



# Streamer Game Recommender

**Natasha Borders**  
**Jeremy Chow**  
**Randy Macaraeg**



A close-up, slightly blurred photograph of a smartphone lying on a dark wooden surface. The phone's screen is lit up, showing the Twitch mobile app interface with a purple header and various icons. In the background, a portion of a white keyboard is visible. A white diagonal line separates the text area on the left from the image on the right.

1.

# WHY STREAMERS ON TWITCH?

# 560 billion

Minutes watched in 2018

# 1.2 million

Average number of concurrent users

# 4.4 million

Unique monthly streamers

“

Choosing a game (or category) to stream on Twitch is one of the hardest choices to make and the one that will impact your success more than anything else.

-Mark Longhurst of The Emergence,  
In “How to Choose What Game to Stream”



## How do we help streamers?

**Goal:** Pick the best game for growth

- ▶ **Retain** existing viewer base
- ▶ **Attract** new viewers and subscribers
- ▶ **Explore** new but familiar content

Meet **Melany**, a new Twitch streamer!



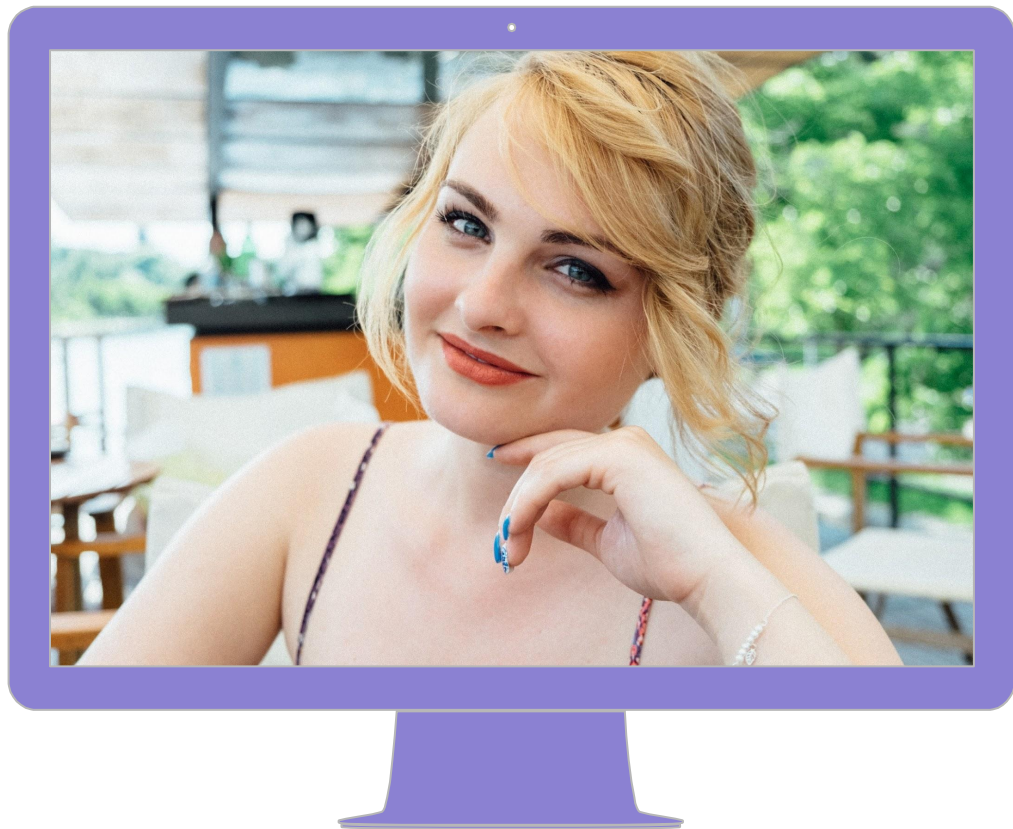
**Needs to know:**

**Should she stream  
her favorites?**

**Which content will  
result in more views?**



What about **Alex**, who already streams?



**Needs to know:**

**What is the next big game her existing viewers will love?**

**What other games should she try to keep her momentum going?**

<https://twitch-streamer-recommender.herokuapp.com/>



# 2.

## DATA PROCESSING



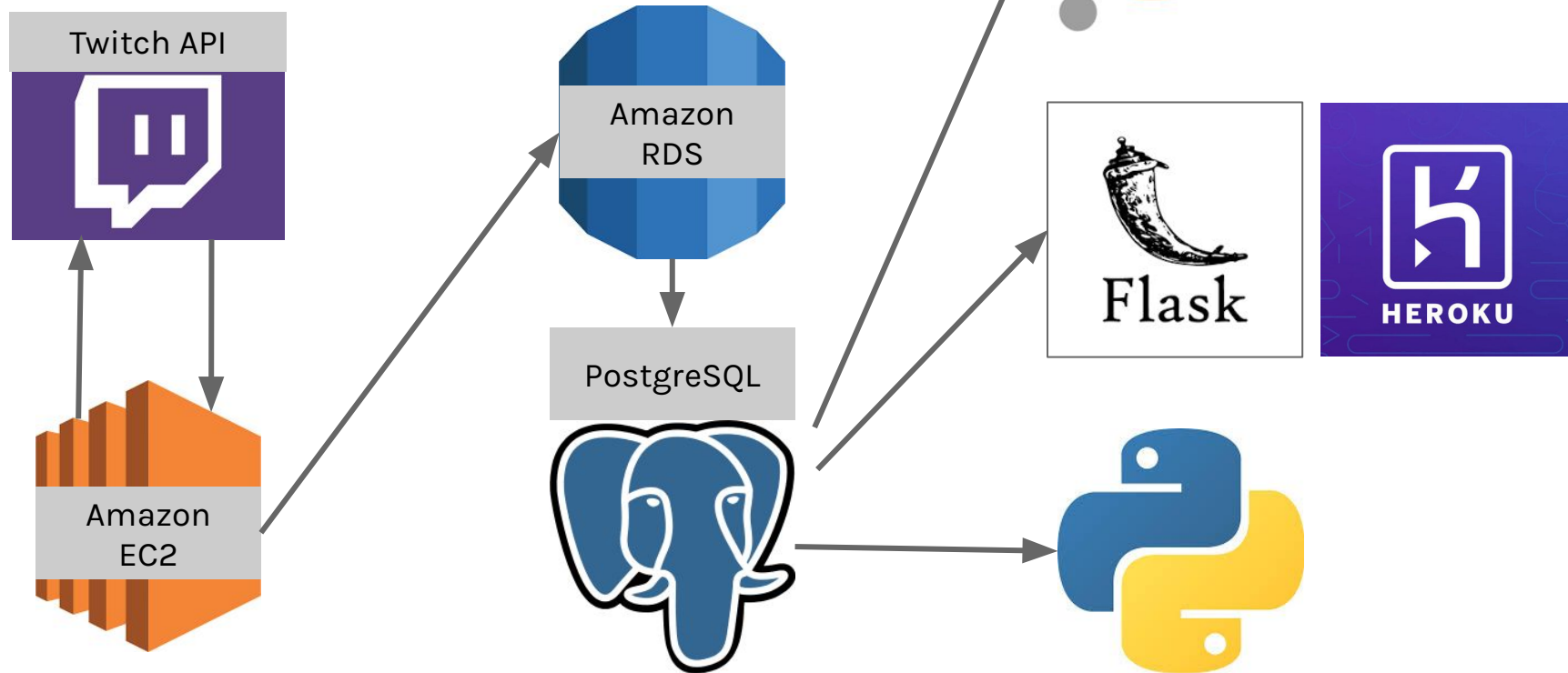


## The Data

- ▶ 40 days worth of **aggregate** channel and game data from Sully Gnome Twitch analysis website
- ▶ 7 days of **streamer-level** data directly from Twitch using the API and Amazon RDS



# Building the **Data Pipeline**



# 3.

## The Recommender





## Framework: Surprise

**USER:**



**GAME:**



**RATING:**

**1-5 Range  
Success Metric**



## Winning Models

- ▶ **FAST** - can keep up with constant inflow of data
- ▶ **LOW ERROR** - keep getting smarter over time

Algorithm	Test RMSE
SlopeOne	0.58
BaselineOnly	0.79
KNNBaseline	0.89



# 4.

## When to Stream?





## Long Short-Term Memory Neural Network (**LSTM**)

- ▶ **Long-term memory** allows model to quantify growth over time, seasonal trends
- ▶ **Short-term memory** allows model to take into account recent/daily trends
- ▶ **Ability to ‘forget’** allows network to look back further without speed penalty of other networks (RNNs)



## LSTM INPUTS

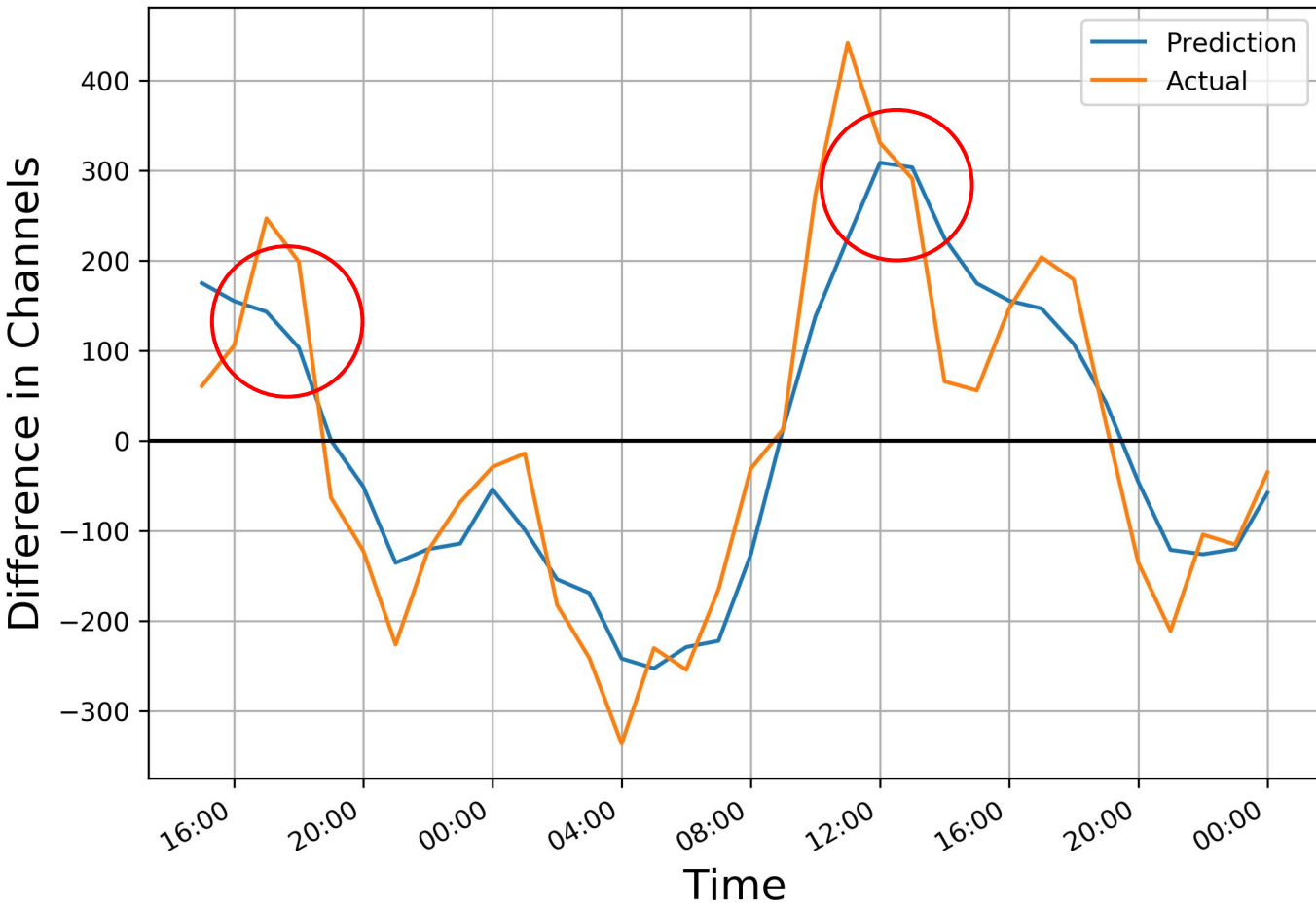
- ▶ **Input “Lag”:** Looked at entire last week of data (168 hours behind)
- ▶ **Model Lookback:** Looked at the last 24 model states
- ▶ **Predictions:** Predicted next week of traffic in order to make recommendations (168 hours ahead)



## **LSTM** Metrics

In Sample RMSE	Test Sample RMSE
10%	10%

Difference in Channels vs. Time (League of Legends, 5/7-5/8/2019)



# 5.

## Growth Score







## Growth Score

- ▶ Order recommendations by growth potential
- ▶ Quantify cult following / monopoly of top streamers
- ▶ Take into account individual game success



## Growth Score

$$\begin{aligned} \text{Score}(\text{game}, t) = & C_0(\text{Recommender score}) + C_1 \frac{\frac{\text{num\_viewers}(\text{game}, t)}{\text{num\_channels}(\text{game}, t)}}{\frac{\text{num\_viewers}(\text{game}, t - 24 \frac{\text{hours}}{\text{day}} * 7 \frac{\text{days}}{\text{week}})}{\text{num\_channels}(\text{game}, t - 24 \frac{\text{hours}}{\text{day}} * 7 \frac{\text{days}}{\text{week}})}} + \\ & C_2 \frac{\text{num\_viewers}(\text{game}, t)}{\text{num\_viewers}(\text{most\_popular\_game}, t)} + C_3 \frac{\text{num\_viewers\_of\_top\_8\_streamers}(\text{game}, t)}{\text{num\_viewers}(\text{game}, t)} \end{aligned}$$



## Growth Score

$$\begin{aligned} \text{GROWTH} &= \text{Streamer Affinity} \frac{\frac{\text{num\_viewers}(\text{game}, t)}{\text{num\_channels}(\text{game}, t)}}{\frac{\text{num\_viewers}(\text{game}, t - 24 \frac{\text{hours}}{\text{day}} * 7 \frac{\text{days}}{\text{week}})}{\text{num\_channels}(\text{game}, t - 24 \frac{\text{hours}}{\text{day}} * 7 \frac{\text{days}}{\text{week}})}} + \\ \text{SCORE} & C_2 \frac{\text{num\_viewers}(\text{game}, t)}{\text{num\_viewers}(\text{most\_popular\_game}, t)} + C_3 \frac{\text{num\_viewers\_of\_top\_8\_streamers}(\text{game}, t)}{\text{num\_viewers}(\text{game}, t)} \end{aligned}$$



## Growth Score

**GROWTH  
SCORE** = **Streamer Affinity** + **Game Growth**

$$C_2 \frac{\text{num\_viewers}(\text{game}, t)}{\text{num\_viewers}(\text{most\_popular\_game}, t)} + C_3 \frac{\text{num\_viewers\_of\_top\_8\_streamers}(\text{game}, t)}{\text{num\_viewers}(\text{game}, t)}$$



## Growth Score

$$\begin{aligned} \text{GROWTH} \\ \text{SCORE} &= \text{Streamer Affinity} + \text{Game Growth} \\ &+ \text{Game Popularity} + C_3 \frac{\text{num\_viewers\_of\_top\_8\_streamers}(game, t)}{\text{num\_viewers}(game, t)} \end{aligned}$$



## Growth Score

$$\begin{aligned} \text{GROWTH} & \\ \text{SCORE} & = \text{Streamer Affinity} + \text{Game Growth} \\ & + \text{Game Popularity} + \text{Market Penetrability} \end{aligned}$$





## Future Work

- ▶ **Scale** the model to work with Twitch streaming data in realtime
- ▶ **Train** the recommender algorithms and improve their accuracy with more data
- ▶ **Integrate** growth score into app

**THANK YOU!**

Find our app: <https://twitch-streamer-recommender.herokuapp.com/>



**Natasha Borders**



natasha@natashaborders.com



natashaborders



natashaborders



**Jeremy Chow**



jeremyrchow@gmail.com



jeremy-chow



jeremyrchow



**Randy Macaraeg**



randy.macaraeg@gmail.com



randysm



rmacaraeg

# 6.

## APPENDIX



# OUTLINE

- ▶ Overview
- ▶ Data Analysis & Process
- ▶ Modeling & Results
- ▶ Recommender Demo
- ▶ Future Work

# Result - PostgreSQL Database

## **stream\_data**

- stream\_id text
- user\_id text
- user\_name text
- game\_id text
- stream\_type text
- title text
- viewer\_count int
- started\_at timestamp
- language text
- time\_logged timestamp

## **game\_information**

- game\_id text
- game\_name text
- Pic\_url

## **game\_genres**

- game\_name text
- game\_genres text





# Recommender WEBSITE

Built on Python and  
Flask, hosted on Heroku:

[https:](https://twitch-streamer-recommender.herokuapp.com/)

[//twitch-streamer-recommender.herokuapp.com/](https://twitch-streamer-recommender.herokuapp.com/)

The screenshot displays the GameCAST web application. On the left is a dark sidebar with the title 'GameCAST' and a user profile section for 'USERNAME' (BigTimeStrimmer) with links to 'Stats' and 'Forms'. The main content area has a purple header and contains three sections: 'User Data' with input fields for 'Twitch Username', 'Preferred Genres' (containing 'Action, Shooter, FPS'), and 'Streamed Games' (containing 'Game1, Game2, Game3'), followed by a 'Submit' button; 'Games Recommended'; and 'Genres Recommended'. The bottom of the page is a solid purple block.



## Business **Implementation**

- ▶ **Improve** streamer experience and retention
- ▶ **Encourage** channel growth and diversity
- ▶ **Increase** revenue for Twitch