers and their capability within technical domains.

1. **Lessons Learned / Skills Developed**

A first principle is a basic assumption that cannot be deduced any further (Clear, 2022). Aristotle defined a first principle as “the first basis from which a thing is known” (Clear, 2022).

The common basic queuing system attributed to Erlang is a modification of Little’s Law (Hernandez-Suarez, 2010). I read extensively about Little’s Law and Erlang’s work on queuing theory.

By attributing first principles I first learned about the math behind queuing theory before programming the math in python. A welcome challenge that I faced was that I could not find an existing math related library for solving my queuing theory problem. Therefore, I learned the math and just programmed the math explicitly in python.

**Text

Description automatically generated**

There is an important lesson here that I now carry with me. The “First Principles” approach to problem solving.

1. **Further Development Plans?**

Queuing theory can be applied to logistics and operations and network and system design. I plan to do further and more advanced development by applying queuing theory to system design theory with computers and computer networks.

I feel like we live in the shadow of math giants like Erlang and Little and I am humbled in understanding that the applications of their work to new or future problems is seemingly endless.

# Project # 5 - My “CI/CD” Pipeline for - Continuous Improvement / Continuous Development

*Class: N/A*

*Date: Fall 2019 - Present*

|  |  |
| --- | --- |
| Project Name | Statement of Work (SOW) Automation |
| ~~Project~~ (Research) Overview | Pre-Sales Engineers have many tasks. Create an automated system for scoping enterprise software deployments. |
| Problem | SOWs take 3-5 days, maybe even two (2) weeks to complete depending on task load and technical skill competency. |
| Tools & Techniques | Use Python and Excel to automate the SOW creation and continuous development / continuous improvement. |
| Insights Gained | Many processes can be improved with Python automation. |

**Learning Objectives / Outcomes:**

**Data Collection**

Created a temporary database in a tabbed excel file.

* Database built manually by aggregating Client requirements over time and developing on the ServiceNow platform.

**Programming Language**

Python and Visual Basic (VBA)

**Communication / Visualizations & Analytics**

Chart, bar chart

Description automatically generated

Chart, bar chart

Description automatically generated

**Development / Evaluation Ethics**

This project requires the technical understanding of Python, Excel, and ServiceNow. I wish that I could have delegated more of the database buildout component of this project to subordinates or others. The project outcome had been challenging to communicate. I expect and hope that with a completed project framework and proof of concept (POC) that I can get help with building out the database.

**Lessons Learned / Skills Developed**

Data manipulation skills were developed while working on this project.

**Further Development Plans?**

I am going to keep working on SOW automation and other work-related automation projects.

# Works Cited

Clear, J. (2022). *First Principles: Elon Musk on the Power of Thinking for Yourself.* Retrieved from Mental Moedls: https://jamesclear.com/first-principles#:~:text=A%20first%20principle%20is%20a,Scientists%20don't%20assume%20anything.

Hernandez-Suarez, C. (2010). *An application of queuing theory to SIS ans SEIS epidemic modesls.*