Attribute Model:
Problem statement: Given the product review of a user, identify what attribute of the product he/ she is talking about.
Tools & Models used: EDA, Topic modeling, LSTM sequence tagging, Tensorflow, Python.
Description:

1. The main problem while solving this problem statement was that there was no labelled dataset available for the attribute tagging. Even though we had a huge amount of user reviews they were not labelled with attributes.
2. In order to create labelled dataset we used unsupervised learning method. We extracted all the nouns from these reviews, which became our candidate attributes and we trained a topic modelling algorithm i.e NMF to identify the topics. Using this NMF model we were able to identify the different attributes user is talking about.
3. Problem with this model was the developed model was domain specific. For example: if the model is trained on reviews of mobiles and laptops, it won't be able to identify the attributes of automobiles or furniture.
4. Hence in order to overcome this problem we labelled all our dataset using NMF model and trained a LSTM model using this dataset. LSTM model is domain independent model.
5. We used cross entropy loss and Adam optimizer in LSTM model.

Review ranking model
Problem statement: When a user submits a query, sort the candidate reviews according to the relevance of these reviews to the query.
Tools & Models used: EDA, Microsoft QA dataset, LSTM model, Triplet loss, Similarity, Python.
Description:

1. We used the dataset which was already available.
2. We developed a LSTM model which used a type of contrastive loss called Triplet loss function to calculate the cost and we used Adam optimizer in order to find the gradients and update the weights.
3. Triplet loss involved 3 latent representations: A. Anchor representation: we took question embedding as anchor. B. Positive representation: We took correct answer and represented it as positive latent vector. C. Negative representation: We also had wrong answer for any given question which we represented as negative latent vector.
4. Using Anchor, positive, negative latent vectors we calculated the triplet loss and build review ranking model.
5. During the inference we passed user query and candidate reviews to the LSTM model and got the corresponding context vectors. We sorted candidate reviews based on similarity between query and candidate reviews.

Classification model:

Problem statement: Ranking all the reviews for a particular query is computationally expensive. How would you reduce the computation timing
Tools & Models used: EDA, LSTM/Regression Classification model, Tensorflow, Elasticsearch, Python.
Description:

1. We labelled all our reviews with corresponding category.
2. Whenever a user makes a query we will find out the category of the query and we will use only those reviews whose category is same as that of query category.
3. In order to do this we used elasticsearch and category classification model.
4. We scrapped the data from different websites. While crawling we looked at the meta data of the pages and figured out the category of the page.
5. Using this labelled dataset, we trained a LSTM sequence classification model using cross entropy loss and Adam optimizer.
6. Whenever a new review has been inserted to the DB we run our category classifier on it and we will store the category of corresponding review.
7. Whenever user makes the query, using elasticsearch we will get only those reviews whose category is same as query category and pass it to review ranking model.
8. The output of the review ranking model will be the response for user query.

Sentiment analysis
Problem statement: Identify the sentiment associated with each review.
Tools & Models used: EDA, LSTM sequence classification, Tensorflow, Python
Description:

1. We used IMDB movie review sentiment dataset.
2. We trained LSTM sequence classification model for sentiment analysis.
3. Given a review we pass this review to the LSTM. We will get the context vector for that particular review
4. Using this latent representation of context vector we do classification using fully connected layer.
5. We used binary cross entropy for loss calculation and We have used Adam optimizer for finding out the gradients and back propagating the errors and updating the weights.
6. We also used other techniques such as logistic regression, random forest, naive bayes, SVM, xgboost classifier and ensemble models.
7. LSTM sequence classification model performed well over other models.
8. We also used OpenAI sentiment neuron to compare our model results.

ElasticSearch:

Problem statement: Review ranking model takes lot of time to rank all the reviews of a particular category to send response to user query. How do you solve this problem.
Tools & Models used: ElasticSearch, Python
Description:

1. Elasticsearch is a search engine based on the Lucene library. It provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents.
2. Since review ranking model was taking lot of time we used elasticsearch to select few candidate reviews from a particular category pool.
3. We will pass user query to the elasticsearch and specify the number of documents we are interested in to answer the user query.
4. We can not use the results of elasticsearch because elasticsearch doesn't promise the semantic similarity.
5. Hence we used our review ranking model on top of elasticsearch in order to refine the order of response reviews.
6. We installed elasticsearch and used elasticsearch pip package to communicate with the elasticsearch.
7. We also saw how to create and index documents in elasticsearch.

Scraper:
Problem statement: In order to recommend reviews to user when he submits a query about a particular product, we need some reviews to recommend. How do you collect these reviews from different websites.
Tools & Models used: Python, scrapy.
Description:

1. We wanted to write scarper which will scrap all the reviews from different websites.
2. We wrote a scraper for amazon. Which will crawl through different pages and products of amazon and fetch all the meta data and reviews.
3. We used this data for training our classification model.
4. We also used this review dataset to give response to user queries.

We have also used

1. Scheduler
2. flask