Final Report:

Shark Tank Assistant

Version 2.0 approved

Leann Blanc, Colin Chenoweth, Aditi Mukkara, Sahil Patel, & Mehlam Saifudeen

December 4, 2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Initial version	9/5/23		1.0
Version 1.1	10/20/23	Progress Report 1st half	1.1
Version 2.0	12/4/23	Final Report	2.0

1. Background

1.1 Purpose and Audience

This project report explains the various requirements for the first release of the Shark Tank Assistant Application. The user base for this application is wide, primarily targeting sharks and applicants on Shark Tank. However, the user base can also include users of other shows such as Dragon's Den, a spinoff of Shark Tank in Canada and Poland, Money Tigers in Japan, and Tu Oportunidad (Your Chance) in Spain along with other Shark Tanks in countries such as India. The document was created to guide the development of the application with the goal being to develop each of the listed requirements. For the users, this document provides information on the functionality and usability of the application as well as progress thus far.

1.2 Product Scope

With entrepreneurial spirit rising around the world, fostering creativity and encouraging individuals in their aspirations is more important than ever. One such way is Shark Tank which provides seed funding for individual companies. Our application allows for users to tailor their Shark Tank application based on their specific company and product details including but not limited to industry, size, ask amounts, equity offered, and previous sales. Benefits of this app include allowing users to better obtain funding and valuation deals, instead of potentially less meaningful applications. In addition, our application can be used by sharks who are investing funds into the companies to see how their previous investments match up to new products being pitched.

2. Technical Project Requirements

2.1 Product Perspective and Goal

- Create a program that provides SharkTank contestants with personalized feedback on their proposal and product, including but not limited to potentially revealing whether a shark will accept their offer.
- Utilize available datasets and past episode transcripts to train the model.
- Have a user interface that is a pseudo Shark Tank application to get information on the product and proposal, and return useful feedback.
- The interface will also have graphics that show users how the sharks have invested in companies across the years along with which industries, products, and pitches have had the most success.
- In addition to this the app model looks at keywords and phrases that are a part of some of the successful pitches of Shark Tank across the years.
- Measure "usefulness" by comparing the original feedback generated to the feedback generated once the user has incorporated the recommended first round of changes in their pitch.

2.2 Product Functions

2.2.1 Front End

The first page includes graphs, tables, and other interactive visualizations to provide the user with general information regarding previous Shark Tank contestants. For example, the page starts with interactive pie charts showing users the composition of pitches by industry for all pitches as well as successful pitches. Additionally, the user can choose from a variety of bar graphs displaying different information such as the ask amount or deal amount based on what industry their product is going to be. The goal of each contestant is to have a shark invest in their pitch. To gain insight into each shark, the user can select a shark's name to be provided with a variety of relevant information such as the average amount the shark invested per industry, the average equity amount per industry, the highest amount invested, and the proportion of pitches invested in.

For example, the user can select filters such as the industry they are interested in or the investor they are targeting. Accordingly, the page will display information such as the average investment amount per investor per industry. This page will also include interactive elements such as pie charts and bar graphs.

The second page includes an application for the user to fill out. This application includes specific aspects of the actual 17-page Shark Tank application. Some of these aspects are the industry, pitcher gender, state, age, amount asked, and whether or not they are profitable amongst others. Once the user submits the application, the program uses the built-in model to tell them if their pitch will be successful or not on the Shark Tank show. In addition to this, the model will work on its own feedback to see if there is a further improvement in the chance of the entrepreneur getting a deal if they choose to accept the initial feedback of the model.

2.2.2 Back-End

Feature Extraction:
The goal is to create an application-like feature set. It is easy to expand from simple facts about the proposal/product, to eventually include more complex data like the transcript of an episode or the description of a product/demonstration. To start off, we created a program to extract the individual pitches from the set of transcripts of the episode data that we had collected. From here we were able to build an NLP to extract additional feature data from the pitch such as if the entrepreneur's company was profitable and what were their total sales since the start. Moreover, we created a program to obtain the ten useful words in determining successful pitches in each category (after removing common words such as 'the', 'um', etc). This was done by looking at the words with the largest weights when training a logistic regression model using the frequency of each word as a feature. These words then function as new features by looking at how frequently they occur in each pitch to help make the model perform better. This can also tie in with using the model to help the user further edit their pitch to get the best deal possible

Machine Learning Model(s):
In order to understand which model would work best for our application, we compared the runtime, accuracy, precision, and F1 score for several classification ML models, both supervised and unsupervised, such as Logistic Regression, SVM, Neural Networks, Gradient Boost, Random Forest, and Naive Bayes classifier models. After doing some manual Backward Elimination feature selection, the Logistic Regression model turned out to have the highest F1 score (0.79) and close to the highest accuracy (0.65) and precision (0.65), so we decided to incorporate this model into our application to give the user the best predictions for their pitch.

2.3 Dataset

For general statistics regarding past Shark Tank investments we will use this open source dataset providing details on episodes. This dataset includes information from the contestant's application such as their original amount ask, the original equity ask, their gender, and their industry. It also includes information regarding how successful the contestant was on the show using measures such as whether or not they got the deal, how mean each investor invested, and the deal valuation.

Shark Tank US Dataset

For further information regarding rhetoric or negotiations after our developments with this data, we looked into other data sources such as Shark Tank show transcripts or data from other similar shows across the world. From here we were able to create an NLP to obtain additional information regarding the past pitches such as sales numbers and profitability to improve the model we have.

2.4 Code

https://github.com/ColinChenoweth/Shark-Tank-Assistant

3. Project Execution

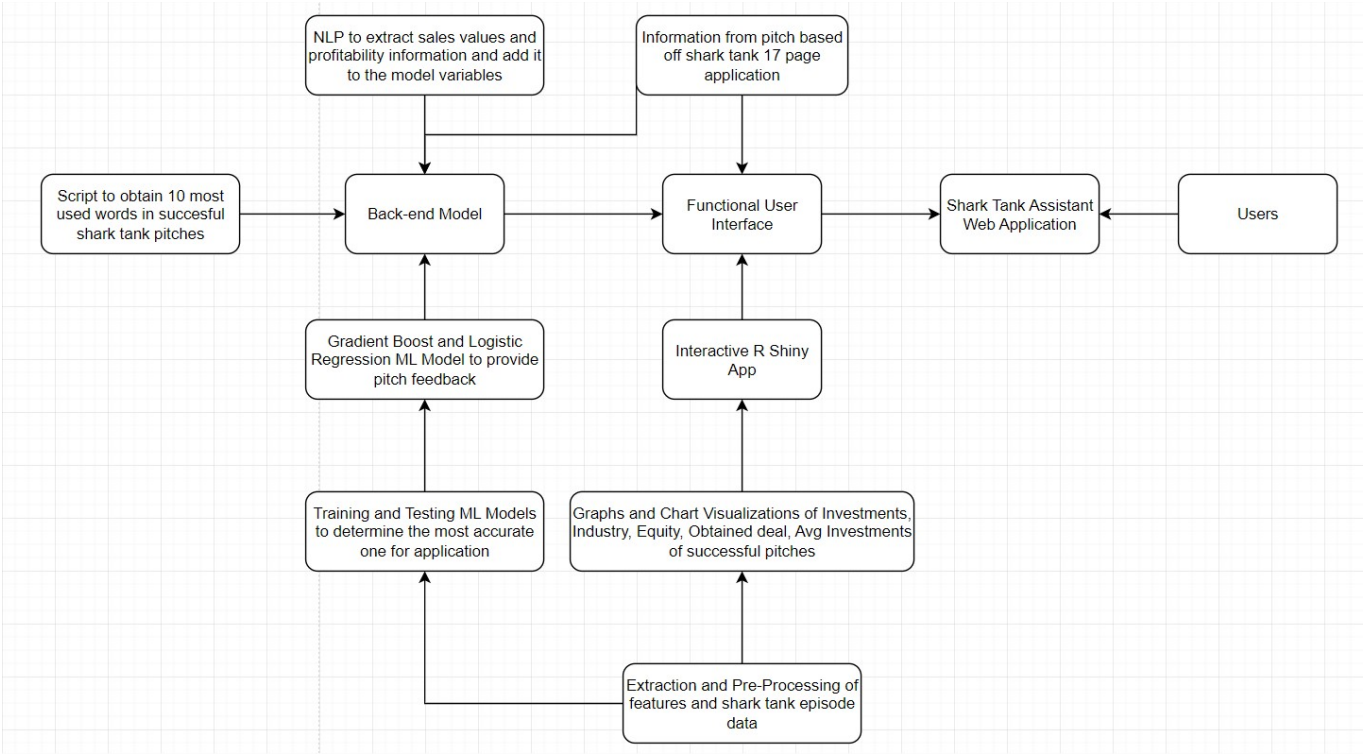
3.1 Completed Work

To start off, we were able to obtain a dataset that included product information, valuation, and final deal, which sharks invested, among other variables which we used to train the model and create a variety of visualizations. We created a variety of visualizations that a user can use to guide them if they do not actually want to run the model. To start off, we have an interactive pie chart that allows the user to see a product breakdown by industry of all pitches and one of the pitches that sharks invested in. We created a bar chart showing the Original Offered Equity vs. the Total Deal Equity Per Industry on Average as well as a bar chart to show the breakdown of the pitchers' gender per industry. Additionally, we created an interactive shiny interface that allows users to select an industry they are interested in and choose whether they are interested in seeing a bar chart of the

original ask amount, valuation requested, total deal amount, or deal valuation for products in that industry from past episodes. This can give users an overall idea of what they can expect before filling out an application for Shark Tank or using our model to determine a valuation for their company. There is now also another interactive Shiny app that provides information about the specific coaches. This is helpful if a user is interested in working with a specific coach, then the user is able to find information specifically pertaining to them.

In addition to this, we obtained the transcripts of close to 240 episodes from Hulu which can be used to build an NLP that can help us expand on the current dataset, allowing for more training data to be used to make the model more precise and accurate. We also extracted the most important features that can help both us and the user determine whether their business idea would obtain an investment from the shark or not. These features included industry, pitcher average age, gender, state, city, original asked amount, sales amount, profitability, and the original amount offered and equity offered. Using the data we used 7 different Machine Learning models including Random Forest, Naive Bayes, Gradient Boost, and SVM, amongst others. The accuracy, precision, and F1 scores of the models were also included to see which one would perform best. From here we obtained the highest accuracy and precision for the Gradient Boost Model and decided to select that for our application. Combining the data in the features and the ML models we were able to obtain estimate scores of whether a SharkTank deal was obtained or not based on the features mentioned.

A design of the front-end of the application was developed using Flask which incorporated all the main features that are intended to be included in the front-end of the web application. The front end will interact with the backend model that will be used to provide feedback on a user's pitch idea using important information from the 17-page long SharkTank application. The front end now contains the charts and graphs developed in the R shiny app in an interactive manner and provides a user-friendly environment.



3.2 Final Application Vs. Design Specifications

We were able to successfully implement all of the features in the application that we had listed in our design specification. When the app.py code is executed, we see the front-end design implementation. The very first page shows various interactive graphs and charts such as a pie chart to describe the distribution of the various types of industries for all proposed pitches. This is followed by a pie chart showing the percentage of pitches in each industry that accepted a shark offer along with an interactive table that allows the users to select industries and any other specific data they want to look at. This description of the front-end very well meets the project design specifications made both during the initial and midterm reports. The next aspect of the application is the highlight of the project is the personalized feedback using a model that looks at variables such as equity offered, ask amount, type of industry, pitcher's age, and sales amount amongst others to predict whether a shark would invest in their deal or not. We were able to generate a model using the pitch transcripts to give a prediction value for this and some other variables such as total deal amount, equity, and deal valuation as well. There were also scripts created that could extract additional pitch features from the transcripts such as the most frequently used words in successful pitches and sales and profit information. All of these aspects of the model backend match the design specifications.

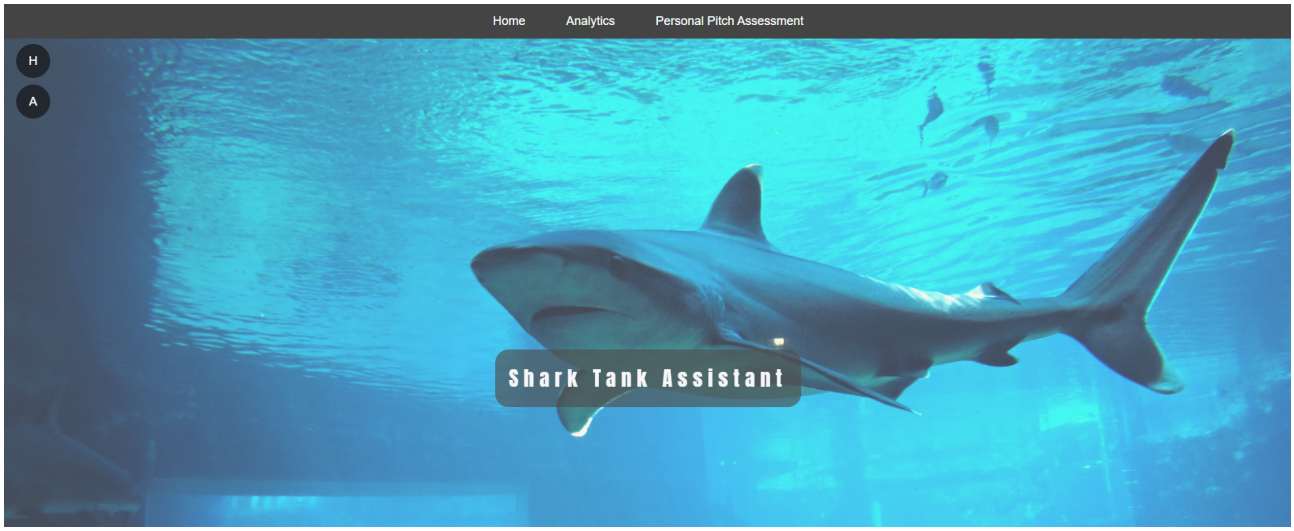
One aspect of the project that we had included in the design specification but were unable to provide in the final product is a feedback system where once the user implements the changes in their pitch they can get an updated prediction. However, this may be possible through re-entering all the data as per the shark tank application once again. In addition to this, the model can only

provide predictions to a certain level of accuracy and does not currently provide a lot of feedback on the pitch or the business description itself. These features are some that we had planned to include in our final application but were unable to do so for the deadline. Nevertheless, we do plan to include these soon to make the application as helpful and reliable as possible in the future.

3.3 Testing

We did test our application using example pitches from the latest shark tank season. The results we obtained are as below:

1. The image above shows what the user sees when they navigate to our webpage! At the top, they can choose to navigate to the home page, the analytics section, or the personal pitch assessment page. The analytics section is located if the user scrolls down or selects the “A” as well. The analytics page contains visualizations and graphs for the user to learn more about Shark Tank and previous pitches. The Personal Pitch Assessment lets the user submit a simplified form and receive feedback!

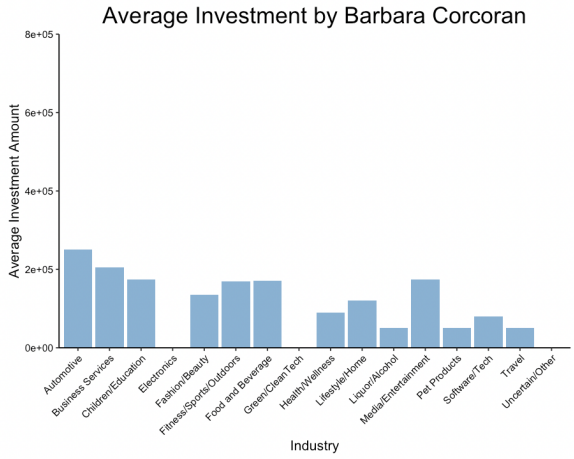


2. On the first page, the user is presented with interactive visualizations and charts on the front-end of the web application. This allows the user to gain insight into relevant information regarding Shark Tank prior to submitting their pitch. One of the sections allows the user to learn about a shark. On the left hand side there is a dropdown that allows the user to select their shark of interest, and several charts are returned displaying information about average investment per industry, highest investment, average equity per industry, and more.

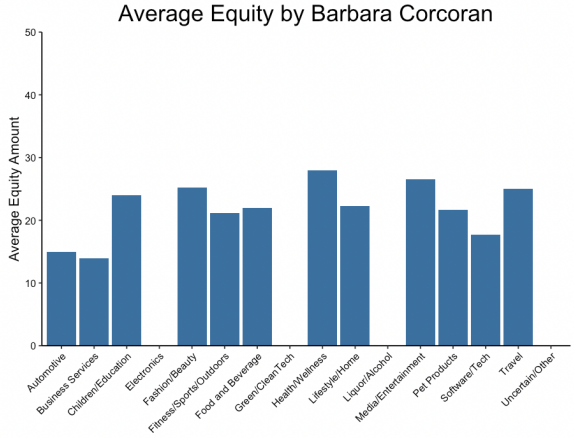
Learn About A Shark!

Select Shark:

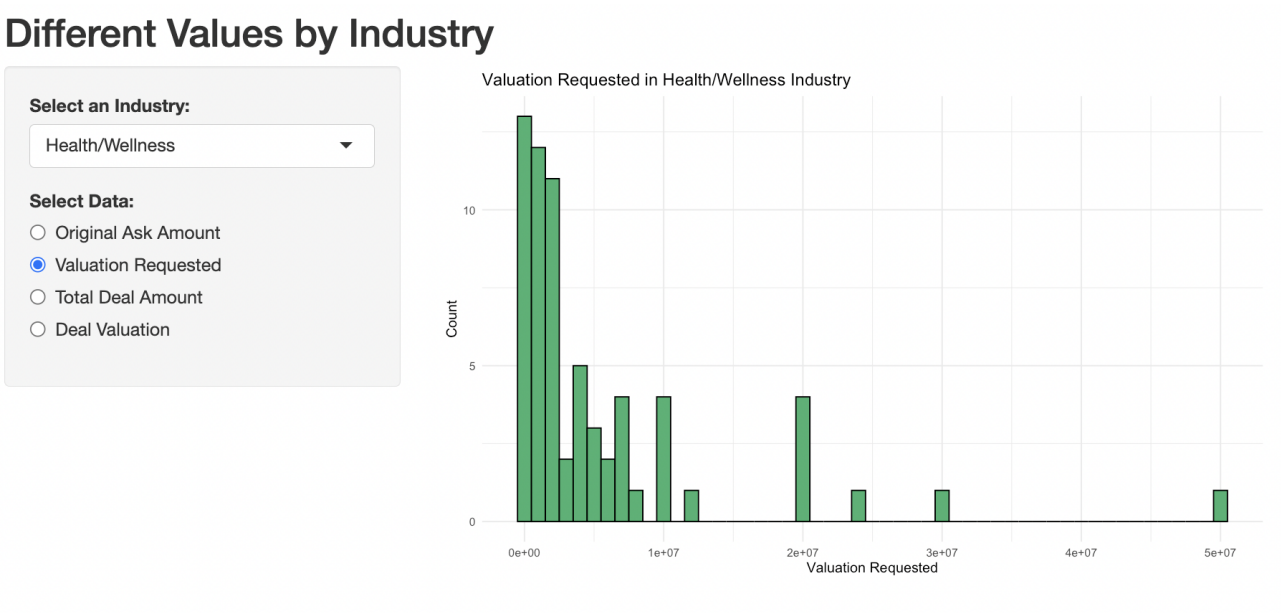
Barbara Corcoran



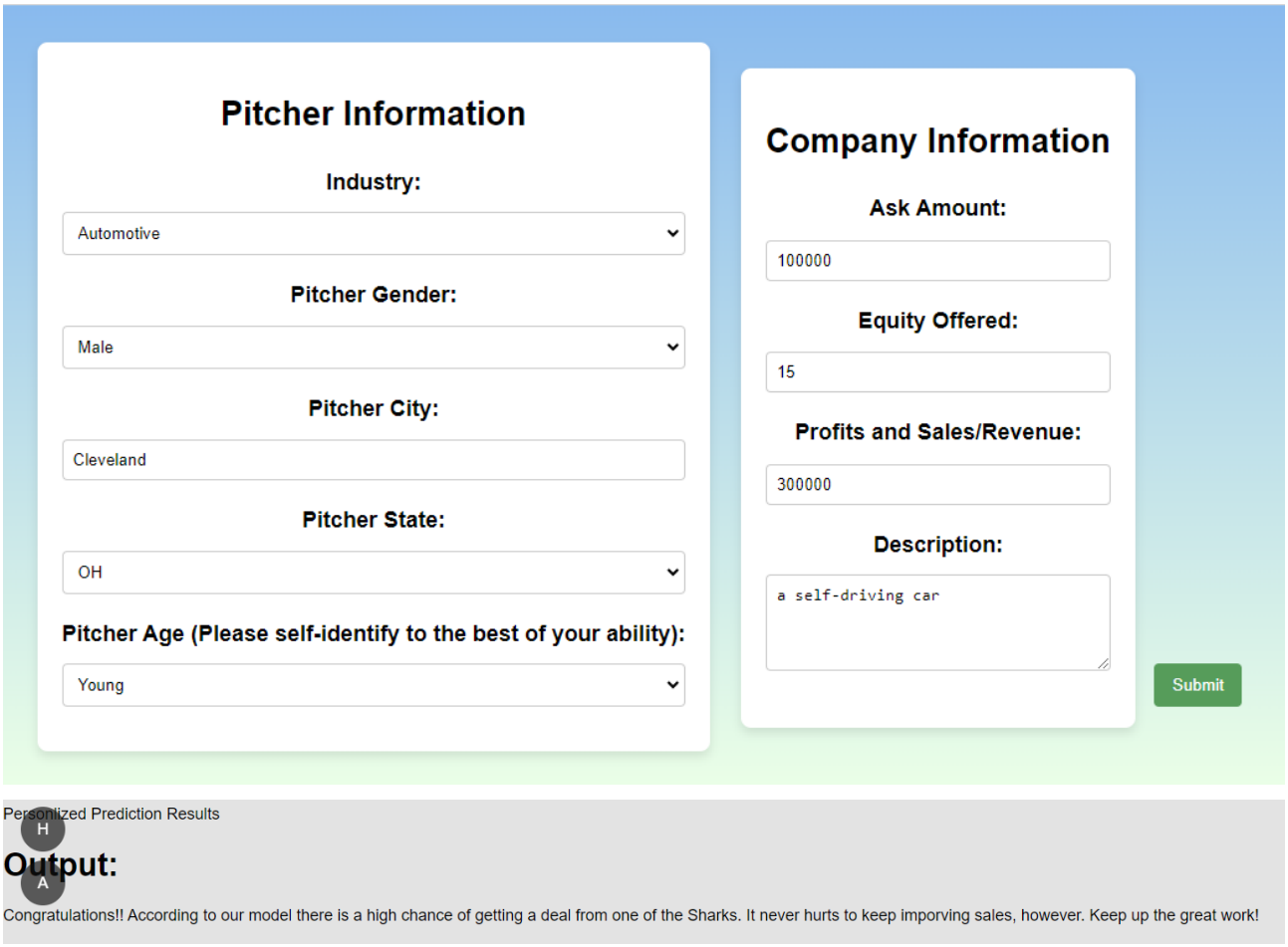

```
Highest Investment by Barbara Corcoran : $ 700,000 in Fashion/Beauty
```



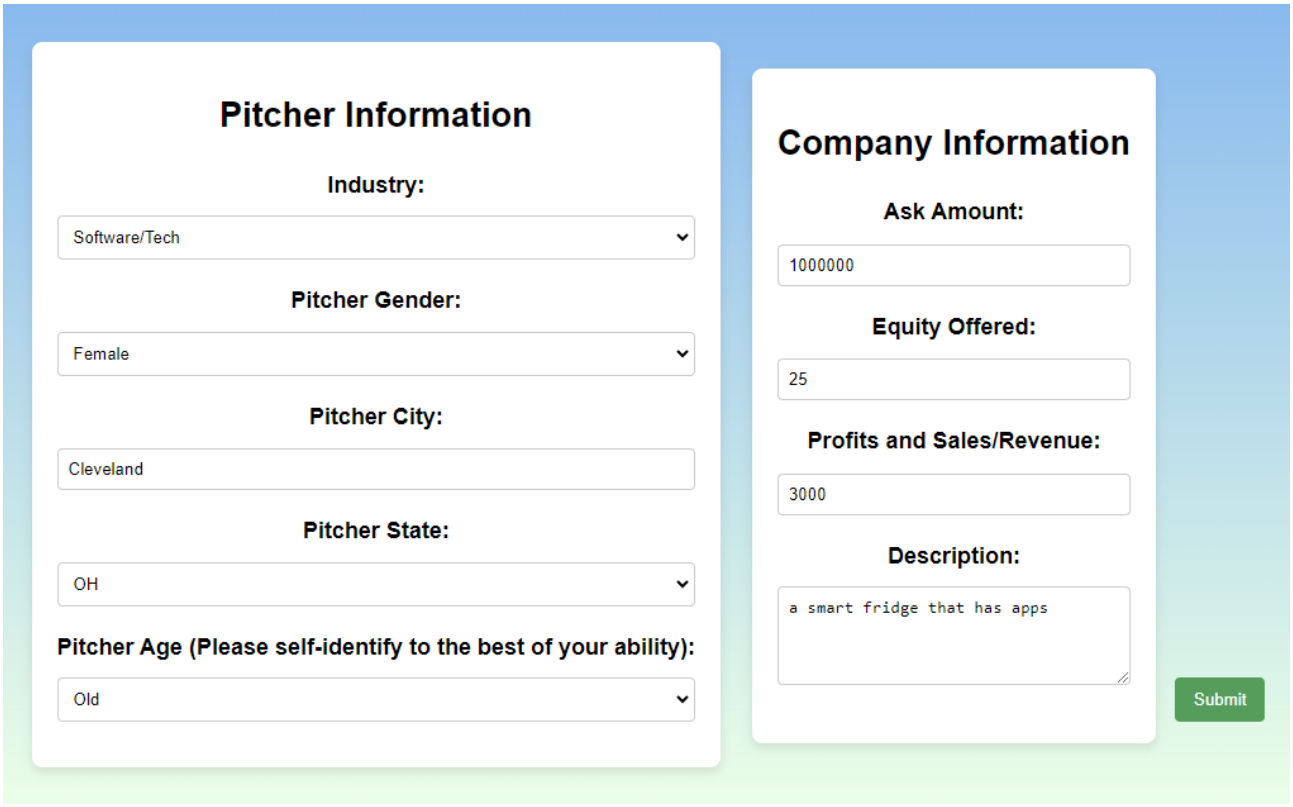
Here is another example of one of the charts the user is presented with on the main web page. This allows the user to gain more information regarding a specific industry of interest. The user can also select which attribute they want to know more about: original ask amount, valuation requested, total deal amount, or deal valuation.

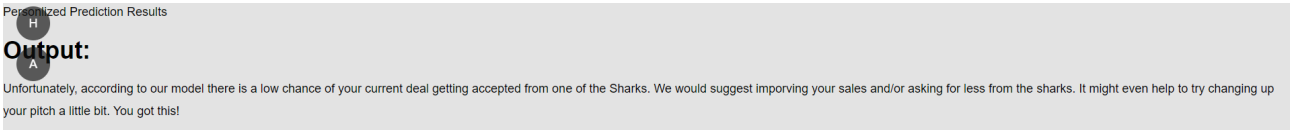


3. Next we have the personalized pitch feedback form which asks for some of the important information about the entrepreneur's business so that our model can process the information and predict whether the entrepreneur gets a deal or not.
    a. Below we have a screenshot of the input for a pitch that we predict would obtain a deal in the Shark Tank. Some of the metrics that play a role in obtaining a deal are the equity, the product and sales revenue, the asked amount, and the equity offered. We can also say what can be improved in the pitch or in the business in order to obtain a better deal from the sharks.



b. Similarly below we have pitch information from a deal that we predict would not obtain a deal in the Shark Tank. This can again be determined mainly from factors such as the amount requested, the equity offered, the sales revenue, and the industry as well. We also provide some feedback to the entrepreneurs that can help them in the future when raising venture funding from sharks or any other independent Venture Capitalists.





3.4 Timeline

- [x] ~~Week 1: Set up environment~~
- [x] ~~Week 2: Submit Project Proposal~~
- [x] ~~Week 3: Find dataset and episode transcripts~~
- [x] ~~Week 4: Research and decide on best models~~
- [x] ~~Week 5: Train/debug various models for basic classification~~
- [x] ~~Week 7: Plan user interface and complete the implementation of first look of UI~~
- [x] ~~Week 8: Submit Progress Report and Demo Video~~
- [x] ~~Week 9: Select best Model in backend for Individualized Pitch Feedback~~
- [x] ~~Week 10: Expand more on the Model to Improve Accuracy~~
- [x] ~~Week 11: Train/debug model for all feedback~~
- [x] ~~Week 12: Combine model and user interface~~
- [x] ~~Week 13: Debug and combine all the code and make poster for intersection~~

- ☑ ~~Week 14: Practice presentation~~
- ☑ ~~Week 15: Submit Final Project Code and Report~~

3.5 Individual Contributions

A. Leann
    a. R charts and Visualizations (Interactive Pie Charts by Industry)
    b. Shiny App Front-end and Visualizations (Allows user to select an industry and specific value they want visualize)
    c. Plotly and Shiny integrations into webpage
    d. Website Front-end Design
B. Aditi
    a. Shiny App R charts and Visualizations regarding each Shark
        i. Average investment amount per industry
        ii. Average equity amount per industry
        iii. Highest investment
        iv. Proportion of pitches invested
    b. Website Front-end Design
C. Sahil
    a. Building Machine Learning Models
        i. Logistic Regression
        ii. Decision Tree
        iii. Random Forest
        iv. Support-Vector Machine
    b. Website Front-end Design
        i. Built a form page
        ii. Built a home page
        iii. Converted project to Flask to ease integration of python backend with front-end
D. Colin
    a. Building Machine Learning Models
        i. Naive-Bayes
        ii. Neural Network
    b. Testing initial models
    c. Preprocessing of transcripts
    d. NLP term frequency, and choosing top influential words in each category
    e. Performed feature selection
    f. Integrated backend model into the application as a whole
E. Mehlam
    a. Building Machine Learning Models
        i. Gradient Boost
    b. Deal determining features and variables data extraction
    c. Software Design Schema

d. Created an NLP to extract sales numbers and profitability from each pitch.
e. Integrate backend model with the application as a whole