Table of Contents

Award Lookup

Date Start/End	Title	Objective
11/08/2022 12/05/2022	Fix incomplete related project families and redefine family groupings	Recreate data with redefined family relationships

Issue: The raw data of award numbers consists of an incomplete record of related child and parent awards. Due to this, the related families are incorrect within the grant search system.

Approach: 1) Import ICER data; 2) Create calculated columns that search the dataset for an award number based on set rules to create a new list of children and parents; 3) Combine these 2 rule-based columns to form a calculated "Related" column.

Outcome:

An Excel Document was created to calculate the columns based on search rules. A pair of check columns were created to find new child/parent matchings in the data, and a pair of final columns were created to combine the pre-existing and calculated search matchings. With this, the co-parent is also extracted from the parents list. From the 2 final columns, a new "Related" column is created that combines them to form an immediate family. More levels of the family structure are found by continuing this search matching with the family columns: "R2", "R3", "R4", and "R5." Columns R4 and R5 are the same because there are no new levels found after R4.
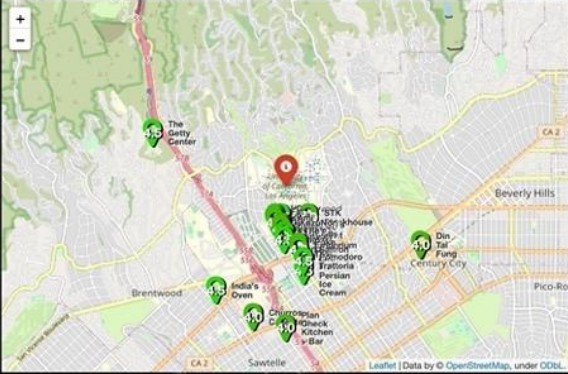
BI Search Tool

Working on getting permission to showcase this project.

Search, Review, and Explore Restaurants (Website)

I was touring Greece and Rome and found that it was difficult for me to use Yelp to find nice places to eat. In frustration, I decided to create my own website to take advantage of reviews online as well as other data and piece it together to find the most suitable place to eat based on my query. I expanded this concept by implement NLP models, sentiment analysis, and adjective visualization to improve the user interaction.

BBC News NLP Report

Using a few models and different Python libraries, I took a look into a folder full of .txt files. In this report, I get to answer a few questions, analyze country name dropping, look at sentiment and polarity, classify articles into topics, and more.

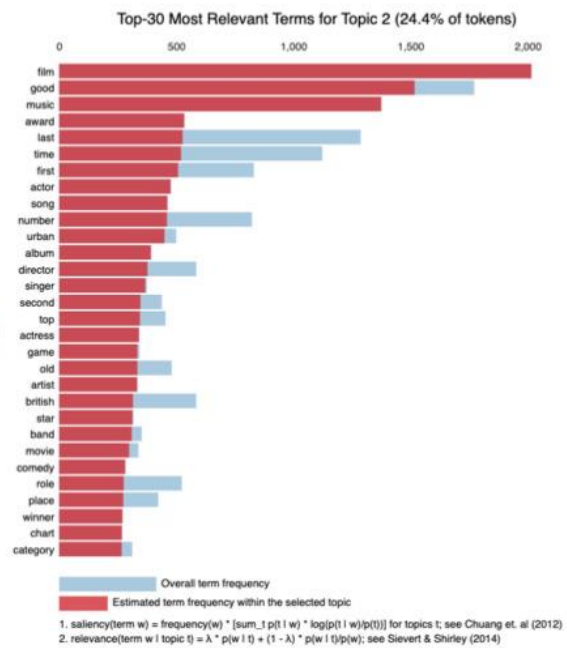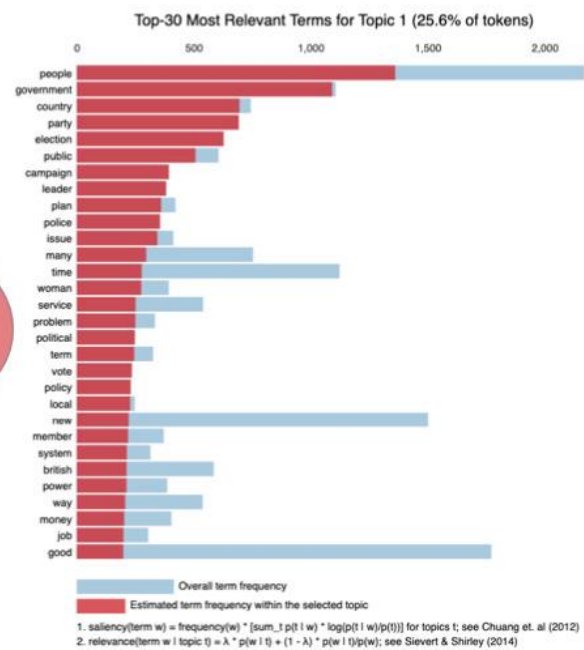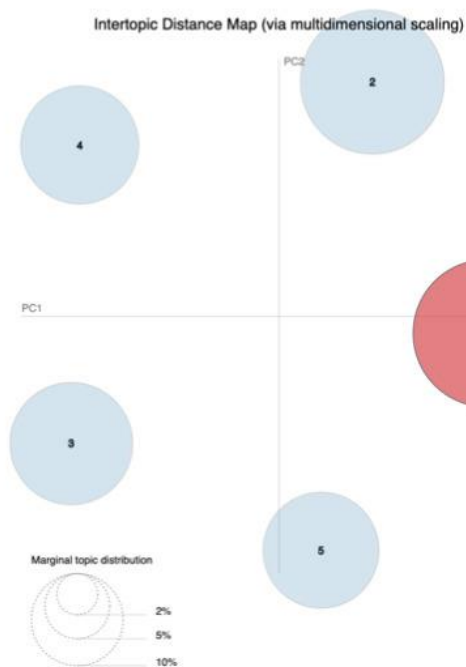1. *Describe the methodologies you used in your report.*
   To answer the question of what is the most mentioned G20 country, I thought the simplest route of retrieving the common topics of the articles was to get the topics directly from the articles' words. So I used Counter to count the most common word sequences per article category after cleaning the text and then plotted these phrase frequencies. After thinking how this may be flawed because it risks skewing the common phrases towards articles that are repetitive, I then added an interim step to make every word unique per article, so there are no repeats. This new step solves the issue, so I will note and accept the drawback of some two-letter phrases not being counted in some rare circumstances. There is likely another route to count uniquely without taking this loss, so given more time, I would look into that. For a count of repeated word usage, I counted the number of times specific words were found in the text, and for every individual article with a match, I would tally it. I plotted these counts for a polished final result.

2. *What tools did you use for this analysis?*
   I used Python on Google Colab to test code quickly and look at the data available, and then I used Visual Studio Code to run the Python code locally and create a PDF report.

3. *If you had more time, how would you strengthen your report?*
   If I had more time, I would integrate the geopolitical entity detection and geocoding I set up for fun with the simple country name counter to create a more accurate list of most mentioned countries out of all countries globally using the Google Maps API. I would also apply more topic models to check if they may have greater coherence. I would also like to stress test the script's speed by adding more news articles and articles in different topics to the dataset.

Intertopic Distance Map (via multidimensional scaling)

PC2

4

2

PC1

1

3

5

Marginal topic distribution

2%

5%

10%

Top-30 Most Relevant Terms for Topic 1 (25.6% of tokens)

people
government
country
party
election
public
campaign
leader
plan
police
issue
many
time
woman
service
problem
political
term
vote
policy
local
new
member
system
british
power
way
money
job
good

Overall term frequency
Estimated term frequency within the selected topic

1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Intertopic Distance Map (via multidimensional scaling)

PC2

2

4

PC1

1

3

5

Marginal topic distribution

2%

5%

10%

Top-30 Most Relevant Terms for Topic 2 (24.4% of tokens)

film
good
music
award
last
time
first
actor
song
number
urban
album
director
singer
second
top
actress
game
old
artist
british
star
band
movie
comedy
role
place
winner
chart
category

Overall term frequency
Estimated term frequency within the selected topic

1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

SQL Challenge (LA Clippers Final Interview)

Before my L.A. Clippers final interview, I was given the challenge assignment of analyzing seating, tickets, and attendance data to answer some business queries and then package my solutions into a business presentation. After completing my presentation, I found many points to improve my skills on, and I know I gained a lot from the opportunity.

Esports EDA Assessment

Determine if this dataset needs any preprocessing. If so, clean the dataset and document your steps. If not, explain how you came to that conclusion.

Multicollinearity has a negative impact on many popular ML models. Check if this dataset experiences any multicollinearity. If so, reduce the impact until an acceptable point. The data will definitely require some preprocessing due to the presence of null values, indicated by "?"s. I will also convert column types, remove outliers, and remove high VIF columns.

Code below removes '?' values but that has consequence of removing League 8␣
,→data, which we need what few data points we have # Code to split data between complete data and incomplete row values
Remove recorded playtime when hours per week > average hours awake, given 8␣
,→hours of sleep/day df =df[df['HoursPerWeek']
Remove outliers #Median Absolute Deviation
Drop high Variance Inflation Factor variables

3 Determine what are the most important features that could help predict a player's rank?

The features most strongly correlated to LeagueIndex are APM, ActionLatency, and GapBetweenPACs. Correlation is important because it measures the linear relationship between two factors. For instance, a higher APM and GapBetweenPACs is associated with a higher LeagueIndex. Conversely, a lower latency is associated with a higher league. TotalHours and SelectByHotkeys will also be shown as important features in the demo model I built. These results are natural because these features are the building blocks of what enables a player to improve: low latency and total hours. If latency is higher, a player will not be able to showcase all their skills and will also have lower action speed. A minimum total hours is also necessary to practice enough and beat the learning curve. A natural indicator of skill, albeit biased. Meanwhile, players in higher leagues tend to be more skilled, i.e. faster actions (APM, Hotkeys). A more nuanced measure of skill (strategic/calculative skills) is not provided in the dataset. A skill indicator like win rate would be valuable, but players are matched against those in the same skill bracket. Because of this, the win rate will approach 50% when there is 1 winner and 1 loser per game. Therefore, action speed is the most prominent differentiator. Raw speed is not normally an effective metric for competitive games, but StarCraft centers around multi-tasking, so performance will be stunted with fewer actions, regardless of strategy or skill.

Your team's Starcraft2 coaching staff loved your project! They think this is perfect for scouting rising stars. Using your discoveries from (3), create a function to find players who should be given a chance to become professionals. Explain why your set of players make sense.

> By looking at the feature distribution grouped by each LeagueIndex, I saw a statistical difference between those in the professional league versus those in lower leagues. By taking the core stats of Professional League players, I believe I can filter the players and create a list of players with high potential. League 7 players have an average playtime of 31 hours/week (10 hours more than League 6 players and almost double League 5 players), which represents the time commitment required. There is also a baseline of at least 1000 total hours of playtime, which is flexible but serves as a foundation for players' skills. I believe there is further error in the total hours played input because higher league players may create second accounts after playing for a while. Continuing by analyzing the minimum, first quartile, third quartile, and maximum values, I decided to filter as shown below: APM >= 250, GapBetweenPACs <= 23, ActionLatency <= 40 For a smaller list, also filter by HoursPerWeek >= 25, TotalHours >= 1000.

Hypothetically, if you were to move forward with creating a fully-fledged model to predict LeagueIndex, what model(s) would you consider and why? (Don't actually implement anything!)

> Linear regression and SVM may have trouble due to the hyper-overlapping of the players between leagues in terms of the other features in the dataset, as seen in the plotted pair plot. The regression would not accurately place a line through the scatter plot while differentating between leagues. KNN, Decision Tree, and Random Forest models are popular picks because they are based on robust algorithms that systematically succeed, so I would probably start with them to pick out the league for the similar-looking players. I did some preliminary research into trying to model the data already for Question 3, and I believe regression makes more sense in terms of accuracy and output than classification (although it depends on the goal).

Socials Dashboard (Google Data Studio)

I was elected to be the Data Analytics Team Lead during my WebSniffer internship. I provided guidance to my team members while communicating with upper management. I helped my team by offering technical expertise and advice on data analysis, and I also worked with the team to identify and resolve any obstacles that arose. One of the key methods I helped the team achieve its goals was by providing clear communication: I ensured that everyone on the team understood their roles and responsibilities by working closely with the team to ensure that everyone had the resources they needed to be successful. Overall, my experience as a team lead taught me the value of collaboration and problem-solving in achieving team goals; I believe that these skills are essential for success in any team environment, and I am confident I can contribute these pillars to any future team I am accepted into.
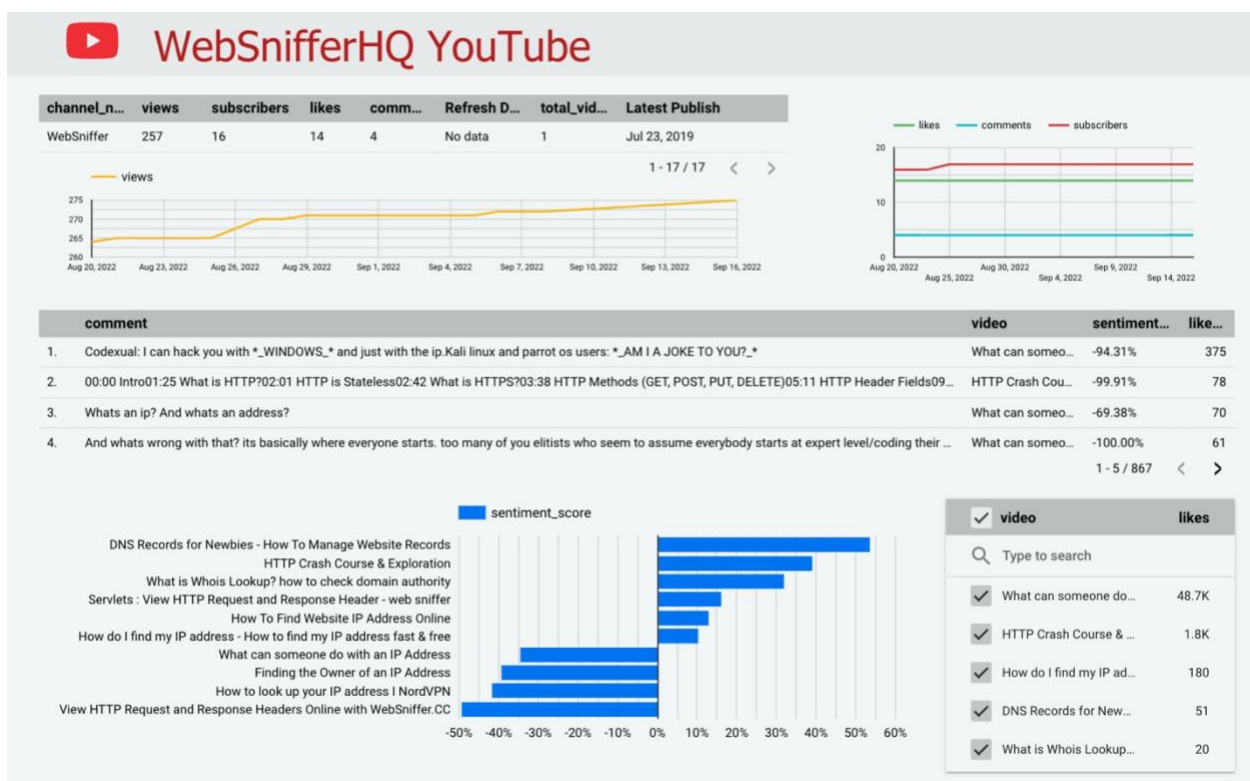
Receipt OCR Website

Receipts are a transcript of all of our expenses, but many people choose to throw them anyway. I wanted to repurpose this nuisance of paper into useful statistical measurement of cost management. By creating a simple app to read receipts and transform their numbers into a report, I accomplished my initiative to add utility to something that has few immediate benefits.
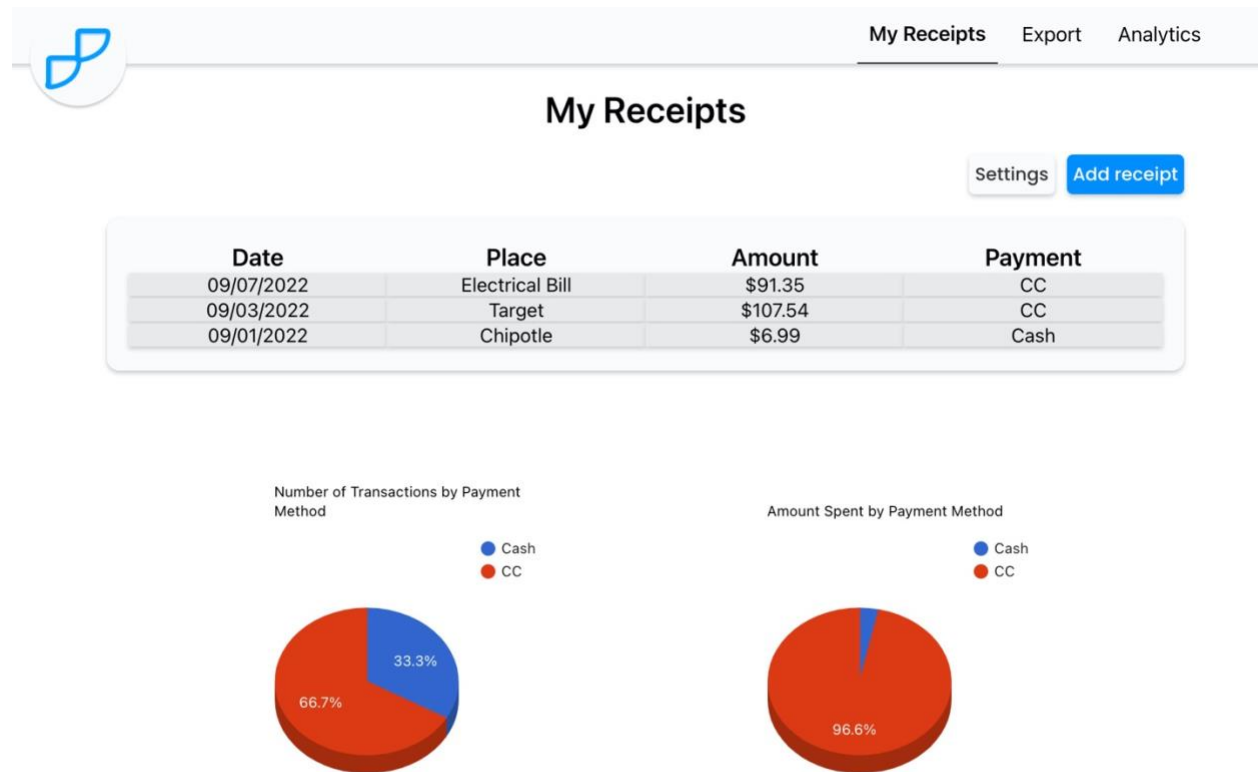
Loan Predictor App

I created a simple framework to host my machine learning classification model. I thought that the finished product was a interesting display of the capabilities of machine learning, so I am happy to have created it.

NFT Market Analysis

While researching cryptocurrencies during my internship with Fayble Inc., I decided to create a dashboard on my own time to capture the state of the market. We had weekly presentations, and I thought this dashboard would serve well in tying my conclusions together, especially as a showcase of the market to my clients in the audience.

For instance, I found that the recent downward surge in prices during the end of summer 2022 was propelled by unfavorable macro structures: liquidation, derivatives, and loan markets. Most investors were liquidating their holdings, so there was less cash pumping the market and the prominent action became selling.
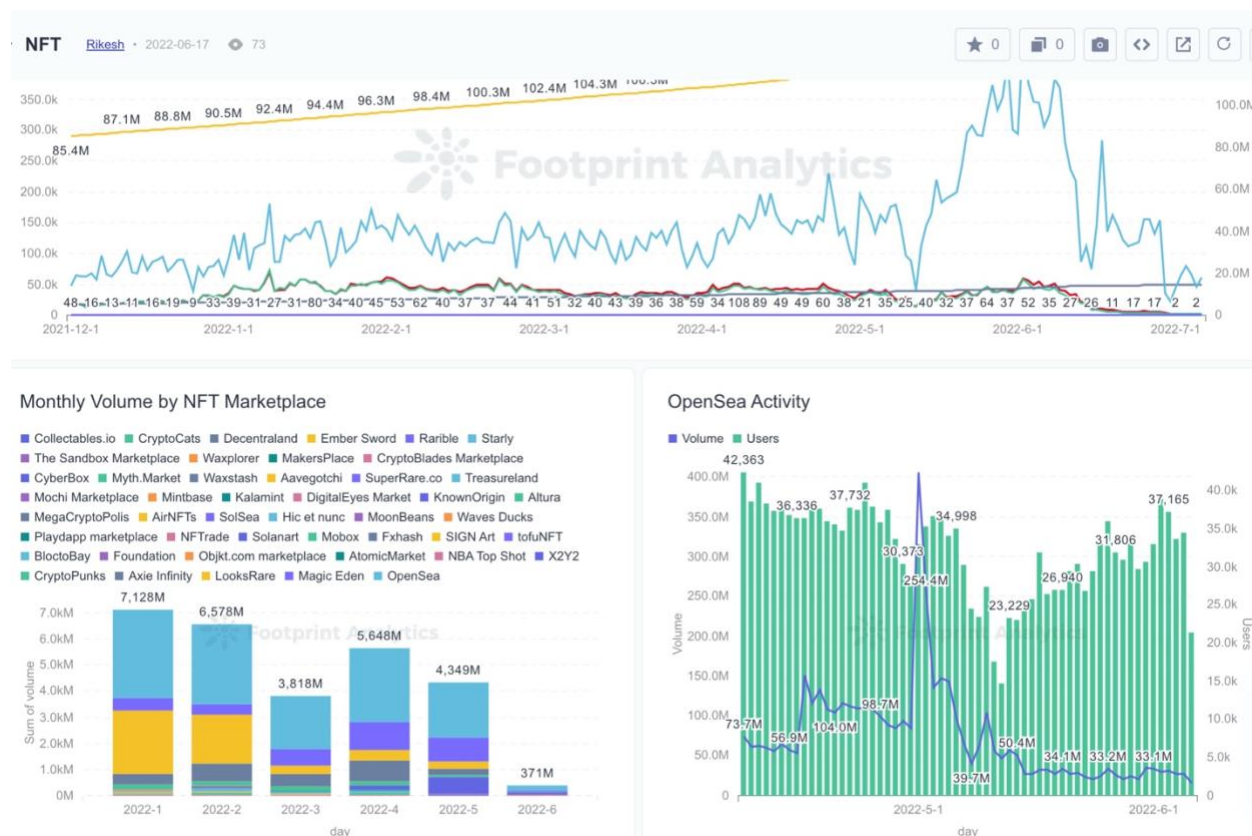
Portfolio Website

When I was creating my website initially, I was very excited because it was my first project, and I had been waiting for the opportunity to make a personal website for a while.

To do this, I used HTML, CSS, and Javascript to set up everything. One of the most difficult features to implement were the rotating backgrounds upon page refresh, but I persevered to get it to work, adamant on publishing the website only after I figured out how to set up this ostensibly trivial feature. It took much patience, but I finally found a solution. Then, I added mobile support for different screen resolutions and contact screen page to accompany my resume.
Login authentication security

Stock Analyzer

I wanted to play around with real world financial data and try my hand at predicting the future of the stock market. This is that.

Food Image Recognition

I needed to track my caloric intake and nutritional gain for my Nutrition class, and I thought of the perfect solution: Neural networks! I used a set of 100 common food images to train my model and built a Streamlit app to be its framework.