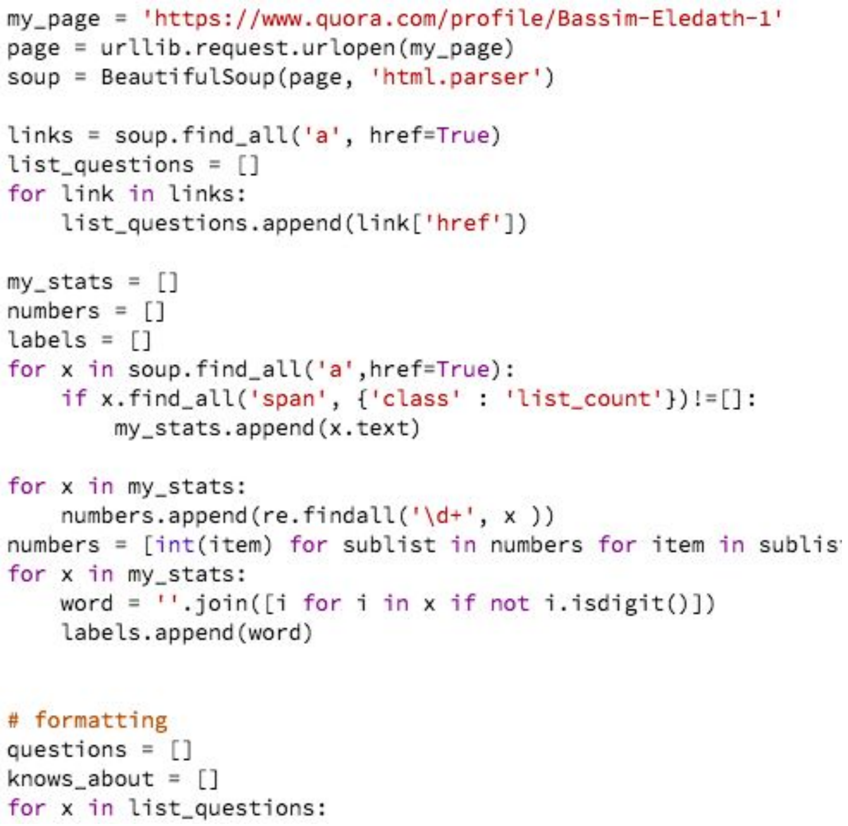
Why Quora?

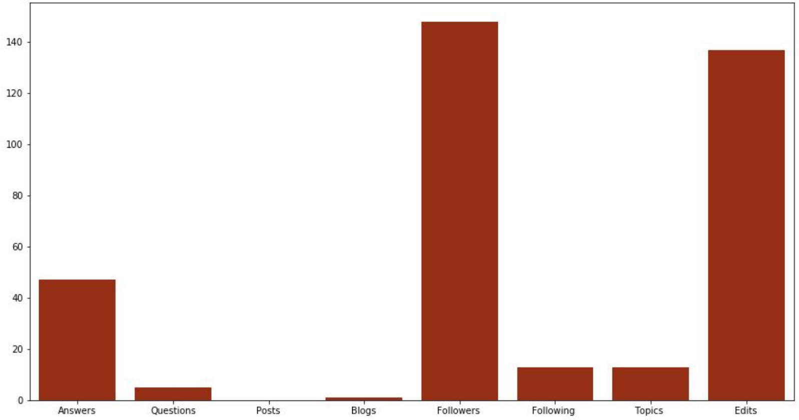
Before I get into it, let me tell you about my **24-hour Quora challenge**. I gave myself 24 hours to find and play with any quora-related data I could find on the internet. You can see what I finally ended up with by clicking here.

Project 1: Scraping my Quora profile

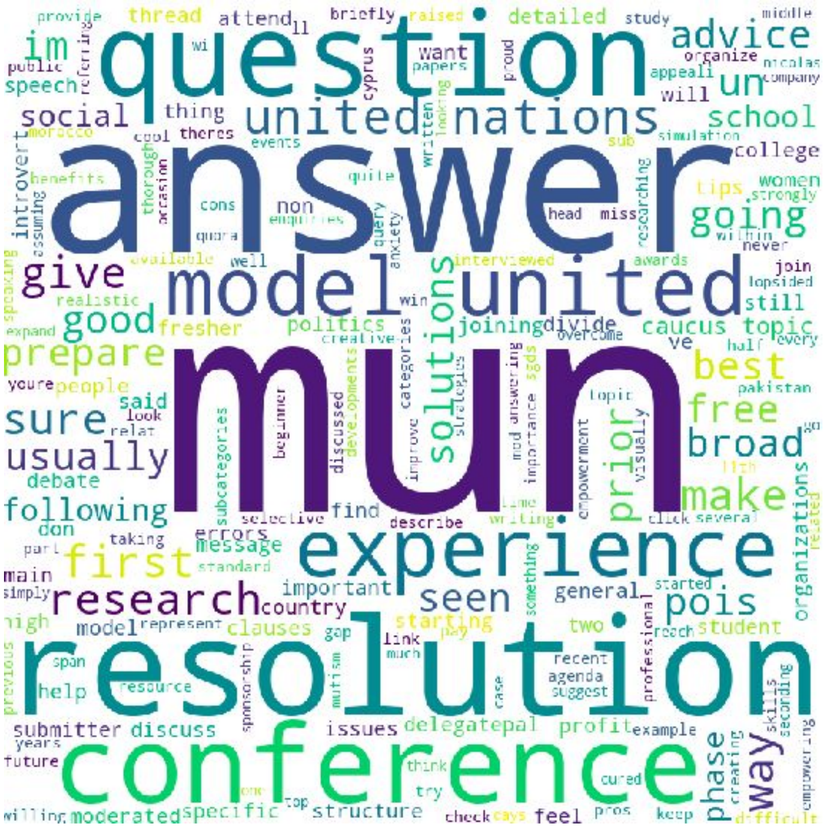


```
my_page = 'https://www.quora.com/profile/Bassim-Eledath-1'
page = urllib.request.urlopen(my_page)
soup = BeautifulSoup(page, 'html.parser')

links = soup.find_all('a', href=True)
list_questions = []
for link in links:
    list_questions.append(link['href'])

my_stats = []
numbers = []
labels = []
for x in soup.find_all('a',href=True):
    if x.find_all('span', {'class' : 'list_count'})!=[]:
        my_stats.append(x.text)

for x in my_stats:
    numbers.append(re.findall('\d+', x ))
numbers = [int(item) for sublist in numbers for item in sublis
for x in my_stats:
    word = ''.join([i for i in x if not i.isdigit()])
    labels.append(word)


# formatting
questions = []
knows_about = []
for x in list_questions:
```

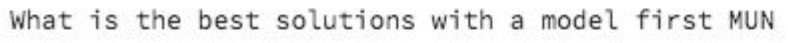
- Used python package BeautifulSoup to scrape data from my profile.
- Parsed and cleaned data and saved questions and answers as .txt files.
- Plotted my Quora stats.



Project 1: Scraping my Quora profile



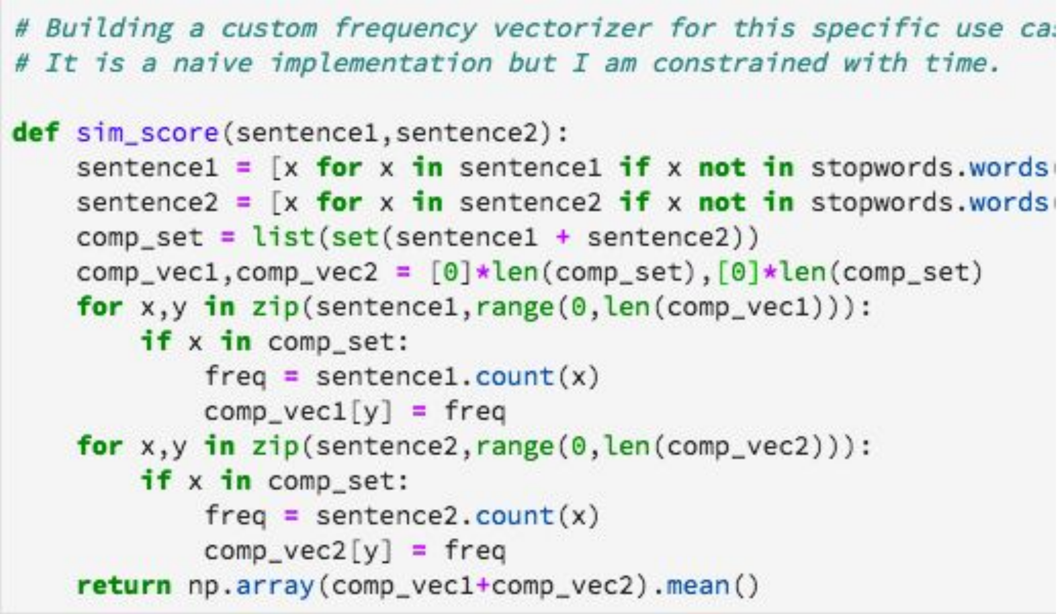
- Plotted wordclouds of my questions and answers (using wordcloud package). Unsurprisingly they were about Model UN (I used to be a fanatic).
- I fitted my scraped questions and answers with textgenrnn, a trained text generation Recurrent Neural Network, and generated sample questions and answers. The results were eerily impressive for the limited data and number of epochs trained.
- One of the generated questions is shown below. You may view all generated questions and answers on my github.



```
What is the best solutions with a model first MUN
```

Project 2: Quora question pairs data

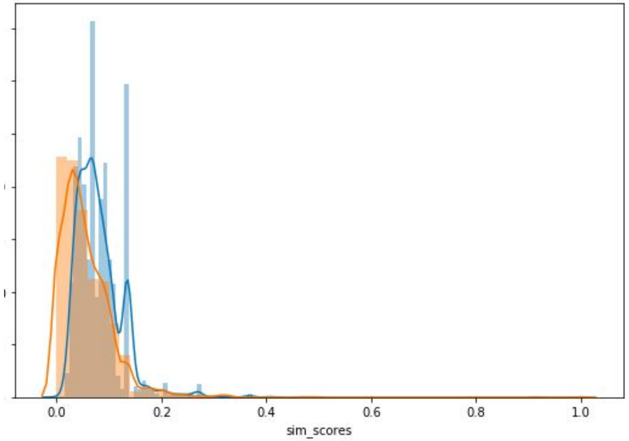
Given that I had limited time, I disregarded the possibility of constructing a machine learning model or using high-end NLP practices. I decided, instead, to create a naive yet quick custom model based on what I thought would best measure question-pair similarity. After some document cleaning and tokenization, I applied my custom function shown below.

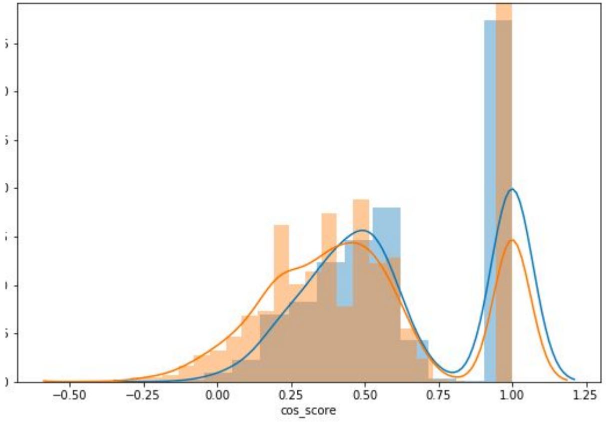


```
# Building a custom frequency vectorizer for this specific use ca
# It is a naive implementation but I am constrained with time.

def sim_score(sentence1,sentence2):
    sentence1 = [x for x in sentence1 if x not in stopwords.words
    sentence2 = [x for x in sentence2 if x not in stopwords.words
    comp_set = list(set(sentence1 + sentence2))
    comp_vec1,comp_vec2 = [0]*len(comp_set),[0]*len(comp_set)
    for x,y in zip(sentence1,range(0,len(comp_vec1))):
        if x in comp_set:
            freq = sentence1.count(x)
            comp_vec1[y] = freq
    for x,y in zip(sentence2,range(0,len(comp_vec2))):
        if x in comp_set:
            freq = sentence2.count(x)
            comp_vec2[y] = freq
    return np.array(comp_vec1+comp_vec2).mean()
```

Project 2: Quora question pairs data

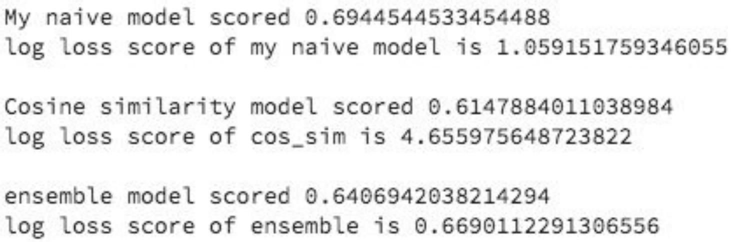
- My function simply takes 2 questions as input, vectorizes them as two lists of word frequencies, and takes the mean of the two output vectors. In other words, my model favors question pairs with many matching words.
- I decided to try a variation of my output where I took the cosine similarity of the two vectors instead of their mean. As show on the right, the graph above (plotting the mean vector values) shows a little more distinction in question pair similarity scores between duplicates and non-duplicates (orange is non-duplicate distribution).
- At this point I suspect both models have high recall, but low precision. They are generally right when they classify a question pair as duplicate, at the cost of having many false positives, as their distributions overlap significantly.



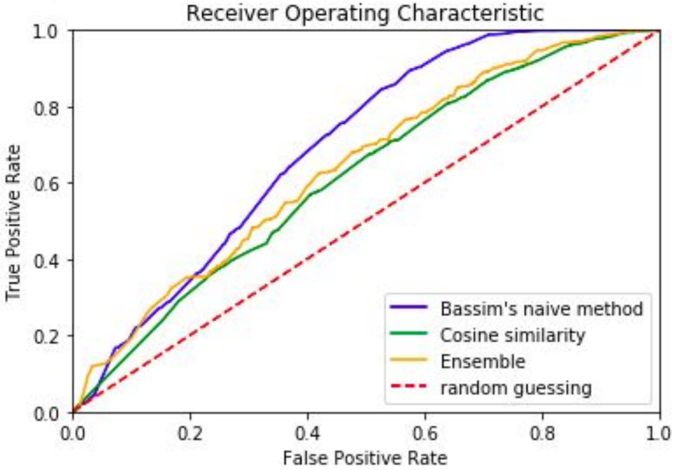


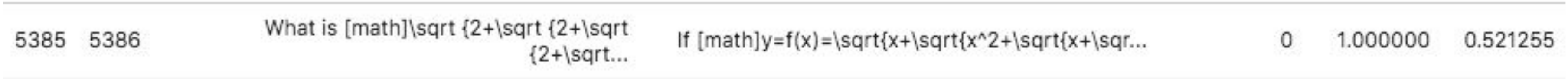
Project 2: Quora question pairs data

- The ROC AUC proves my previous assumption.
- My naive model was biased towards long questions that have the same words (as shown below), particularly the math questions. It does not take context into account.
- The Cosine model on the other hand was biased towards short questions with similar words.
- Since the objective is to minimize log-loss, both models do badly as they are penalized heavily for their confidence in false positive cases.
- As both models are biased towards different question types, I make an ensemble prediction that averages both model predictions and voila, the log loss is now 0.669, which is lower than the ‘words in common benchmark’ on Kaggle.



```
My naive model scored 0.6944544533454488
log loss score of my naive model is 1.059151759346055

Cosine similarity model scored 0.6147884011038984
log loss score of cos_sim is 4.655975648723822

ensemble model scored 0.6406942038214294
log loss score of ensemble is 0.6690112291306556
```





5385	5386	What is [math]\sqrt {2+\sqrt {2+\sqrt {2+\sqrt...	If [math]y=f(x)=\sqrt{x+\sqrt{x^2+\sqrt{x+\sqr...	0	1.000000	0.521255

I want to intern at Quora because...

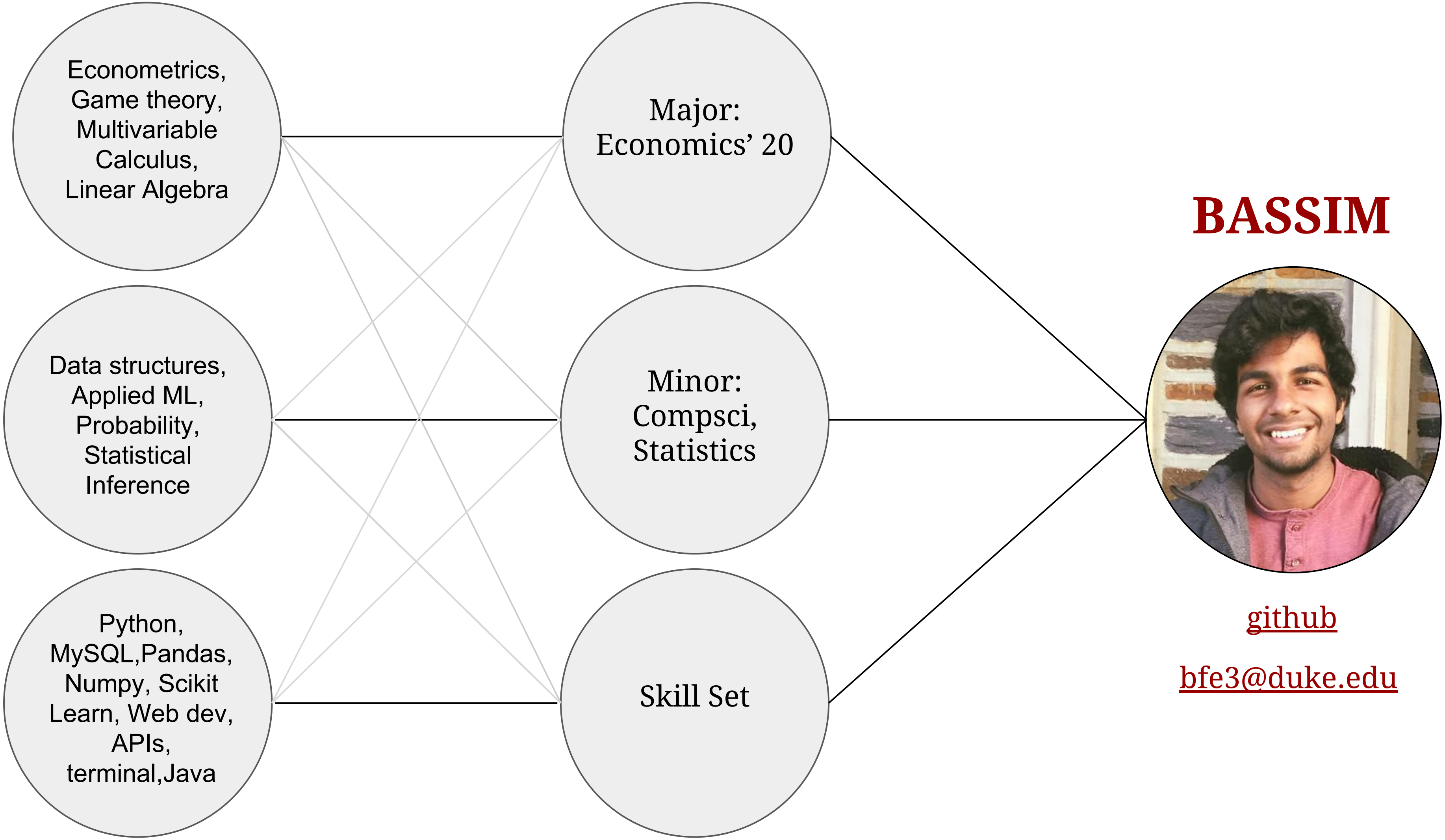
Back in 2017, when I was looking into traction channels to promote my webapp (delegatepal.com), I was primarily targeting Facebook and Instagram. But after weeks of meager organic growth, I turned to other channels. Quora was one of them.

Answering question after question about Model United Nations (which is what the app is about), I realized Quora was the perfect traction channel. I could genuinely help people by answering their queries and in the process direct them to my application to further help them. The analytics confirmed this: Quora is still to this day the biggest source of organic traffic for delegatepal.

This relationship between consumers and producers, in an information exchange platform, is something that I believe is worth further investigating if not refining. At Quora I'd like to better understand this relationship, even if it is at an abstract level, and test assumptions surrounding it using data.

My Résumé

BASSIM



Econometrics, Game theory, Multivariable Calculus, Linear Algebra

Data structures, Applied ML, Probability, Statistical Inference

Python, MySQL,Pandas, Numpy, Scikit Learn, Web dev, APIs, terminal,Java

Major: Economics’ 20

Minor: Compsci, Statistics

Skill Set

github

bfe3@duke.edu

SUPERVISED LEARNING

Home Credit Default Risk Competition (Kaggle)

- Objective: classify whether future loan customers would default.
- Attempted several iterations of preprocessing, feature engineering, model selection and hyper parameter tuning.
- Final submission consisted of an ensemble of XGBoost and LightGBM models - finished in the top 20%

MNIST dataset challenge (handwritten digits)

- Employed various classification models to maximize ROC-AUC.
- Used PCA and linear discriminant analysis and compared results. On my github page.

Hot Dog Not Hot Dog

- Inspired by my favorite episode from ‘Silicon Valley’, built a model to classify images as ‘hot dog’ or ‘not hot dog’.
- Used transfer learning by retraining Resnet50

UNSUPERVISED LEARNING

Wells Fargo Campus Analytics Challenge 2018

- Objective: Segment users into clusters based on information on their carbon footprint.
- Preprocessed messy data: filled missing values, deleted irrelevant columns, created interaction and summation columns that summarized carbon footprint usage.
- Final preprocessed data had one row for each user, instead of 27 as in the original dataset.
- Trained 5 clustering models - k-means, hierarchical, mean shift, gaussian mixture and spectral clustering.
- Created an evaluation script that selected best model based on 3 evaluation metrics - cluster normality, mean distance and Variance Ratio Criterion.
- Used best model (mean-shift) to segment users and construct eco-friendly product recommendations based on cluster properties.

FINANCE

CryptoBabes - a fintech collaboration

- Queried cryptocurrency data using scraping API
- Feature engineered columns and plotted results using web framework (made with dash).
- Did sentiment analysis of respective cryptocurrencies using twitter and reddit comments (queried with API) for the time period of the data.
- Compared sentiment analysis results with stock. results, to see if there was any meaningful correlation.

DATA SCIENCE CONSULTING

Freelance Data Science Consultant at AWA Digital

- Remote internship. My client: Canon.
- Query Canon's data using Google Analytics and SQL.
- Construct hypotheses worth testing such as "Registered vs Unregistered users at checkout".
- Conduct analysis using Python and summarize findings as presentations.
- Provide recommendations based on conducted analysis.

COMPUTER VISION & DEEP LEARNING

Data Science Intern at Data Prophet (Summer'18)

- Objective: (a) identify detective automobile manufacturing parts (b) create a model that identifies the angle on dials of metal plates (the angles made up the serial number of the respective plate).
- For (a) I had to construct a pipeline that retrained an existing NN model (VGG-16) to classify defective manufacturing parts.
- Created a script that took labelled images as input and produced augmented images to increase training set size and model accuracy.
- For (b) I worked on using testing two models - facial keypoint detection and object detection.
- For the former I first labelled key-points on images and then trained the keypoint detection model to recognize the key-points and classify them as point1, point2 etc.
- I used Tensorflow's object detection API to classify the points as square objects instead of points, but the keypoint detection gave better results.
- Once I had the points, I trained a neural network with points as input and angles as output.