Arjun Sarup

Rajandeep Singh

Sajel Shah

Vineet Yellu

Yash Bhate

InstaPromote Final Report

Project Overview

Capitalizing on the rapid rise of advertising spend on social media (Instagram in particular), our offering suggests the optimal brand influencers for varied commercial use cases. Through image recognition, sentiment analysis and topic modelling we provide an in-depth analysis of any users profile leading to improved brand visibility. Our elegant and intuitive user interface displays the results in a dashboard filled with topic recommendations, demographic analysis and follower sentiment profiles. The like prediction model incorporates varied metadata from each instagram influencer's profile (including followers, following, number of posts, time of posts, influencers category) to improve accuracy. In addition, we leverage image object detection to further improve the accuracy of our like predictor and suggest object to include in your promotional post. This model is robust across various instagram profiles and can handle images, videos and text media-types.

Value Proposition: Why It's Useful

Digital media has quickly grown to become the largest advertising focus for firms (as measured by spend); spend on social media advertising has risen to 58B in 2018 (up 21% from 2017). This rate of growth is faster than the increase in spending on search ads, the largest source of digital advertising spend at the moment. Furthermore, the opportunity to develop the image and reputation of your brand and foster discussion about your offerings suggests an unrealized benefit of social media advertisement over more conventional forms like search ads. The higher level of user engagement and relevance of social media advertisement also suggests a larger return on investment than for other advertisement.

Instagram is the preferred social network for individuals under 30 and has the fastest growing user base with over 1B MAU. Digital advertising spend on Instagram is growing at 5% Q-o-Q, even faster than Facebook at 3.14%. Given the platform's visual nature and high user engagement rate, Instagram is an especially valuable and effective social media marketing tool for many industries - 98% of fashion brands are active on Instagram.

By suggesting optimal brand influencers for their specific use cases, we allow companies to better balance the cost and reach of their advertising campaign by prioritizing influencers that maximize their exposure per dollar spent. Because we find the optimal influencers for an industry, companies can increase ROI and possibly reduce advertising cost by finding lesser known influencers that can advertise at lower cost with similar reach to target audience.

How It Works

The landing page, any user of InstaPromote will see the following advertising domains: fitness, fashion, tech., entertainment. Users can either:

- request information about a specific influencer in our current database through the search bar at the bottom, or
- click on any of our displayed domains to see the potential influencers they can consider for that domain.

Clicking on an influencer's profile will display a curated portfolio of relevant information about that influencer, including:

- their cover photo,
- the average sentiment response to their posts,
- list of common topics they have addressed through their posts,
- like predictor that, based on the inputs the user gives (day of the week, hour, image), will predict the likes the influencer gets for any specific post.

To garner the data for the aforementioned features, we modified a pre-existing instagram-scraper so that it could **scrape** the top 10 comments in addition to scraping the image and media-metadata (post timestamp, caption, likes) for each post. For the **average sentiment response**, we used **Google Cloud's NLP API** with a timeout modification so that we didn't overflow the request limit of said API.

For the list of common topics, we carried out **topic modelling** for each and every influencer and used PYLADAVIS to generate a wordmap, the screenshot of which we display on the "portfolio info." page described above. Finally, the like-predictor itself was trained using the following features: day of the week (one-hot-encoded), epoch, the objects in the image (one-hot-encoded) and the previous likes for the past ten posts. We classified the objects in the image using YOLO, "an object detection model that passes a single neural network over the entire image. This network divides the image into regions and predicts bounding boxes and probabilities for each region" (from the README).

Each celebrity has their own pre-trained like-predictor that is updated in the background with new data every other day. Having a pre-trained like predictor allows us to save time and

make real time predictions for the potential posts that the user wants to test. Based on the number of likes an influencer receives for any given post, and the sentiment of their posts, potential users can easily track and compare across influencers to select the ones they need for their product.

Our Journey

In the first iteration, we started off with a broad, ambitious goal - to use an available, alternate dataset to optimize decision-making in solving real-world problems. As students in constant search of datasets to play around with, publicly available resources were either inadequate or too curated for out purposes. After probing our day-to-day interactions, we realized social media presented a treasure grove of information and building pipelines through a scraping tool would allow us to apply sophisticated models akin to the ones we learned in class. As we looked at different social media tools, we realized scraping through LinkedIN could pose legal ramifications and Tumblr and Pinterest did not truly have a universal reach. We used Instagram as it was immensely among our age group and there existed no industry standard tool to easily get the data we required - the API had severe shortcomings to arrive at any verifiable conclusions.

UI

For the project, we created a website for the user to interact with our product. Our website is build using HTML5 and linked with the Python backend using Flask and WTForms. We initially intended the product to train models on demand at a click of a button for the like predictor, which was very feasible and working perfectly initially. However, as we added YOLO object detection to our project, the training time increased drastically. This made us reconsider our entire approach and we decided to pre-train all the models, and only to predict on the fly. In a real world scenario, we would retrain these models every day or so.

Web Scraper

The first issue we had to solve was developing a web scraper. Given the lack of an industry standard webscraper for Instagram we looked at open-source code on Github to survey existing tools. One tool in particular seemed to perform some of the tasks we required. Considering our project relied heavily on comments, we modified this scraper heavily to fit our needs. We re-wrote GraphQL for the scraper so it would download only a fixed amount of comments per post (this was essential as posts with too many comments crashed the scraper), and a fixed amount of comments per user. We also added the ability to download 15 images at the same time along with time-outs to make the scraper fast and robust. After building this for one user, we wrote a complex script and pipeline to perform this task for multiple users and send that data to a repository.

Like Prediction Model

Initially, after getting the data, we performed some EDA and noticed a high correlation between the number of likes for a post and the average number of likes for the most recent posts. For the MVP, we mainly used this feature aka, we averaged the previous likes of the post and one-hot-encoded other basic features like the day of the week that the post was made. We then trained a simple linear regression model on the metadata of the data we had fetched. This had a reasonably high accuracy, but we felt the need to include more features.

One such feature was the epoch of each post, which functioned as an unbiased (aka, not dependent on any specific timezone) observer of time. We assumed that the more an influencer posts, the more followers you will get over time, so there must be some correlation between the epoch and the number of likes that a user will get. Indeed, our assumption was fairly accurate,

After learning about CNNs and DNNs in lecture, as well as transfer learning, we realized that we could use an object detection model and actually integrate the content of the actual post besides its metadata into our like predictor. Seeing how YOLO v3 has been so popular with object detection, we used a pre-trained YOLO model and classified the objects in all the images of each influencer. We saved the objects predicted for each image in a .csv file, which we then merged with our original dataframe. Using one-hot-encoding for the objects, we now had features that went beyond the immediately available data and actually used the image itself. This allowed us to improve our testing accuracy across the board - at least by 5%

Some features that we abandoned along the way included: numComments for the previous 10 posts (it didn't improve accuracy since there isn't a strong correlation between the number of comments and the number of likes for an image), numPersons (the number of people in the image --> distinguishing in terms of age matters more than the number of people in an image because, based on our EDA, a baby and her mom tend to get higher likes than just two adults), hashtag popularity (would require scraping the web for every hashtag in every post for every influencer and would occur after the initial data has been generated → we'd effectively be peaking into the future while we are supposed to be operating in the past).

Finally, we considered using MemNet, a hybridized pre-trained CNN that was constructed by a pHD candidate at MIT to predict the memorability of an image. Since there would theoretically be a high correlation between memorability and numLikes, we thought that this final feature would make our accuracy jump up across the board for every influencer. However, since MemNet is deployed on Caffe, we had to translate it to Keras, which caused a lot of issues related to dimensionality transfer and missing layers (Keras doesn't have the

"norm" layer and other layers Caffe does). We hope to integrate MemNet in the future as we continue iterating upon the product.

Sentiment Analysis

A good metric for measuring if an influencer is adequate at propagating a product forward is analyzing the sentiment of user reactions to the influencer's post. In our MVP, we scraped the top ten most popular comments for each of the last hundred posts posted by the influencer. Then, we ran a sentiment analysis algorithm on this subset of comments and posts marking each comment as positive, negative, or neutral. We used Google's Natural Language API for sentiment analysis, which helped us tokenize slang words and emojis in the comments. For our final product, we optimized our code to stop every few hundred posts and restart again to avoid the maximum usage limit of Google's API. We reported the overall sentiment of users reacting to the influencer as a ratio of number of positive, negative, or neutral comments over the total number of comments we analyzed. Using the last hundred posts of an influencer gave the best indication of how users are reacting to this celebrity in the status quo. The top comments per post are the most indicative of the general sentiment of the users because those are the comments that were most engaged by users. This method stayed true to ratios that were calculated using more comments and more posts because the relative sentiment of users reacting to a particular influencer converges over a period of time.

Topic Modeling

Topic Modeling is a feature that we thought of while training the initial like predictor. We realized that we had the metadata for every influencer, aka, the data related to every post the user had made before. So, using topic modelling, we could get a list of the most common topics the user had explored before through their captions.

Besides this being a tool for users/advertisers to observe prior engagement with a related product in an influencer's portfolio, we are also working to implement a matching algorithm that could allow advertisers to search for their product on our website and we would then display those influencers who have engaged with that product before.

We would theoretically do this by storing the topics in a JSON/dictionary, and then iterating through that dictionary to search for matching influencers whenever someone enters a product they want to advertise.

Links

Github Link: https://github.com/arjunsarup1998/team_30_data-x
Google Drive Links: Link 1, Link 2
Instagram Scraper: https://github.com/rarcega/instagram-scraper