YouTube Video Analysis

CSYE 7200 Final Project Presentation

Team 2
Deepika Balasubramanian, Binghui Lai,
Siyuan Xu, Weikun Zhuang

Use Cases

- System processes video statistics from YouTube API and retrieves top trending videos across the platform
- System also finds the topics of trending videos and plots the count in each trending topic
- The model predicts the ‘Likes’ count of trending videos, trained from past trending video data
- User inputs a specific topic of interest and receives a list of trending videos within that topic. (Show Today’s trending videos)
- A web interface (dashboard) which shows visualization of trending video counts in different categories and description and image of top trending videos. (Showing statistics changing trend instead)

Methodology

- Use Google YouTube API to read the real-time trending video data, and use Scala to write the File I/O system to read daily data into .csv files
- Use Spark dataset to read in and preprocess the data and Spark MLlib for machine learning model training
- Use Scala plotting libraries to make graphs for visualizations (Using Databricks notebook to generate graphs, including processing)
- Use Play framework to build the web interface which allows users to interact with and provide users the results and graphs the application generated (Visualizations kept in Databricks)

Data Sources

- We will collect over 30,000 rows of data from the YouTube API on trending videos within this month (once every day on top 50 trending videos) and, to complement the required dimension, existing datasets for past YouTube trending videos on Kaggle.

  [Crawled data in past 3 weeks + Kaggle dataset for past 2 years = 31000 rows]

- Some of the features of data will include:

  Video ID, Title, CategoryID, Tags, View_count, Likes, Comment_count

Milestones

Week 1 : Nov. 10 - 16 Collect YouTube API data and Kaggle datasets

Week 2: Nov 17 - 23 Implement data structures and I/O system to ingest and store data and prepare for analysis. Start on UI interface implementation

Week 3: Nov 24 - 30 Integrate with Spark, apply Spark MLlib to perform analysis

[UI and file I/O system]

Week 4: Dec 1 - Dec 7 Visualization, UI interface wrap-up, and tidy things up

[Model training, dashboard]

What will we program in Scala?

- Use Python to interact and retrieve data from YouTube API
- Use Scala to preprocess the data
- Generate Spark dataframe and apply ML algorithms using Spark MLlib
- If Play framework used, we might also include JavaScript code to integrate visualizations generated by Scala [Javascript was not sued. Visualizations switched to dashboard]

Acceptance Criteria

- Web interface response time should be within 4 seconds

  (Training on the fly and takes on average ~ 8 seconds)

- If regression model was used, R-square score should be around 0.7

  (The R-square was around 0.6)

Goal of the Project

- Build a big-data system that supports streamline processing of large-volume data
- Getting insights on YouTube analytics and explore the characteristics of trending videos
- Predict the trending videos' like, and/or view, comment counts by analyzing how the video statistics change over time

Possible Improvements

- Fitting a more advanced model and possibly incorporate more features and time-series component in the model training - with support of existing Spark Libraries.
- Improvements on the UI - incorporating more web design and more seamless integration with other components.