My pronouns: she/her/hers.

Self-Intro

My name is Weishan. I gained my master's degree in Business Analytics last year from UC Davi. After graduation, I joined Andela as a Data Scientist. I worked closely with the product team to lead all the experimentation efforts. I leveraged A/B testing to optimize user workflow and machine learning algorithms. I am also one of the key members who built the recommendation system, which is used to recommend suitable talents given a bunch of job information. My analysis helped the team to increase the engagement of the system by 30%.

Prior to Andela, I was a quantitative researcher at Ipsos, a marketing consulting firm. There I used statistical analysis to help client companies understand customer behaviors, track brand health, and promote concepts to products. I also took a senior business analyst position at Mininglamp, another marketing consulting firm. There I focused on evaluating marketing performance using unstructured social media data.

Experiment - Use LaunchDarkly to Run A/B Testing

Andela did not conduct A/B testing before the data science team came. The PMs are not familiar with the procedure of A/B testing. The limited traffic on the website is also another challenge.

The original idea is to develop an internal experimentation platform but we are short of hands to make it happen. Instead, we want to turn to third-party software, and educate the team to nurture the culture of experimentation first.

I am responsible for the design and analysis of A/B testing. I will communicate with the PM about metrics, estimate the length of the experiment, determine the randomize unit, and create a short report to interpret the result and draw conclusions when the tests ended.

Challenges1: Bayesian vs Frequentist
One thing that is special about LaunchDarkly is that it uses Bayesian approaches instead of Frequentist, which is new to me. I used my spare time to learn about the differences between these two methodologies and went through the official doc to understand this approach.

Bayesian mainly differs from the frequentist approach in two aspects. One is performance in small sample size. It is hard to observe significant difference with small sample size, which is the situation we need to handle at Andela. It is also common when we want to run test within a certain niche group. At the early stage of building the product, we don't want to hinder the launch of new features, and we also don't want to take random risk. Bayesian approach will serve as a good tool to give us an idea about how much risk we are taking when making the launch decisions. By keep tracking of all the tests that we ran, at the end of the year, we will be able to see the whole risk portfolio and whether the changes lead to a better product.

The other difference between Bayesian and Frequentist approach is the interpretation of statistics. There is no statistical power, p-value, or statistical significance in Bayesian. The output of Bayesian approach is the probability to be the best, relative difference from control, credible interval, and posterior mean.

- **probability to be the best**: Probability to be best is the likelihood that a variation had the biggest effect on the primary metric. Of all the variations in the experiment, the one with highest probability is the best option to choose as a winner.
- **relative difference from control**: The relative difference from the control variation is the difference between the mean of the control variation, and the upper and lower bounds of the credible interval of the variation you're testing. This range contains 90% of the variation's probable values. For example, imagine you have a control variation with a mean of 1%, and the variation you're testing has a lower credible interval of 1.1% and an upper credible interval of 1.5%. The difference between 1 and 1.1 is 10%, and the difference between 1 and 1.5 is 50%, so the treatment's relative difference from control is 10% to 50%.
- **credible interval**: The credible interval is the range that contains 90% of probable values.
- **posterior mean**: The posterior mean is the average value of the posterior distribution of the variation. If you were to continue collecting data, the posterior mean is the future average numeric value for the variation that you should expect. It doesn't capture the uncertainty in the measurement, so it should not be the only measurement you use to make decisions.

Sources: https://docs.launchdarkly.com/home/analyzing-experiments/results

Challenges2: individual level data for further analysis
Read the official doc to figure out the process of building the data pipeline from receiving data on Google Cloud Pub/Sub

Client Portal Remove One of the Sign-up Page - Limited Traffic

Situation:
Our first A/B testing was launched on the client portal, we want to see whether removing one of the landing pages of the signing-up process can increase the sign-up rate.

Task:
We paid for a third-party software service called LaunchDarkly, which allows quick set up for A/B testing. It follows the Bayesian approach to analyze the experiment result. We coordinated with the engineering team to guide them to set up the test, worked with the product team to design the experiment, and finally analyzed the result and made recommendations after the test ended. The biggest challenge that we faced in this test is the limited traffic. On average, there are around 1500 traffic per month on the website.

Action:
- **Metrics**: Given the low traffic, we advised the team to focus on metrics on the top. After communicating with the product manager, we determined to use sign-up rate as the primary metric.
- **Randomized Unit**: Given the primary metric sign-up rate, in this test, we only focus on new users.
- **Prior Distribution**: Since sign-up rate follows Bernoulli distribution (a client either finished signing up or not), we choose beta distribution as our prior distribution.
- **Sample Size / Length of the test**: One advantage of Bayesian A/B testing is that there is no sample size limit on Bayesian A/B testing. We usually advise the team to run for at least two weeks to capture the weekly patterns. At the end of the first week, we collected around 500 samples. However, I found that the credible intervals of the sign-up rate are too wide ranging from negative to positive, making it not informative to make conclusions. I convinced the team to extend the test for another week. I observed that the credible intervals narrowed down and gradually stabilized as more traffic came in. So I decided to terminate the test.
- **Decisions**: And started to analyze the results. The probability to be the best (it was generated using Monte Carlo Simulation) and credible interval. When the probability to be the best for the treatment group exceeds 90%, we claim the treatment to be the winner. (90% is the threshold that we set ourselves. It can be adjusted according to how much risk that we would like to take.) Another metric is the 90% credible intervals which contain the center 90% of the posterior distribution. The lower end of the credible is around 1%, indicating a small effect size the new change can bring.

Result
We eventually convinced the product team to hold back the change. The smooth testing process also encourages other teams to conduct A/B testings.

Once upon a time, we embarked on an exciting journey to conduct our very first A/B test on the client portal. Our mission was to determine whether removing one of the landing pages from the signing-up process could significantly increase the sign-up rate. With eager anticipation, we delved into the task at hand.

To facilitate our experiment, we invested in a third-party software service called LaunchDarkly. This invaluable tool enabled us to set up our A/B test swiftly and followed the Bayesian approach to analyze the experiment's results. Collaborating closely with the engineering team, we provided guidance on setting up the test infrastructure. Working hand in hand with the product team, we meticulously designed the experiment, ensuring a well-defined testing process. And finally, once the test concluded, it was our responsibility to analyze the results and provide insightful recommendations.

Our primary challenge in this test was the limited traffic on the website, with an average of around 1500 visitors per month. Given this constraint, we advised the team to focus on the top-level metrics to glean meaningful insights. After fruitful discussions with the product manager, we determined that the sign-up rate would serve as our primary metric of interest. Focusing solely on new users, we employed the Bernoulli distribution to model the sign-up outcome and chose the beta distribution as our prior distribution.

In terms of the sample size and test duration, Bayesian A/B testing offered us the flexibility to continue gathering data without strict sample size requirements. However, we advised the team to run the test for at least two weeks to capture any potential weekly patterns. At the end of the first week, we had collected approximately 500 samples. However, upon analyzing the results, we noticed that the credible intervals for the sign-up rate were uninformative, spanning from negative to positive values. Recognizing the need for more data, we successfully convinced the team to extend the test for another week. As more traffic poured in, we observed the credible intervals gradually narrowing down and stabilizing, leading us to make the decision to terminate the test.

With the test completed, we delved into analyzing the results. Through the power of Monte Carlo Simulation, we calculated the probability of the treatment group being the best and examined the 90% credible intervals. Our predefined threshold for declaring the treatment as the winner was a probability exceeding 90%. In this case, the probability of the treatment group outperforming the control group fell within the expected range. Additionally, the 90% credible intervals indicated a small effect size, with the lower end hovering around 1%.

Armed with these findings, we successfully convinced the product team to hold back implementing the proposed change. The smooth and insightful testing process also fostered a newfound enthusiasm for A/B testing within our organization, encouraging other teams to embark on their own experimentation journeys.

When running A/B testing on to-B platform, the biggest challenge is short of traffic. We came up several ways to handle this situation.

1. **Bayesian Approach**: The first is to use Bayesian approach to get the probability of treatment and control instead of running tests with the frequentist approach and never reach a significant result.
2. **Top Metrics**: The second one is to guide the product team to focus on metrics sitting on the top of the funnel such as website visits, and sign-up conversion rate. When they came down to metrics like the number of clients posting the first job post, we will give them warning that the sample size will be too small to be informative.
3. **Test Big Changes**
4. **Emphasize the worst scenario**: it is the lower end of the credible interval

Other ways:
User research, heuristic analysis, lower statistical power, reduce variances, boostrap

Sources:
https://www.portent.com/blog/cro/how-you-can-run-a-b-tests-on-low-traffic-sites.htm

Talent Portal Remove Resume Upload Page - Early Stop

Situation:
We conducted an A/B testing on the talent portal, we want to see whether removing the uploading resume step can increase the talent sign-up rate. It is practical because the resume will not be needed until the talent successfully passes our technical test.

Task:
To facilitate the experiment, we invested in a third-party software service called LaunchDarkly, renowned for its swift setup capabilities for A/B testing. The beauty of this service lay in its

implementation of the Bayesian approach, which allowed us to run tests quickly. During the experiment, the product manager took the extra step of manually emailing talents in the treatment group, encouraging them to send their resumes. However, as time went on, the PM came to me with the request to end the test prematurely to alleviate the manual workload.

Action:
One of the challenges we encountered with Bayesian A/B testing was the absence of a specific sample size requirement, making it challenging to determine when to conclude the experiment. Being closely involved with the experiment, I kept monitoring the results on a daily basis. Gradually, the probability of the treatment group outperforming the control group reached an impressive 99% and remained stable, indicating a clear preference for the treatment group. Additionally, the credible intervals, which provide a range of plausible values, were reasonably narrow. While I cautioned the PM about the potential impact of missing weekly patterns due to the early termination, we collectively agreed to conclude the test. (I'm aware that there are other theories, such as expected loss and ROPE, for determining the optimal test duration.)

- **Metrics**: After communicating with the product manager, we selected the sign-up rate as our primary metric.
- **Randomized Unit**: Given the primary metric sign-up rate, in this test, we only focus on new users.
- **Prior Distribution**: Since sign-up rate follows Bernoulli distribution (a talent either finished signing up or not), we choose beta distribution as our prior distribution.
- **Sample Size / Length of the test**: There is no sample size limit on Bayesian A/B testing and the length of the test was not rigidly defined. But we advised the team to run it for at least two weeks to capture any potential weekly patterns. However, due to the PM's request for early termination and our internal discussions, we wrapped up the test after one week, collecting a total of 1450 samples.
- **Decisions**: After concluding the test, I delved into analyzing the results. The probability to be the best (it was generated using Monte Carlo Simulation) and credible interval are two metrics we used to make decisions. Our predefined threshold for claiming the treatment as the winner was a probability exceeding 90%. In this case, the probability of the treatment group being the best soared to an astounding 99%. Another significant metric was the 90% credible intervals, which encapsulated the central 90% of the posterior distribution. The lower end of the credible intervals rested around 5%, indicating a reasonably substantial effect size that would be brought about by removing the resume upload process.

Result:
Armed with these compelling findings, we made the brave decision to remove the resume upload step from the talent sign-up process. The subsequent month and beyond witnessed a

remarkable 6% increase in the sign-up rate, signifying the success of our experiment and reinforcing the value of our decision.

In the end, our fearless exploration, guided by Bayesian A/B testing principles, allowed us to revolutionize our talent portal, streamlining the sign-up process and empowering more talented individuals to join our platform. It was a tale of innovation, collaboration, and data-driven decision-making that reaped tangible rewards for our organization.

AM Experiment - Experimentation Design and Analysis

Stroy1: Normal Design of the Test (long-term to short-term)

Situation:
We built an optimization model to automate the process of adding talents into the recruiter's consideration bag, enabling them to swiftly assess talent profiles and make informed decisions that would accelerate the hiring process and boost job margins.

Task:
To ensure the success of this innovative model, my task was to design an A/B test that would thoroughly evaluate its performance. Here's how we went about it

Actions:

- **Randomized Unit**: Firstly, I needed to define the scope of our experiment. I decided to randomize the test at the job level, specifically focusing on unfilled jobs. I divided these new jobs into two equal groups, with one group receiving weekly assignments from our automated system (AutoMatch, or AM), while the other group would not.
- **Metrics**: time to close, margin per job, both of which are long-term metrics.

Next, I identified the key metrics that would help us gauge the model's effectiveness. I chose two long-term metrics: "time to close" and "margin per job." These metrics would provide valuable insights into how the model impacted the efficiency of the hiring process and the financial returns on each job.

Considering the nature of the hiring process, which typically takes a minimum of two weeks to fill a job, I decided to run the experiment for at least one month. This timeframe would allow us to capture sufficient data and draw meaningful conclusions.

However, to gain early insights into the model's performance, I devised an additional approach. I decided to monitor the "click-through rate" of talent profiles, which represents the number of profiles clicked divided by the total number of recommended talents. This would help us understand if recruiters were engaging with the tool and exploring the talent pool. Additionally, I kept an eye on the "remove rate," which indicated the number of talents removed from

consideration divided by the total number of talents recommended. This metric would provide insights into whether the recruiters found the recommended talents suitable for their needs.
By closely tracking these preliminary metrics, we would gain valuable indications of user adoption and engagement before analyzing the long-term metrics. This information would serve as an early indicator of the model's effectiveness and user satisfaction.

- **Length of the test**: It usually takes at least 2 weeks if not more to fill a job. Therefore, we decided to run this experiment for at least one month.

Stroy2: Low Adoption Rate

Situation & Task:
After two weeks of the launch of AutoMatch, we encountered a challenge—the adoption rate was only around 1%. Determined to improve this, we established a feedback channel for the recruiters, hoping to gather insights into their experience with the system. Unfortunately, even after one month, the adoption rate continued to fluctuate around 2%.

Actions:
Undeterred by this setback, we decided to dig deeper and understand why the adoption rate remained low despite the reasonably good quality of our recommendations. To gain insights into the system's performance, we focused on two crucial metrics: match fitness and estimated margin.

By calculating the match fitness, we were able to quantify the degree to which a talent and a job position were a good fit for each other. This metric highlighted that our recommendations were indeed reasonable and aligned with the recruiters' needs. Additionally, we estimated the margin, which involved subtracting the talent's preferred rate from the job budget. This allowed us to showcase the financial benefits and value that the system could bring to recruiters.

However, the challenge lay in increasing the actual usage and acceptance of our recommendations. In response to this, our CEO took an active role in encouraging the recruiters to explore the recommended talents and provide feedback. This feedback proved to be invaluable as it shed light on areas where our system could be further improved.

We carefully incorporated the feedback into our models, making enhancements that addressed specific pain points. For instance, we implemented limits on regions and critical skills that could not be overlooked. These adjustments were aimed at making the recommendations more tailored to the recruiters' requirements and preferences.

Result:
Through this iterative process of gathering feedback, incorporating improvements, and refining our models, we steadily began to witness positive changes. The CEO's push for engagement, coupled with the valuable feedback from the recruiters, played a pivotal role in driving the

adoption of our system. Gradually, we saw the adoption rate rise, and our recommendations started gaining traction and recognition within the recruiting teams.

This experience taught us the importance of actively seeking feedback and incorporating it into our models to enhance performance. By continuously refining our system based on real-world user experiences, we were able to overcome the initial adoption challenges and create a more valuable and widely embraced tool for recruiters.

Story3: Early Result in Long-term Metrics

Let me share a story about a situation I encountered while working on an experiment and how I resolved it. I had developed an analysis script to calculate two important long-term metrics, namely the time to close and margin per job, using the logging data. However, on the very first day of the experiment, I noticed something unusual. There were several jobs that already had values for these long-term metrics, which was impossible since it typically takes at least two weeks or more for a job to be filled.

Tasked with investigating the issue, I questioned whether it was a problem with my code, the experimental design, or the logging data itself. To begin with, I meticulously examined the logging data, and indeed, I found evidence of some jobs being closed. This discovery led me to alert the entire team, including the product manager, engineers, and other data scientists, about the anomaly.

During our discussion, one of the engineers pointed out that each job could have multiple positions associated with it. Intrigued by this observation, I delved deeper into the logging data and discovered that some jobs had a mix of filled and vacant positions. Consequently, this resulted in the premature observation of long-term metrics, as the positions filled before the launch of the AutoMatch (AM) system were affecting the analysis.

However, the engineer also mentioned that it wasn't feasible to randomize at the position level, which presented a challenge in the experimental setup. To address this, I took it upon myself to modify the code. I created a linkage between jobs and positions, excluded the positions that were filled before the AM system's launch, and conducted the analysis at the position level instead of the job level. This adjustment ensured that the analysis accurately reflected the performance of the AM system for the remaining vacant positions.

As a result of these efforts, we were able to ensure that the experiment was logged correctly and that the subsequent analysis was reliable and free from any distortions caused by prematurely closed positions. This experience taught me the importance of thorough data examination and collaboration with the engineering team to resolve issues and maintain the integrity of our experiments.

AM - Increase Cadence

Two weeks after AM was launched, the adoption rate is around 1%, and we only got two feedbacks from the recruiters. We understood the importance of gathering feedback from recruiters to improve the system. It urges us to discover potential approaches to motivate recruiters to use the tool and give feedback.

To delve deeper into the issue, I conducted an analysis where I plotted a histogram. The x-axis represented the time window between job postings and the invitation of talents to apply. The findings revealed that most actions were taken by the recruiters within 48 hours of posting the jobs. This presented a crucial insight – since we were originally running AM on a weekly basis, specifically on Mondays. For jobs which are posted during Tuesday to Sunday, they need to wait until the next Monday to get the recommendations. We were missing the golden opportunity to generate recommendations during this time-sensitive window.

To address this challenge, we proposed increasing the AutoMatch (AM) cadence from a weekly frequency to a daily frequency. However, this change came with several associated challenges that needed to be tackled:

1. Modify the code to enable dynamic recommended talents: We needed to enhance the codebase to allow for real-time updates and recommendations based on the latest available data.
2. Clarify the recommendation logic: It was crucial to have a clear and well-defined logic that determined which talents would be recommended to the recruiters.
3. Handling situations where recruiters ignored recommended talents: We needed to devise a strategy for situations where recruiters chose to ignore the recommended talents and didn't remove them from the talent pool.
4. Retaining invited talents in the pool: It was important to keep the talents who had already been invited to ensure a smooth and consistent experience.
5. Retaining talents with viewed profiles: Talents whose profiles had been viewed but not yet removed from the talent pool should also be appropriately managed.

To overcome these challenges, I collaborated closely with the product team to clarify the recommendation logic and ensure a seamless transition. Additionally, I worked in tandem with the engineering team to split tasks and modify the codebase, enabling dynamic daily recommendations.

Once the changes were implemented, the adoption rate went from 1% to 30%.

Recommender – choose LightGBM

Our main objective is to develop and implement a recommendation system that will suggest suitable talents for jobs. This system will enable self-service and reduce human costs. In simpler terms, our clients will be able to visit the website, easily find talents that meet their requirements, and send interview invitations without the need for a human recruiter's intervention.

To begin this project, I first met with the recruiters to gain an understanding of their talent selection process and criteria. Then, I delved into exploring the database to familiarize myself with the available data points. Eventually, we decided to approach this as a classification problem. From the recruiters' perspective, we considered job-talent pairs that were selected by the recruiters as our target 1. However, we encountered a challenge with an imbalanced dataset since the number of positive cases was limited. To address this issue, I proposed a solution: for each job, we randomly select an equal number of talents who were not selected, alongside those who were selected. While this approach may not be perfect, it serves as a reasonable starting point that we can refine later.

Next, I conducted exploratory data analysis (EDA) to examine each potential field, focusing on the percentage of missing values and the accuracy across different databases. After discussing with other team members, we decided to use LightGBM as our model. LightGBM is a state-of-the-art, tree-based gradient boosting model known for its speed, accuracy, and scalability. We chose LightGBM for the following reasons:

1. Many of our features exhibit correlations, and a tree-based model like LightGBM can effectively handle collinearity issues.
2. LightGBM helps to prevent overfitting, a common concern in machine learning models.
3. It offers a wide range of parameters that can be customized to suit our specific problem.
4. LightGBM outperforms other tree-based models such as XGBoost in terms of speed. It utilizes a histogram-based approach for gradient boosting, grouping data into discrete bins and building histograms to summarize the data. This approach allows for splitting using groups of data points instead of individually, leading to improved efficiency.
5. LightGBM is capable of handling missing values and outliers, which is crucial for our dataset. We have missing values in some key features that we prefer not to impute or discard due to the limited number of positive targets.

By selecting LightGBM as our model and considering its capabilities, we aim to address collinearity, prevent overfitting, and effectively handle missing values and outliers. This decision aligns with our goals and requirements for the recommendation system project.

Recommender – reduce data processing time

During the prototyping phase of our recommendation model, we encountered an issue where it took approximately 8 seconds to generate results for a single job across the entire talent pool. This was clearly not an optimal experience for our end users. To understand the problem, I meticulously timed each section of the code and discovered that the data pre-processing part consumed the majority of the processing time.

To optimize the data processing time, I collaborated with the team to brainstorm ideas and devised and implemented the following two strategies:

1. Conducted feature selection: By reducing the number of features, we were able to streamline the data processing pipeline and expedite the overall process.
2. Identified the most time-consuming feature calculation: Through careful timing, I discovered that calculating the similarity score for similar skills was taking a significant amount of time. To address this, I explored ways to optimize the calculation by vectorizing it and leveraging libraries such as NumPy and Numba to accelerate the computation.

By implementing these optimizations, I successfully reduced the time required to generate recommendations for a job from 8 seconds to less than 1 second. This significant improvement made our prototype ready for deployment, ensuring a much smoother and faster user experience for our end users.

Recommender – NLP with similar skills

When it comes to matching talents with jobs, skill is undeniably one of the most crucial factors. We recognized that SQL and MySQL, to some extent, can be considered as the same skill. To enable the ability to find similar skills, I utilized internal freetext data and job industry datasets that I found online. With this data, I trained a n-gram Word2Vec model, which generates embeddings for all the skills we have in our taxonomy.

The Word2Vec model has facilitated some of the most important features in our recommender. We also incorporated it into our search feature on the recruiter's portal. This integration significantly enhances the chances of finding qualified talents when recruiters input the required skills.

To demonstrate the effectiveness of the Word2Vec model, I applied Principal Component Analysis (PCA) to reduce the dimensionality of the embeddings from 300 to 2. This allowed me to plot some common skills like SQL, Python, R, Java, and C++, along with their neighboring skills. The resulting plots clearly illustrated that similar skills cluster closely together, showcasing the power of the Word2Vec model.

Recommender – automate feature selection

After creating the first recommendation model prototype, we discovered that it took approximately 8 seconds to generate results for a single job. This was not a pleasant experience for our end users. To understand the cause, I carefully timed each section of the code and identified that the data pre-processing part consumed the majority of the time.
I discussed this issue with the team and together we brainstormed various ways to speed up the pipeline. One of the solutions we explored was feature selection. Since we had 54 features, we

realized that many of them shared similar definitions. By implementing feature selection, we aimed not only to improve the running time but also to address the problem of collinearity.
As the main coder responsible for the data pipeline, I took the initiative to lead this effort. I saw it as a valuable opportunity to familiarize myself with different feature selection techniques available in the field.

One of my teammates suggested looking into SHAP values, which caught my interest. Upon further investigation, I discovered various techniques such as SHAP, Boruta, Permutation, and BorutaSHAP. After evaluating their methodologies, pros and cons, and estimated implementation time, I decided to proceed with BorutaSHAP. This technique is state-of-the-art and has an existing Python package that allows for quick implementation. To ensure clarity, I documented my thoughts in a Google Doc and shared the technical details with the team members. Fortunately, my proposal received their approval, and we moved forward with BorutaSHAP.

In the end, I successfully reduced the number of features by half while maintaining a similar model performance. Furthermore, I automated the entire feature selection process, including utilizing BorutaSHAP, determining different candidate groups using multiple BorutaSHAP results, Shapley values, and inherent feature importance. Additionally, I computed the model performance for models trained with different candidate groups.

By leveraging BorutaSHAP and automating the feature selection process, we achieved improved efficiency in our pipeline and made significant progress in enhancing our recommendation model.

Recommender – iterate model

Situations:
The model struggles to distinguish between highly qualified talents and those with less ideal qualifications. Most of the match fitness scores cluster around 90%, indicating a lack of differentiation. Only talents with match fitness scores of 98-100% possess all the necessary skills.

Task:
We need to improve our model’s ability to differentiate bad cases.

Actions:
We came up with some potential solutions:
First of all, incorporate the feedback from another model to build robust negative labels.
There is a model called AutoMatch, which is used to match talents with jobs based on match, margin, response rate, and other factors. It shares the same match fitness model with the recommendation engine. The match fitness model uses LightGBM model to calculate to what extent that a talent and a job is a good match.

In AM, the recruiters can remove the recommended talents if they are not qualified, which is a clear indication of bad cases. We can train a new model using data with clean negative labels and combine the newly trained model with the original model deployed.

Secondly, we decided to reconstruct the target labels to include the client's judgments in addition to the recruiter’s decisions.

Furthermore We also experimented with different loss functions during the training phase. Different loss functions prioritize various aspects of the model's performance. By exploring various loss functions, we aimed to identify the one that would optimize the model's ability to identify bad cases effectively.

Lastly, we were also thinking about applying deep learning approaches

Result:
Through these actions, we aimed to address the challenges and improve the model's performance in identifying bad cases accurately. By incorporating accurate negative cases, enhancing the target variable, utilizing deep learning approaches, and experimenting with different loss functions, we expected the model to achieve better accuracy and provide more reliable results.

Recommender – refactor training and data pipeline

Situation:
When we want to iterate the model, a minor change such as filter out some of the features, or update the training will cause a lot of changes in the code base.

Task:
It calls for a refactor of the existing pipeline in order to make the iteration more flexible.

Actions:
I communicated with the tech lead, explaining to him the function of each code block and redesign the pipeline with him. The training and data pipeline has the following functions:

1. Getting data from the database
2. Pre-process the data which includes data transformation and validation
3. Feature calculation:
    a. use the pre-processed data to calculate features for each job-talent pair
    b. Use Word2Vec to get the similarity score between skills
    c. Calculate IDF for skills
4. Train the model
5. Feature selection
6. Retrain the model with selected features and upload them into GCP

We redesigned the pipeline based on the idea that we need to keep the package that we ship to be as lean as possible. To make that happen, I implemented the following changes

1. We make the pipeline query-free. So the entry point of the pipeline changed from query to dataframe. By doing so, when we want to update our training data, we do not have to change the rest of the pipeline according to the queries as we used to.
2. Create a mono-repo to store data queries and common data transformation and validation procedures in order to be shared across different projects
3. Move Word2Vec and Skill-IDFs to the mono-repo and therefore make it independent from the training pipeline.
4. Enable training with other models such as SVM, XGBoost, in addition to LightGBM
5. Automate the feature selection process.
6. Automate the process of uploading all the trained models along with its training data to GCP for later use

Result:
The change makes the model more easy to be deployed and iterated because within the package that we deliver to the MLE team, we only keep those codes that are required for generating recommendations. Next time, when we want to replace the current model with a new one, what we need to do is just to change the model version in the code and there is no need for any additional engineering efforts. In addition, it also makes it easy for us to update the training dataset, and train the model with different algorithms.

Mininglamp – Sentiment Analysis & LDA

We used keywords to filter out those posts that are of our interests such as related to a certain brand, product, or campaign.

We then use Sentiment Analysis to find the negative comments

Use LDA to find the trending topics that the customers are talking about

Mininglamp-automation tool

We are creating an e-commerce performance tracking report for the clients. It took a lot of time to make the slides. After finishing the report for the first quarter, I proposed to connect with the IT team to automate some of the repetitive processes.

People are showing negative opinions to change. First, it takes time to communicate with the IT. Secondly, there may be bugs of the first version of the deliverable and it will take time to double-check the result, which may not save time in the next quarter.
More categories are added. The structures are set. Decide to develop a prototype on parts of the slides.

Improve efficiency and reduce manual errors. The prototype translated into a routine tool.

Mininglamp-responsible for media analysis

work with limited time or resources
The whole team is new to the company and only one associate manager has expertise in media analysis using the internal system. I partnered with my manager to track the quarterly performance of three brands. Our project is untraditional in a way that we need to combine eCommerce data and social media data to conduct analysis.

I was fully responsible for the media analysis even though I had no experience at that time.

Connected with mentors from another group to learn about how to come up with keywords to filter the data that is related to our project, and clean the data. I spend extra time after work studying the previous cases. Communicate with my manager to learn about the findings drawn from the eCommerce data so that I can combine the report in a consistent way.

At last, I independently generated the media analysis for the online shopping performance tracking report, successfully supporting the findings that we discovered from the eCommerce scraped data. Convinced the client to keep purchasing social media services for the next year.

Ipsos - Generation Z Analysis for an eCommerce platform

I was involved in a GenerationZ Analysis for an eCommerce platform. The key metrics in the age group 18-25 years old kept dropping. The company believes that this segment of users will be the engines of the future consumption. In order to seize this opportunity, the company came to us to with an urgent need to understand the young generation. In this project, we covered four major aspects. Life values, consumption values, online shopping situations, and brand perception.

We designed our research based on the structure of a well-established segmentation model used in the company. It involves both qualitative and quantitative technics. For the qualitative part, the researcher will visit some selected customers at their homes and observe their lifestyle and shopping process. Ultimately, we summarized 8 segments of GenZ customers based on their life values and consumption values, and 20 online shopping situations. There are some interesting shopping situations that I can share, for example, sometimes people go shopping online when they have a bad day and want to cheer themselves up. Some people are just bored and keep scrolling down the app to kill time. Some people are super interested in new products and want to be the first ones to try them. The examples can keep going. I can share more later if you are interested.

Following, we used quantitative research to quantify each segment and shopping situation, such as demographic information, hobbies, most frequently used eCommerce apps, and other social media platforms.

At last, we recommend seven potential scenarios for the client to focus on.

Ipsos - Use Eye-tracking Techniques

- Are Right, A Lot: work with incomplete data or info
- Learn and Be Curious: how do you stay inspired, innovative, and acquire new knowledge in your work
- Think Big: proudest professional achievement
- Dive Deep: did more than what was required
- Have Backbone; Disagree and Commit: the unpopular decision of yours

It was in a concept test, Concept A won over two others in every aspect except for package design. This is a problem that cannot be solved solely by utilizing survey data.

Two weeks before I ran into this test, I learned about eye-tracking techniques which can collect consumer behavior data to generate new perspectives for package evaluation. However, researchers hesitated to apply it, because it takes a lot of time to embed new technology into the traditional process and it is hard to convince clients to pay for a costly service which has the risk of failing to facilitate business decisions.

With the belief that behavioral data are reliable sources to solve the problem at hand and the eagerness to learn cutting-edge tools, I determined to propose the adoption of eye-tracking in the concept test I was involved in. I stayed late after work for a week to digest the training materials by myself, proactively booked my manager's time to walk her through my progress and discuss my ideas about how to implement the model in the project, and negotiated with the vendor to offer a discount for the first deal and implemented a full-version output into the pitch book to gain the client's permission.

Finally, I successfully employed eye-tracking equipment to collect untraditional data. By analyzing multiple sources of data, I discovered tagline positioning sometimes confused readers. After identifying the underlying cause and adjusting the tagline positioning, the product entered the market successfully. As one of the company's first adopters of the eye-tracking technique, I won the clients' recognition and repetitive orders

Ipsos - Use Logical Thinking to Solve Problems

- Invent and Simplify: simple solution to a complex problem

- Think Big: proudest professional achievement

It was in a usability and attitude study. An E-commerce client asked our team to identify the most promising scenarios to attract Generation Z customers in the short term. When our segmentation model failed, we had to develop a new solution.

Knowing the market size of each scenario, I asked myself what the underlying drivers were and which factors could most easily be improved. Following the logic, I examined the equation: market size is equal to the multiplication of customer base, purchase frequency, and bill size. I ultimately decided to focus on purchase frequency instead of customer base and bill size considering these two factors' short-term growth stickiness.

However, our purchase frequency was obtained from customer recall which is not that reliable. I consulted my manager and director, and we all agreed that the customer satisfaction score is a good proxy, assuming the more satisfied customers feel about the service, the more willing they would be to repeatedly use the APP.

And then I tried to map all potential scenarios in a bubble graph. I set the existing customer base as the horizontal axis, satisfaction scores as the vertical axis, and the market size as the bubble size. By doing so, I was able to conclude that the most promising opportunities lay in the lower right of the graph with objectively large market size, comparably big current customer bases, and relatively lower satisfaction scores. My proposal won praise from the client and was adopted as an alternative approach by our team to answer similar requests.

I regard this project as my favorite teamwork because the complexity of the project required a number of discussions to be held with the senior officials. During the discussions, I not only seized the opportunity to express my ideas and prove my value to the team but also closely observed the thought process of the senior members. By developing and following such a thought process I was able to quickly devise an effective solution for the problem.

Ipsos - Brand Health Tracking Reconduct Fieldwork

- Have Backbone; Disagree and Commit: disagree with managers

Logically presenting the data and facts is my favorite way to persuade team members. Once in a brand health tracking project, the awareness rate of a new-launched brand dropped significantly versus last quarter. The director attributed the problem to an execution error and advised me to re-launch the survey. But I found the data were reasonable as I compared the data of the current quarter with all the historical quarters and discovered that the drop is a long-term downward trend rather than an issue caused by the operation failure. I additionally asked the interns to collect the campaigns of the brand which launched in the current quarter. Only 3 online small-scale campaigns were found. I created 2 simple slides with charts to report to the director that low marketing exposure led to a reasonable downward trend and therefore there's no need to re-launch the survey. I successfully convinced him to keep the original results, hand in the report well before the deadline, and draw the client's attention to increasing marketing campaigns for the new brand to boost brand awareness. Following the authority without a second thought will never be my thing. I will keep thinking critically in my career, and actively contribute my unique perspectives to the discussions.

Ipsos - Brand Health Tracking Multiple Tasks

- Hire and Develop The Best: mentor someone
- Insist on the Highest Stardards: team member didn't meet our expectations
- Earn Trust: earn the trust of a group
- Deliver Results: handle multiple assignments

In a project, I was required to handle three work streams simultaneously, including monitoring data quality, drafting analysis requests, and editing reports.

Before kicking off the project, I design an excel template in advance to guide my colleagues from the data collection team to keep a close eye on the performance of key metrics. Also, I coached three newly recruited interns on how to use the auto-charting tools to edit reports and reserved enough time for them to practice. During the project, in addition to checking the key performance report provided by the data collection team every day, I also scheduled a conference call with my colleagues from the data processing team to discuss the application of advanced statistical models like mental network analysis. Subsequently, when all the data were settled, I immediately led the interns to edit two reports by listing notes for editing and checking their outputs regularly to make sure they were on track.

At last, I led the team to successfully delivered the reports to the clients. Through the report, the clients learned about the performance of their products and campaigns and made business strategies for improvement accordingly. This leadership experience has really taught me to address the key issues to different team members at the right stage so that I can successfully monitor each member to deliver their outputs on time.

Ipsos - Use Finding to Make Impact

Leveraging consumer insights to facilitate business decisions made me proud of my job. I once uncovered potential product cannibalization risk and persuaded the client to adjust the positioning strategy for the new product by conducting in-depth customer interviews. I also embedded the industrial cutting-edge eye-tracking technique into the traditional marketing research process to generate additional customer insights to revise the package design and helped the product enter market successfully. I timely delivered brand health tracking reports to inform the client of the performance of the products and campaigns for future adjustment.

Being able to apply both my data analytical skills and business acumen to formulate effective business strategies to drive product success is my key motivating factor in the Big Data era. That's what drives me to pursue a position in Analytics.

Ipsos - Package Test Repetitive Cancellation

- Customer Obsession: difficult customer
- Deliver Results: handle multiple assignments
- Teamwork

Once a package test was delayed three times because the client couldn't manage to get the test packages ready in time. In order to ensure the project could be completed in good time, I needed to work with 3 parties. Firstly, I double confirmed with the clients the final delivery date of the package and the report, so that I can make a schedule accordingly. Subsequently, I turn to the operation team to book the spot for the fieldwork but learned that there was no available spot since it was occupied by another research project. In order to solve this problem, I initiated a meeting with the operation team and the other research team, in which I advised the operation team to hire more part-time workers to increase efficiency and successfully convinced the leader of the other research team to reschedule their survey one day earlier. Eventually, I successfully secured a spot for our project. Lastly, I negotiated with the critical technic vendor to make sure that the required resources would be reserved by mentioning the risk of canceling the order if they couldn't meet the deadline. Ultimately, the efforts of each party ensured the projects were completed in time.

I regard this teamwork experience as a precious one for me because I learn to consciously think in each party's shoes, actively figure out solutions through collaboration, and most importantly push when necessary.

Ipsos - Quali and Quant Coordination

- Learn and Be Curious: how do you stay inspired , innovative, acquire new knowledge in your work
- Deliver Results: handle multiple assignments

We once received an inquiry about a usability and attitude study that required the coordination of a qualitative team and a quantitative team. My task was to integrate the content of each team to complete the pitch book in good time. I accomplished this task by initiating three major cross-team meetings. Back at that time both the qualitative team and quantitative team were occupied by multiple projects. Therefore, I scheduled the first meeting at the very beginning to set a practical schedule for everyone to prioritize their work. And then right after both teams delivered their first drafts, I organized the second meeting and suggested the supervisors of both teams present their research designs respectively. After understanding the selling point of each party, I started to integrate the contents to make the survey design consistent. However, the estimated schedule is more than a month which was beyond the clients' expected due date. In order to address this problem, I initiated the third meeting. The senior officials advised conducting certain parts of the qualitative and quantitative research simultaneously. By doing so, we successfully shortened the research duration. Eventually, the pitch book helped us to win the bidding worth approximately $80, 000. It is rather common to run into teamwork that involves cross-function teams. This experience has taught me to better perform in such situations by holding meetings on key milestones and designing proper discussion objectives to move the whole team forward.

Ipsos - Setting Boundary

- Customer Obsession: apologize to someone

I perceive myself as a reliable person, however, once I failed to deliver output on time for my team. This is an experience that I feel quite embarrassed about but learned a lot from. It was in my first month at Ipsos. Our team was over-occupied with multiple projects. In the meantime, the requests from other teams kept interrupting me. Once they asked me to double-check whether the current shelf arrangement is identical to the shelf design offered by the client. As a newcomer, I had no idea that this task was not my responsibility and regarded it as a little favor to offer. However, the task took much more time to complete than I had imagined, dramatically eroding my working time. Therefore, I missed the deadlines my manager set for a questionnaire design.

Although I informed the manager in advance to avoid negative chain effects due to the delay. I still felt the situation was out of my control. So, I booked my manager's time to have a talk with her after I handed in the questionnaire. During the talk, I sincerely apologized and told her that I found it hard to reject the requests from other teams even though they started to harm my efficiency. My manager recognized that some of the tasks assigned by other teams were not necessarily my responsibility. She told me that I should start to learn to prioritize work and set boundaries. She suggested I strategically allocate my time according to the importance and urgency of tasks and avoid unnecessary efforts by offering smarter solutions. The manager's advice was extremely enlightening to me. Later on, I organized a cross-team meeting and proposed a new communication scheme, which released me from tedious administrative work without compromising team efficiency.

There's no doubt that I have to handle multi-tasks in my future work. I will schedule my time strategically so that it will be more likely for me to excel.

Ipsos-fail to look at the big picture

As many other beginners, I lacked a big-picture view during the early days as research assistant in Shandong University, In my first data collection assignment, although I swiftly replied to all the data requests, the data from multiple sources were incompatible with each other, and turned out to be a failure in the model output. The professor had a serious talk with me and highlighted the importance of thinking in terms of a bigger goal. I sincerely apologized and explained I was so focused on efficiency that I overlooked the ultimate purpose of the exercise. Subsequently, I scheduled a meeting with the professor to understand the full research design and key obstacles. Once I understood these, I further cleaned up the data and completed the task successfully. Afterwards, I learned to consciously spend more time at the very beginning by asking questions to comprehensively understand the request, and therefore seldom get stuck in similar situations. In the coming projects which I will encounter in UC Davis, I will think twice before I get into execution.

Practicum - Understand Business Requirements

- Customer obsession: didn't meet client's expectation

Practicum is a hands-on project opportunity provided by the program. A team of students will be matched with a partner company. During the length of 10 months, we will leverage data analytics to solve one or more business problems the companies encountered.

I was assigned to a project in partnership with the pharmacy department at UCD Health. In this project, we aim to optimize the inventory of drugs. One of our deliverables is the expenditure dashboard. Every year, the pharmacy spends millions of dollars on drug purchasing. We want to provide transparency into the purchasing process and therefore allow the pharmacists to ask meaningful questions and discover potential financial opportunities.

Finding a common language with people who are not in the analytical world is important. (A lot of people want technology but they don't know how to express their needs or the complexity behind it.) The first challenge we encountered in the practicum is to get the correct requirements from the pharmacy team. New to the healthcare industry, it is hard for me to come up with meaningful questions to figure out the business demands in the first place. What's worse is that pharmacists are also rather new to the technical domain. It is hard for them to express their demands. Hence when we asked the partners what do they want to include in the dashboard, they either want everything or cannot name one thing at all.

In order to overcome the bottleneck, instead of asking some general and big questions, I tried to encourage the end-users to start from their daily work and come up with specific use cases that they are doing on the daily basis. I paid close attention as they expressed their thoughts, and identified those repeated processes and the key dimensions they want to deep dive in. Next, I quickly decomposed the requirements and drafted some simple graphs or charts to present and check my understanding. After the meeting, I assembled the team to review the notes and discuss the best ways of visualization to meet the requirements.

After several iterations and testing, the final dashboard was deployed and laid the foundation for the following inventory analysis.

(What's more, to help the pharmacists to overcome the fear of technical tools, we held a learning session to navigate them through the dashboard using concrete using cases and collected valuable feedback from them to iterate the dashboard. We also provided documentation that contains the interpretation of the indicators in the dashboard. The pharmacists can refer to it. We are happy to see at last that all partners are all amazed by our final deliverables.)

Practicum - Narrow Down Scopes

- Ownership: work on a project with unclear responsibilities
- Frugality: work with limited time or resources

6 months into the practicum, our focus is still on the expenditure dashboard. I can feel the frustration of my teammates because they don't have an opportunity to horn their analytical skills. The morale of the whole team is down. As the leader of the project and holding the belief that advanced predictive analysis will bring in more business values, I decided to shift the team's focus.

It was in the weekly meeting with our partners, I proposed to conduct a time-series analysis of the utilization data and predict future utilization in order to suggest more economic inventory levels that also take the patient's demand into account. To overcome the data quality issue, instead of focusing on all the drugs, I suggested that we can leverage the purchasing data which is rather clean to filter out drugs that make up a large portion of spending. Optimizing the inventory is the objective of our practicum and there is a large chance that changing the purchasing strategies of those effective drugs is more likely to lead to savings in spending. And by narrowing down the scope, it will be easier for us to manually check the data quality for each drug and can dramatically shorten the time to get the updated data.

Even though we didn't deliver a model due to data issues, we still gain the thinking practice as we played with the limited data we have. And after the proactive communication, the whole team regain passion for the project.

Practicum – Technical Challenges

One biggest problem we are facing is that it is hard to get all the data that are qualified to be analyzed. First of all, we need more than 15 features for the project. Those features sit in different datasets. The pharmacists know that they have the fields in for example purchasing or administration datasets and therefore we have the information we want. However, the reality is more complicated than that. In order to merge one feature with the existing dataset, we may need to combine three or even more tables to establish the correct relationship. Secondly, after combining the datasets, we found that there are a lot of missing values which is a problem they need to deal with the vendors and will take a lot of time. Thirdly, some observations in the data fields are not accurate which makes it hard to compare the price across different drugs.

Practicum – Dashboard Description

There are two dashboards I constructed in coordination with two other teammates. The first one is the expenditure dashboard. Every year, the pharmacy spends millions of dollars on drug purchasing. This dashboard was designed to provide transparency into the purchasing process and help pharmacists to identify potential trends and discrepancies. Currently, the pharmacists are using this dashboard on a daily basis.

The dashboard consists of two sub dashboards: an overview page and a drill-down page. Let's start with the overview page. At the top of this page, there are two rows of Key Performance Indicators (KPIs) displaying the Year-to-Date (YTD) and Month-to-Date (MTD) percentage of expenditure of three major vendors from whom the pharmacy purchases its drugs. Below the KPIs, you will find a bubble chart and a line chart side by side. The bubble chart illustrates the expenditure share of different hospital departments, such as emergency and cardiology. In the line chart, we exhibit the monthly trend of total expenditure. On the bottom, there is a table showing the YoY, QoQ, and MoM expenditure variances of different hospital departments. Filters are provided to give the user options to choose the year and comparison type. We also color-coding the variances to give the pharmacists a better alert. Both the bubble chart and the line chart are interactive, once you click on the element of the charts, the rest of the table will adjust to show the information of the corresponding departments or months. And you can go to the drill-down page from there.

On the drill-down page, we show the expenditure variances on drug-level. We also provide filters so that the users can slice and dice in their interest.

The bubble chart illustrates the expenditure share of different hospital departments, such as emergency and cardiology. Meanwhile, the line chart displays the monthly trend of total expenditure. Towards the bottom, there's a table showcasing the Year-over-Year (YoY), Quarter-over-Quarter (QoQ), and Month-over-Month (MoM) expenditure variances across various hospital departments. We've provided filters that allow users to select the year and comparison type they're interested in. Furthermore, we've color-coded the variances to provide clear visual alerts to the pharmacists. Both the bubble chart and the line chart are interactive, meaning that when you click on an element within the charts, the table automatically adjusts to display information specific to the corresponding departments or months. From there, users can also navigate to the drill-down page.

Moving on to the drill-down page, we specifically focus on displaying expenditure variances at the drug level. We've incorporated additional filters on this page so that users can slice and dice the data according to their interests and requirements.

These dashboards have proven to be highly valuable to the pharmacists, enabling them to gain insights into expenditure patterns and identify any potential issues. By providing a user-friendly interface with interactive visualizations and comprehensive filtering options, we've empowered the pharmacists to make data-driven decisions and effectively manage the pharmacy's drug purchasing process.

Hackathon - Correct the Problematic Approaches

- Bias for Action: saw some problems and take the initiative to correct

I partnered with two of my classmates to participate in the Hackathon held by UC Davis in partnership with Google Cloud. We tried to discover how did Covid lead to the escalation of Asian hate crimes.

Our project was built on one of the teammate's previous projects. By doing so, we don't need to spend too much time determining and collecting the datasets. However, I found the variables they selected and the analytical approaches they used are problematic which will lead to multi-collinearity, loss of information, and misspecification problems.

I first proposed to exclude the variables that share similar definitions such as claimed income and personal disposable income to avoid multi-collinearity. Secondly, I convinced them to change from logistic regression to linear regression to avoid loss of information. Logistic regression is an approach used for binary classification. In order to meet the model requirement, they converted the number of criminal cases to binary, which sacrifice a lot of the critical information. Linear regression is a more intuitive solution for our case. Lastly, I related to the news I read, and believe that former president Donald Trump's racist speech may also play a role in the rise of the crimes. Therefore, I recommended including that in the model.

My considerations helped us to build a model that is theoretically solid and easy to be interpreted.

Hackathon - Switch from NLP to Desk Research

- Bias for action (take a calculated risk)
- Frugality: work with limited resources and time

I partnered with two of my classmates to participate in the Hackathon held by UC Davis in partnership with Google Cloud. We tried to discover how did Covid lead to the escalation of Asian hate crimes.

As an Asian myself, I related to the news I read and believe that former president Donald Trump's racist speech may play a role in the rise of the crimes. , I recommended including that in the model. There are 7 “Chinese virus” related tweets tweeted by Trump. It isn’t a good attribute to include in the model. Firstly the small magnitude will determine that there will not be a significant relationship. Secondly, what really matters is not the number of tweets Trump posted but the impact brought by those tweets, such as how many people tweeted similar content after him. In order to measure the impact, complicated and time-consuming natural language processing techniques are needed.

With the belief that improper public speech is a crucial factor in the escalation of Asian crimes and the limited time at hand, I proposed to switch from NLP to desk research to help back up this point. I searched and found that there are published papers that are on the topic. I quickly went through them, summarized the findings, and incorporate the content into the slides.

The output enriched our final report. The judges appraised us for using multiple sources to support our stands.

Hackathon - Storytelling

- Learn and Be Curious: solve a problem through superior knowledge or observation

I partnered with two of my classmates to participate in the Hackathon held by UC Davis in partnership with Google Cloud. We tried to discover how did Covid lead to the escalation of Asian hate crimes. In the final linear regression model, we chose hate crime cases as the independent variable, covid-19 cases, economic environment, and public speech are the dependent variables.

From the model result, we learned that personal income is positively related to the number of anti-Asian crimes. while the unemployment rate is negatively related. The finding seems counterintuitive at first glance. I leveraged the economic knowledge I acquired in my undergrad and linked them with the news I read. I came to the conclusion that this is the economic dilemma brought about by the pandemic and government interventions.

(To be specific, when Covid-19 first impacted the US, the government advised people to stay at home to flatten the curve. Business owners dismissed lots of workers to save costs. In order to help people who lost their jobs because of the pandemic and incentive consumption, the government provided three rounds of relief payments. When the number of confirmed cases went down and the government wanted to bring the economy back on track, the side effects came in. People felt comfortable with the free payment and were less motivated to return to work. As customers went back offline, business owners urgently needed more workers to fulfill the demands. But they found it hard to hire a worker. That's why they increased the salary to attract people to work. To summarize, the pandemic severely depressed the economy. The government's policies have short-term benefits but brought pain in the long run.)

My interpretation help the team straighten the storyline and was agreed by one of the professors who focused on economic research.

Technical Project 1 – New Feature Performance Evaluation using DID and CLV

In this project, I want to evaluate the effectiveness of the new online gaming community feature for a gaming company's latest game. This new feature aims to increase user revenue and retention. I used Diff-in-Diff to calculate the effect on revenue and used logistic regression to calculate the retention rate and further calculate the customer lifetime value to measure the new feature's long-term effect.

Technical Project 2 – New TV Design Using Conjoint Analysis

Conjoint analysis is a technique that helps us understand how the customers value different features of a product or service. This technique allows us to determine the attribute importance, ideal product profiles, the price premium for brands, and ultimately the optimal price in terms of profitability.

One company is going to launch a new TV. The analytics team is required to determine an optimal price for the TV. We first gather key features that are likely to influence customers' purchase intention such as brand, resolution, price and etc. And then for each feature, we will assign some levels. For example, for resolution, we have 4K and 8K. Next, we created different versions of TV containing all combinations of different levels of features. We will ask some customers to rank those versions based on their preferences.

After gathering the data, I used linear regression to regress all the features on the rank to get the coefficient of all the variables which are also called part of worth. Based on part of worth, we can calculate willingness to pay and then get the market share at different price points. The price points that lead to the largest market share will be the optimal price.

Technical Project 3 – Brand Map: use PCR to Visualize Brand Competition

A brand map is a graph that informs which brands compete more strongly with others. It is a good way to visualize competition.

The iso-preference line is composed of offers that are equally preferred by customers - along the line, customers are indifferent between the combinations of factors. Offers to the right are more preferred by customers whereas offers to the left are less preferred. On the other hand, the regression line in our case estimates the “overall preference” score based on the different benefits/factors. Each point in the regression line corresponds to the estimated dependent variable value based on collective independent variables.

An ideal vector is the vector to find the importance weights of the evaluation dimensions on the preference. This is perpendicular to the iso-preference lines mentioned above, representing the direction of increasing preferences. When we create a perceptual map to show the consumer’s understanding of the positions of competing brands, the estimated coefficients from the attributes are required to build the axes on the map. The ratio of these two preferences becomes the slope of the ideal vector. With the direction of the vector towards the northeast, it shows us which brand should improve the product design.

We survey customers on attribute rating and overall preference rating for different cars. The ideal number of attributes will be between 15 to 20 and the number of brands is good to be between 5 to 10.

After getting the data, we use principle component analysis (PCA) to combine several attributes to create benefit factors. Following, we use scree plot or eigen value greater than one (E-G-O) rule to determine the number of benefits to retain. And then, we name the benefits based on the weights from the eigenvectors. The larger the weight, the more important the feature is in a benefit. Next, we will regress the filtered benefits on the overall preference and plot the brands of a two-dimension benefits space for each pair of benefits.

Technical Project 4 – Build Hotel Database Using Web Scrapping

In this project, I scraped the listings on booking.com from March 20 to March 25, for three different popular vacation locations in California – San Diego, Los Angeles, and Yosemite National Park. Below are five processes to complete the task.

Firstly, I used Selenium to open a browser and manually navigate to booking.com. By doing so, I found that booking.com is not actively detecting Selenium and hence it is a good target for web scrapping.

Secondly, I leveraged Selenium to automate the search procedures, including typing in the trip destination, selecting the check-in and check-out dates, and ultimately reaching the result pages. I then saved the first 5 result pages for each of the three locations to disk.

Thirdly, I utilized BeautifulSoup to pulled out the saved pages and parsed out critical information including location, hotel name, link, distance from center, price, review score, number of reviews, star ratings, free breakfast, and free cancellation options.

The fourth step I took was to query each hotel address's geolocation via the API provided by positionstack.com.

Lastly, I created a MongoDB collection and store it as a JSON file.