# Stream Me

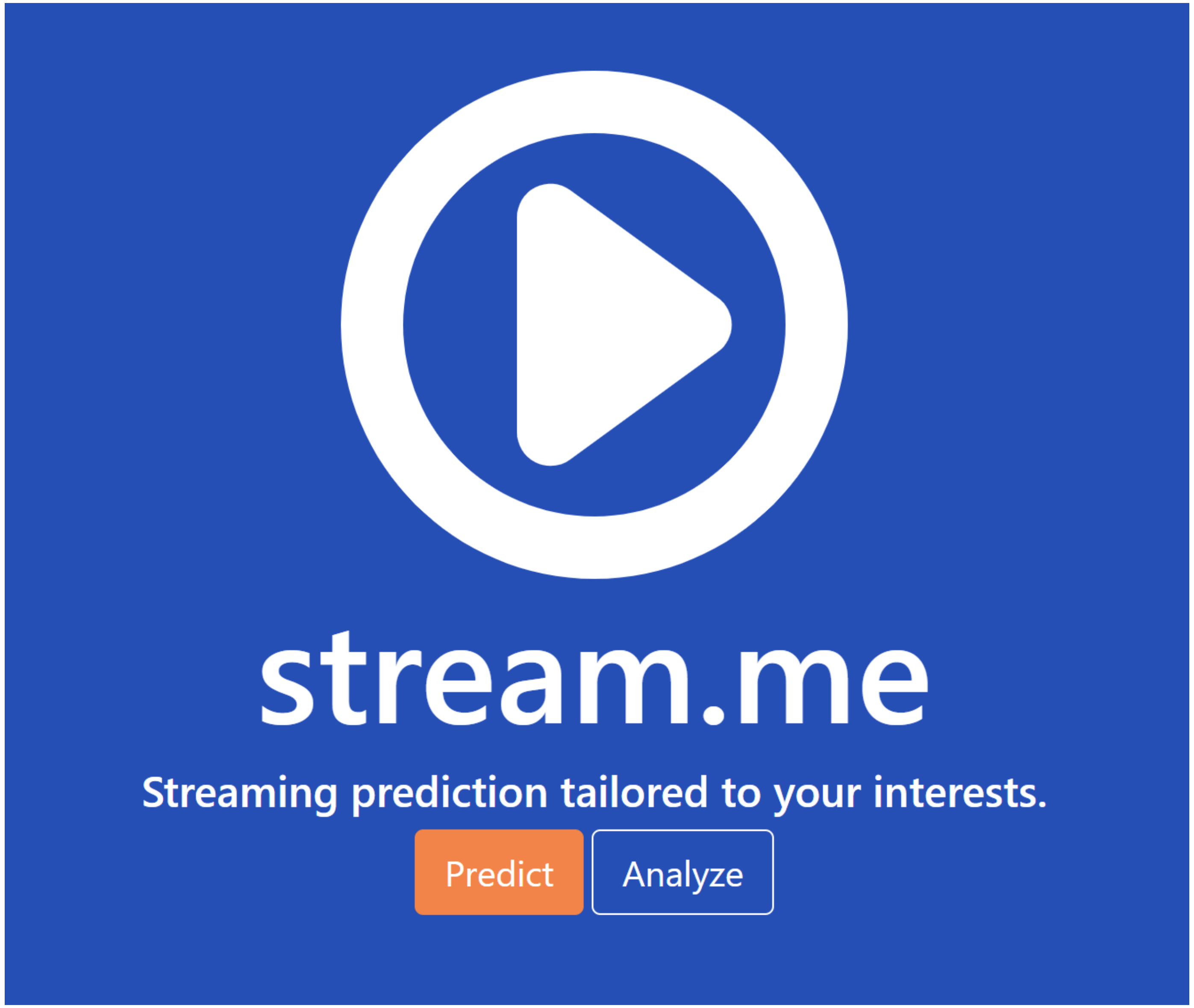
BY ; Timothy Lucas, Robert Hanlon, Kyna Thorberg, Nathanuel Martin

## Outline

- **Data Story** - Can we predict the best streaming service based on movie and show data?
- **Interactive Flask Application** with User Interview, Prediction Algorithm, User Lookup, Customized Visuals
- **JS Library not covered** – Dynamic Type Ahead Inputs using Jquery and RESTful API
- **Dataset** - 16,744 Movies and 5,611 Shows from 4 Streaming Services (Reelgood.com, Kaggle)
- **User-driven Interactions** (Interview and lookup)
- **4 views and API Doc** (Landing Page, Visualizations, User Interview, Dashboard)

## Do you know which streaming service is right for you?

- Our group came together to answer this question for you all because you probably have one of the top streaming services (Netflix, Amazon Prime Video, Hulu, Disney+) but do you really enjoy the tv shows and movies that they offer?
- We created a site that leverages Movie and Tv show data, particularly ratings, genres, and directors to give our user the tool he or she needs and to characterize each service according to the content preferences of the user.
- We used a combination of tools to build the site the first part was to build out a Python Flask application on the server-side with a PostgreSQL database on the back end, with D3 JavaScript library

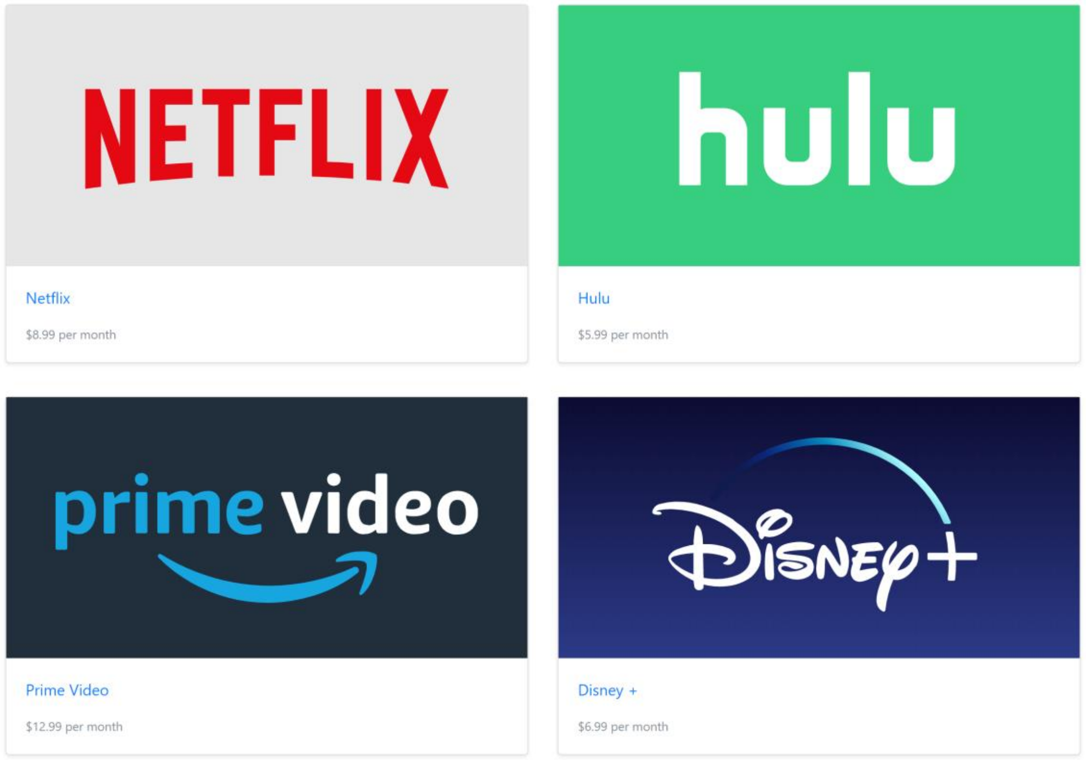

stream.me

Streaming prediction tailored to your interests.

## Landing Page

- Predict - User interview to help filter the original data set then apply an algorithm that predicts the best service.
- Analyze - Visualize the entire dataset or filter based on the user interview results.

## About Page


Netflix

$8.99 per month

Hulu

$5.99 per month

Prime Video

$12.99 per month

Disney +

$6.99 per month

- Learn about the 4 streaming services analyzed by our application.
- Find Logos, subscription Prices and links to each respective service page.
- Abstract explains the purpose of the application.

## Analyze Page

- The Analyze page loads with a dashboard style visualization for all available data.
- Visualizations include Scatter and Bar Plots that represent:
  - Movie Counts
  - Show Counts
  - Price Comparisons
  - Avg IMDB and Rotten Tomato Ratings
- Existing User Lookup Tool (for those that have already completed the interview.

# Predict Page

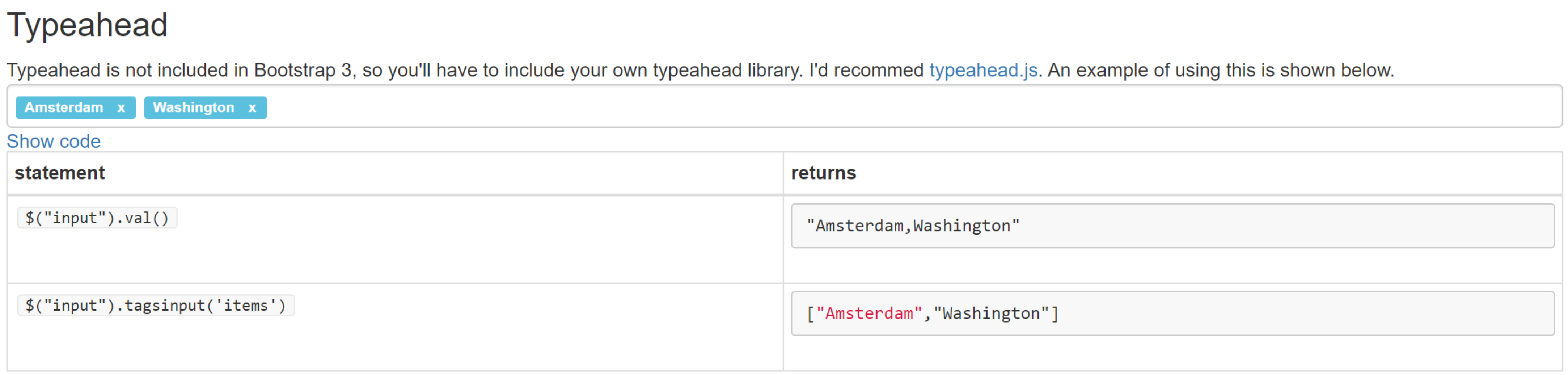
- The predict page is a user driven interview at its heart.
- This page uses a variety of input types to collect bits of data that represent the user preferences.
- When posted this form will create and store a new user profile in our DB.
- The user profile consists of an email address, a randomly generated unique 5-character code, and their answers.
- Then we'll pass that user info through our **Prediction** algorithm and redirect the user to their customized Analyze page with their **Prediction** result.

## How does the prediction work?

- First, we filter and prepare the movie and show data based off the user's 7 interview answers.
- After the data is filtered, we divide what's left into our 4 streaming services.
- We count, we count a few more, we average here, and we average over there.
- Then, we divide those results into 9 categories each with its own respective weight.
- We normalize all 9 categories so they can be equally compared.
- Finally, we multiply the normalized score to the appropriate weight and add all the scores together.
- When it's all over, the highest score wins!

# Jquery – Dynamic Inputs

- https://bootstrap-tagsinput.github.io/bootstrap-tagsinput/examples/
- Allowed us to use the data available on our API Endpoints to include a dynamic type ahead input on the interview.
- Tagging style input similar to those found on blogs or forums.

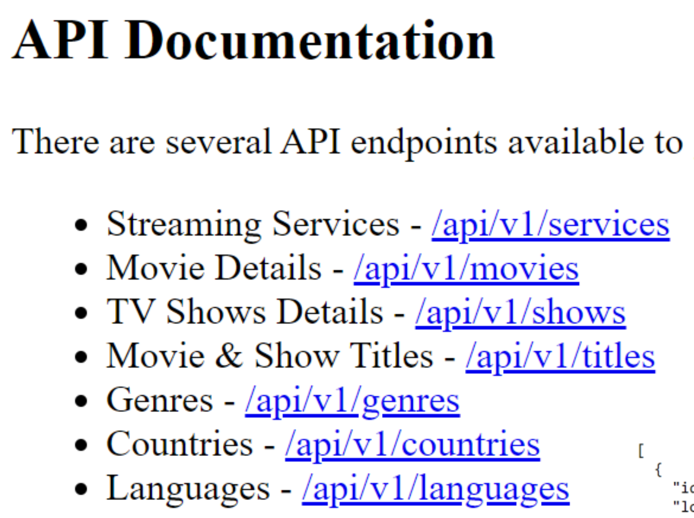


## Typeahead

Typeahead is not included in Bootstrap 3, so you'll have to include your own typeahead library. I'd recommed typeahead.js. An example of using this is shown below.

| statement | returns |
|---|---|
| `$("input").val()` | `"Amsterdam,Washington"` |
| `$("input").tagsinput('items')` | `["Amsterdam","Washington"]` |

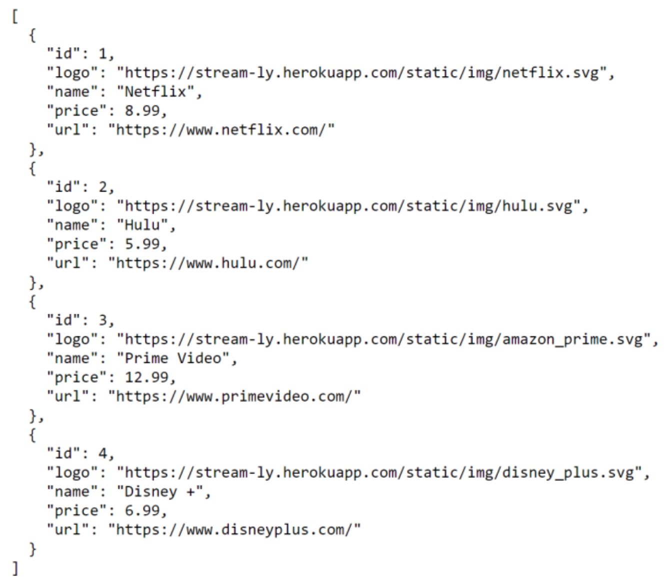
# Restful API



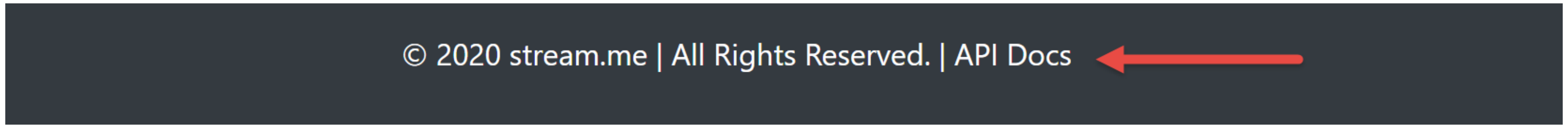
## API Documentation

There are several API endpoints available to grab our raw data sets.

- Streaming Services - /api/v1/services
- Movie Details - /api/v1/movies
- TV Shows Details - /api/v1/shows
- Movie & Show Titles - /api/v1/titles
- Genres - /api/v1/genres
- Countries - /api/v1/countries
- Languages - /api/v1/languages



```
[
  {
    "id": 1,
    "logo": "https://stream-ly.herokuapp.com/static/img/netflix.svg",
    "name": "Netflix",
    "price": 8.99,
    "url": "https://www.netflix.com/"
  },
  {
    "id": 2,
    "logo": "https://stream-ly.herokuapp.com/static/img/hulu.svg",
    "name": "Hulu",
    "price": 5.99,
    "url": "https://www.hulu.com/"
  },
  {
    "id": 3,
    "logo": "https://stream-ly.herokuapp.com/static/img/amazon_prime.svg",
    "name": "Prime Video",
    "price": 12.99,
    "url": "https://www.primevideo.com/"
  },
  {
    "id": 4,
    "logo": "https://stream-ly.herokuapp.com/static/img/disney_plus.svg",
    "name": "Disney +",
    "price": 6.99,
    "url": "https://www.disneyplus.com/"
  }
]
```



© 2020 stream.me | All Rights Reserved. | API Docs

- API Docs - In the footer of the site you'll find a link to our API Doc.
- Endpoints - There are 7 endpoints that each display a different set of JSON. These were used for our visualizations and made available to the public.

## Tools Used

### Client-Side

- HTML, CSS
- Bootstrap
- JavaScript, Jquery
- D3
- Plotly

### Server-Side

- Python
- Flask
- SQLAlchemy
- PostgreSQL
- Heroku

# Missed Opportunities

- There was more data we could have gathered if we had more time:
  - More services, actors & actresses, missing show data, etc.
- We wanted to create a map visualization of how many movies, shows, were available by country.
- We wanted to make a visualization to show how many countries and languages were in the dataset
- Our visualizations should have filtered according to the user data when predictions were displayed.
- Prediction results could have been emailed to the user.