PIPELINE PROJECT ARCHITECTURE

SMART YOUTUBE CHANNEL RECOMMENDER FOR YOUR PRODUCT

- **ADK2159, AMP2313, KS3630, NS3308**

1. A brief description

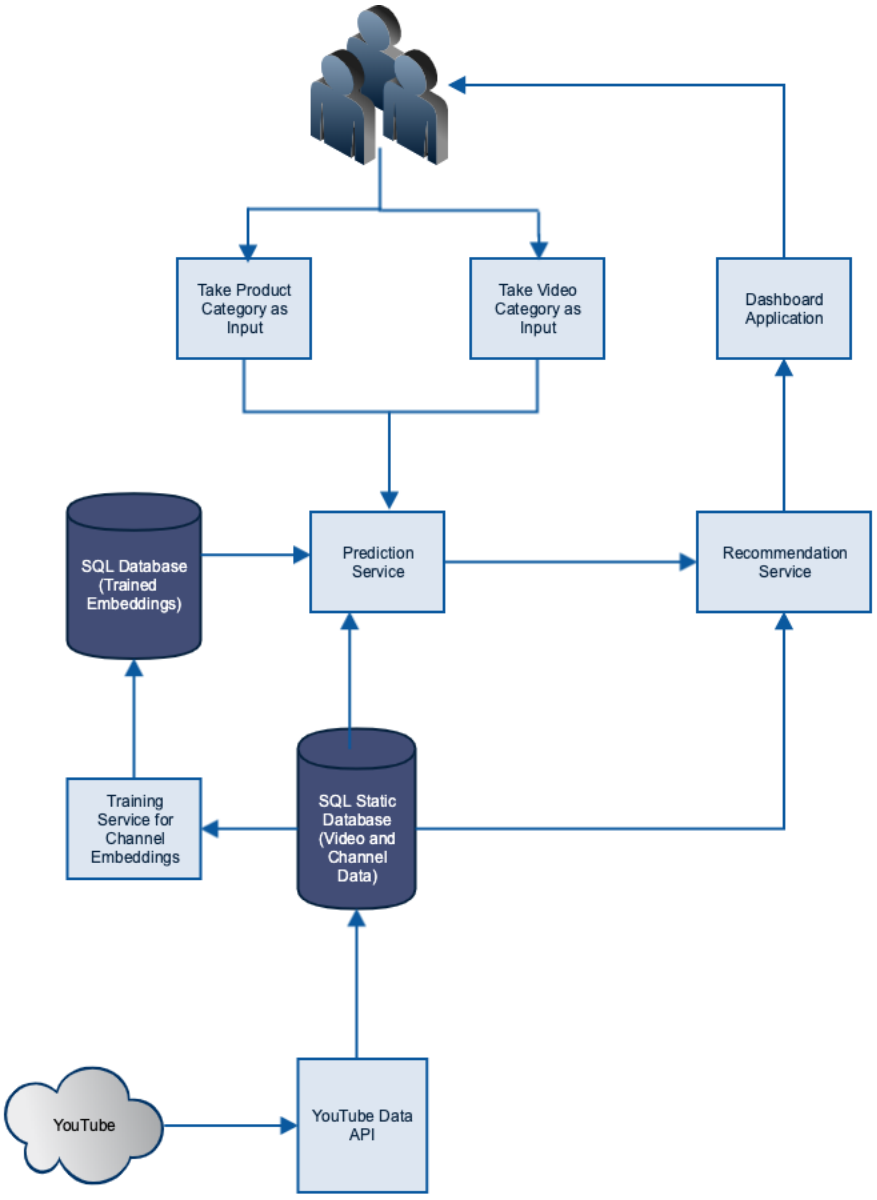
Our product aims to provide YouTube channel recommendations to a company for a product that they want to promote, based on multiple features of the channel, their videos and the product, as explained below:

- Channel and Video Features:
  - Title of the channel
  - Category
  - Tags
  - Views
  - Likes
  - Subscriber count
  - Thumbnail Image
  - Comment text
- Product Features:
  - Product description
  - Video Category
  - Product Category
  - Feature Weightage

Based on the input given by the user, we use the features above to recommend the best channel for them to approach to promote their product.

The architectures for the Minimum viable product and first iterations are given below.

2. Minimum viable product architecture



Overview of the architecture:
The user, from the front-end, will choose a product category (eg: health and wellness, fashion, etc) and video category (eg: motivational, funny, etc) which will go into the prediction service, along with the tags and channel information about the YouTube videos and generate similarity scores, which will be sent into the recommendation service, which will use this data along with the video's like count, channel's subscriber count and so on to rank the channels and recommend the top 5 channels to the user, which will be displayed using the dashboard along with various statistics and information regarding the channel, so they can choose the best one for them.

Description of each service:

i. Training Service:
   a. Input – Text features from the static Video/Channel dataset
   b. Output – Word embeddings for the input data

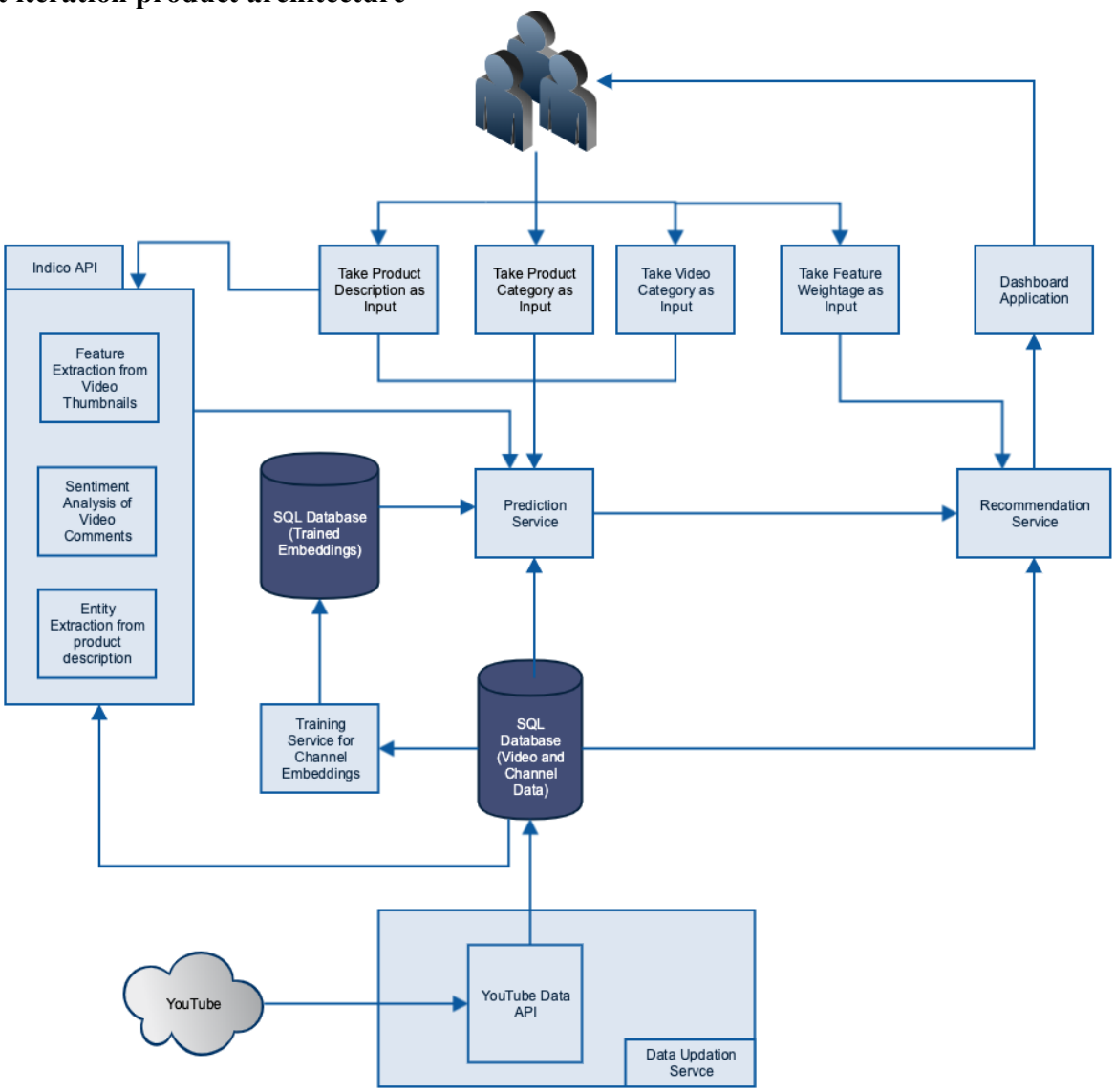
   This service trains a Word2Vec model on text data, and returns word-embeddings for them, and they are then stored in a database. This service only needs to be called once, as for the MVP our data is static.

ii. Prediction Service:
   a. Input – (1) Word Embeddings from the database, (2) User Input, (3) Features of the static Video/Channel dataset
   b. Output – Similarity scores between input and historic data

iii. Recommendation Service:
   a. Input – Similarity scores (Output of prediction service)
   b. Output – Top 5 Channel recommendations

   This service uses the similarity scores that the prediction service provides, and also then uses the scores that have matched the input product and video category to YouTube videos' tags and category name from the database, and the video's like count, channel's subscriber count and so on, aggregates these results by category, ranking the channels accordingly and gets the top 5 channels to recommend.

3. First iteration product architecture



Overview of the architecture:

The user, from the front-end, will choose a product category (eg: health and wellness, fashion, etc), video category (eg: motivational, funny, etc), give a product description and also give weightage to the features that they want (eg: more weightage to number of subscribers, less to similar products, etc). The first 3 input features will go into the prediction service, along with the tags and channel information about the YouTube videos, as well as the output from the Indico API, and generate similarity scores, which will be sent into the recommendation service. The recommendation service will use this data along with the video's like count, channel's subscriber count and so on to rank the channels according to the weights given by the user to recommend the top 5 channels, which will be displayed using the dashboard along with various statistics and information regarding the channel, so they can choose the best one for them.

Description of each service:

i. Data Updation Service:
   a. Input – YouTube Videos
   b. Output – Scraped Data

   This service will use the YouTube Data API to periodically update our database, to ensure that our data has the latest information.

ii. Training Service:
   a. Input – Text features from the Video/Channel dataset (which is now periodically updated).
   b. Output – Word embeddings for the input data

   This service trains a Word2Vec model on text data, and returns word-embeddings for them, and they are then stored in a database. This service needs to be called periodically to re-train whenever we fetch new information and update the database.

iii. Prediction Service:
   a. Input – (1) Word Embeddings from the database, (2) User Input, (3) Features of the Video/Channel dataset, (4) Features extracted by Indico API
   b. Output – Similarity scores between input and historic data

iv. Recommendation Service:
   a. Input – Similarity scores (Output of prediction service)
   b. Output – Top 5 Channel recommendations

   This service uses the similarity scores that the prediction service provides, and also then uses the scores that have matched the input product and video category to YouTube videos' tags and category name from the database, and the video's like count, channel's subscriber count and so on, aggregates these results by category, ranking the channels accordingly and gets the top 5 channels to recommend.