Task 8 HD: Project

Introduction

A recommendation system is a machine learning algorithm that uses data to help users find products and content. They are a subset of AI tools that use complex algorithms to analyze large data sets and rank a user's interest in a set of items by examining user preferences and behaviors. Recommendation systems can be used in a variety of industries, including: E-commerce and retail, Media and entertainment, Personalized banking, Healthcare, and Finance.

In this case study we will be creating a BOT which will automate movie name recommendations for customers based on their provided movie. It will interact with customers using both text and voice inputs, allowing for a more natural and versatile user experience.

Dataset

These files contain metadata for all 45,000 movies listed in the Full MovieLens Dataset. The dataset consists of movies released on or before July 2017. This dataset consists of the following files:

credits.csv: Consists of Cast and Crew Information for all our movies. Columns are:

- movie_id: A unique identifier for each movie.
- Title: Title of the movie.
- cast: The name of lead and supporting actors.
- Crew: The name of Director, Editor, Composer, Writer etc.

movies.csv: The main Movies Metadata file. Contains information on 45,000 movies featured in the Full MovieLens dataset. Features include posters, backdrops, budget, revenue, release dates, languages, production countries and companies.

- budget: The budget in which the movie was made.
- genre: The genre of the movie, Action, Comedy, Thriller etc.
- homepage: A link to the homepage of the movie.
- id: This is infact the movie_id as in the first dataset.
- keywords: The keywords or tags related to the movie.
- original_language: The language in which the movie was made.
- original_title: The title of the movie before translation or adaptation.
- overview: A brief description of the movie.
- popularity: A numeric quantity specifying the movie popularity.
- production_companies: The production house of the movie.
- production_countries: The country in which it was produced.
- release_date: The date on which it was released.
- revenue: The worldwide revenue generated by the movie.
- runtime: The running time of the movie in minutes.
- status: "Released" or "Rumored".
- tagline: Movie's tagline.
- title: Title of the movie.
- vote_average: average ratings the movie recieved.
- vote_count: the count of votes recieved.

Credit dataset output:

```
credits_df.head()
```

✓ 0.0s

	movie_id	title	cast	crew
0	19995	Avatar	[{"cast_id": 242, "character": "Jake Sully", "...	[{"credit_id": "52fe48009251416c750aca23", "de...
1	285	Pirates of the Caribbean: At World's End	[{"cast_id": 4, "character": "Captain Jack Spa...	[{"credit_id": "52fe4232c3a36847f800b579", "de...
2	206647	Spectre	[{"cast_id": 1, "character": "James Bond", "cr...	[{"credit_id": "54805967c3a36829b5002c41", "de...
3	49026	The Dark Knight Rises	[{"cast_id": 2, "character": "Bruce Wayne / Ba...	[{"credit_id": "52fe4781c3a36847f81398c3", "de...
4	49529	John Carter	[{"cast_id": 5, "character": "John Carter", "c...	[{"credit_id": "52fe479ac3a36847f813eaa3", "de...

Movies dataset output:

```
movies_df.head()
```

✓ 0.0s Python

	budget	genres	homepage	id	keywords	original_language	original_title	overview	popularity	production_compan
0	237000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.avatarmovie.com/	19995	[{"id": 1463, "name": "culture clash"}, {"id":...	en	Avatar	In the 22nd century, a paraplegic Marine is di...	150.437577	[{"name": "Ingenic Film Partners", "i 28
1	300000000	[{"id": 12, "name": "Adventure"}, {"id": 14, "...	http://disney.go.com/disneypictures/pirates/	285	[{"id": 270, "name": "ocean"}, {"id": 726, "na...	en	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	139.082615	[{"name": "Walt Disr Pictures", "id": 2},
2	245000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.sonypictures.com/movies/spectre/	206647	[{"id": 470, "name": "spy"}, {"id": 818, "name...	en	Spectre	A cryptic message from Bond's past sends him o...	107.376788	[{"name": "Colum Pictures", "id": {"nar
3	250000000	[{"id": 28, "name": "Action"}, {"id": 80, "nam...	http://www.thedarkknightrises.com/	49026	[{"id": 849, "name": "dc comics"}, {"id": 853,...	en	The Dark Knight Rises	Following the death of District Attorney Harve...	112.312950	[{"name": "Legend Pictures", "id": 923},
4	260000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://movies.disney.com/john-carter	49529	[{"id": 818, "name": "based on novel"}, {"id":...	en	John Carter	John Carter is a war-weary, former military ca...	43.926995	[{"name": "Walt Disr Pictures", "id":

Once data cleansing and feature engineering is done, our merged dataset will be like:

```
movies_credits_df.head()
```

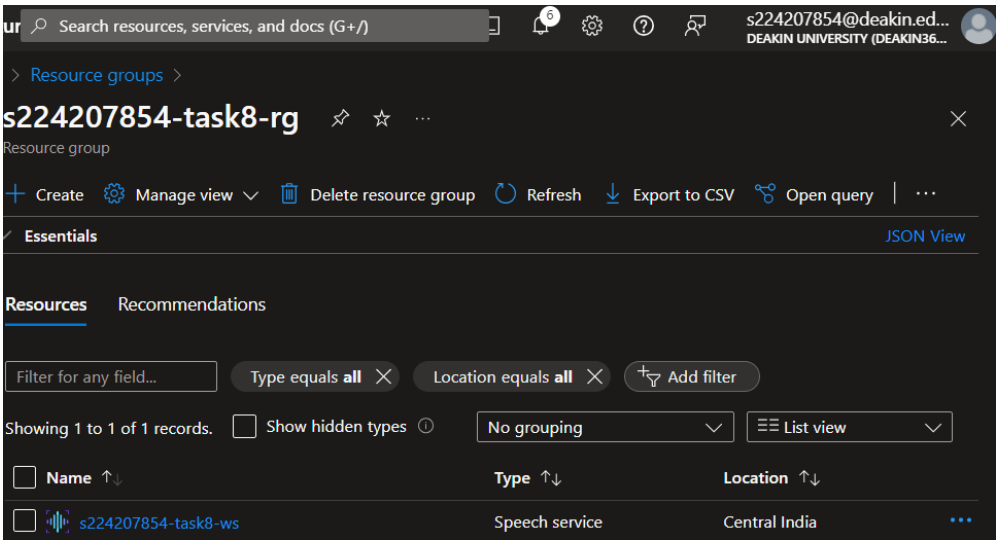
✓ 0.0s Pytho

	id	original_title	genres	cast	director	vote_average	keywords	combine_feature
0	19995	Avatar	[Action, Adventure, Fantasy, ScienceFiction]	[SamWorthington, ZoeSaldana, SigourneyWeaver, ...	James Cameron	7.2	['culture clash', 'future', 'space war', 'spac...	['cultureclash','futu...
1	285	Pirates of the Caribbean: At World's End	[Adventure, Fantasy, Action]	[JohnnyDepp, OrlandoBloom, KeiraKnightley, Ste...	Gore Verbinski	6.9	['ocean', 'drug abuse', 'exotic island', 'east...	['ocean','drugabuse',...
2	206647	Spectre	[Action, Adventure, Crime]	[DanielCraig, ChristophWaltz, LéaSeydoux, Ralp...	Sam Mendes	6.3	['spy', 'based on novel', 'secret agent', 'seq...	['spy','basedonnovel...
3	49026	The Dark Knight Rises	[Action, Crime, Drama, Thriller]	[ChristianBale, MichaelCaine, GaryOldman, Anne...	Christopher Nolan	7.6	['dc comics', 'crime fighter', 'terrorist', 's...	['dccomics','crimefi...
4	49529	John Carter	[Action, Adventure,	[TaylorKitsch, LynnCollins,	Andrew	6.1	['based on novel', 'mars',	['basedonnovel'

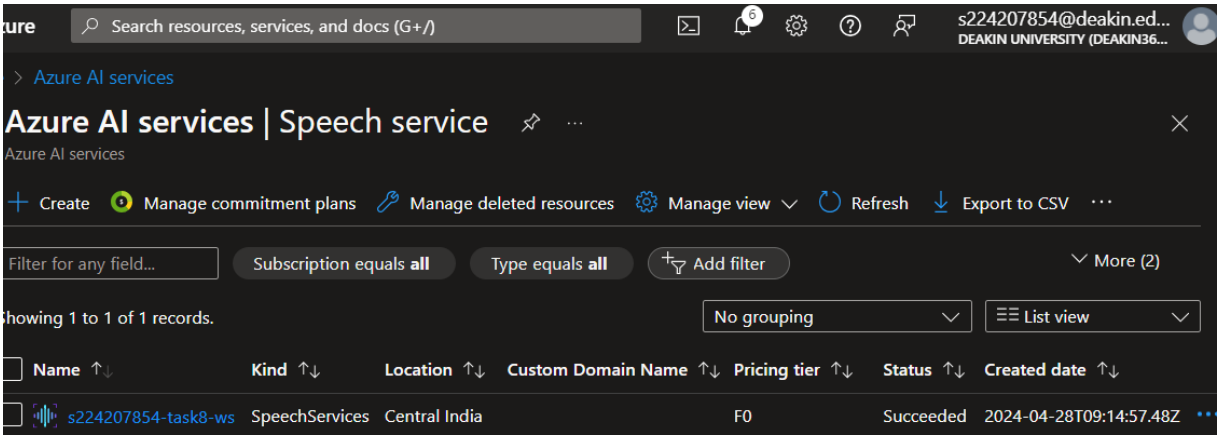
Pre-requisites

In this case study we will be using Azure speech service using Azure Cognitive Services SDK. Before starting coding will be doing some set-ups:

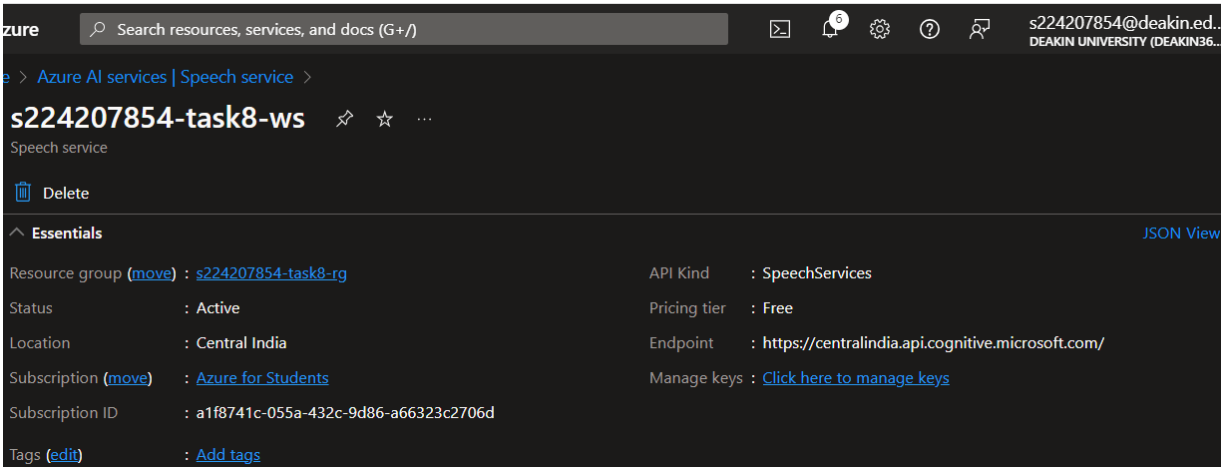
1. Resource Group set-up



2. Creating speech service in Azure AI services



3. In Azure AI service we can visit our speech service



4. Once pre-requisites are done, we can use this info for our case study.

Notebook

In our notebook we will be using movies and credits dataset, where different data cleansing will be performed. Converting of JSON values into string format, removing rows where empty values are present and removal of unwanted characters from different columns. Now will be looking for different screenshots where different checks have been done:

- Changing “genres” column values to string

```
# changing the genres column from json to string
movies_credits_df["genres"] = movies_credits_df["genres"].apply(json.loads)
for index, i in zip(movies_credits_df.index, movies_credits_df["genres"]):
    list1 = []
    for j in range(len(i)):
        list1.append((i[j]["name"]))
    movies_credits_df.loc[index, "genres"] = str(list1)
```

- Identifying and removing empty values

```
movies_credits_df.isnull().sum()
```

```
✓ 0.0s

id               0
original_title   0
genres           0
cast             0
director        30
vote_average     0
keywords         0
dtype: int64
```

```
movies_credits_df.dropna(inplace=True)
```

- Removing unwanted characters from different string columns

```
movies_credits_df["genres"] = (
    movies_credits_df["genres"]
    .str.strip("[]")
    .str.replace(" ", "")
    .str.replace("'", "")
)

movies_credits_df["genres"] = movies_credits_df["genres"].str.split(",")
```

Once data has been cleansed our final dataset will be looking like:

```
movies_credits_df.head()
```

✓ 0.0s

	id	original_title	genres	cast	director	vote_average	keywords	combine_feature
0	19995	Avatar	[Action, Adventure, Fantasy, ScienceFiction]	[SamWorthington, ZoeSaldana, SigourneyWeaver, ...	James Cameron	7.2	['culture clash', 'future', 'space war', 'spac...	['cultureclash','futu...
1	285	Pirates of the Caribbean: At World's End	[Adventure, Fantasy, Action]	[JohnnyDepp, OrlandoBloom, KeiraKnightley, Ste...	Gore Verbinski	6.9	['ocean', 'drug abuse', 'exotic island', 'east...	['ocean','drugabuse',...
2	206647	Spectre	[Action, Adventure, Crime]	[DanielCraig, ChristophWaltz, LéaSeydoux, Ralp...	Sam Mendes	6.3	['spy', 'based on novel', 'secret agent', 'seq...	['spy','basedonnovel...
3	49026	The Dark Knight Rises	[Action, Crime, Drama, Thriller]	[ChristianBale, MichaelCaine, GaryOldman, Anne...	Christopher Nolan	7.6	['dc comics', 'crime fighter', 'terrorist', 's...	['dccomics','crimefi...
4	49529	John Carter	[Action, Adventure,	[TaylorKitsch, LynnCollins,	Andrew	6.1	['based on novel', 'mars',	['basedonnovel'

Now in our dataset we will be using CountVectorizer which converts a collection of text documents into a matrix where the rows represent the documents, and the columns represent the tokens (words or n-grams). It counts the occurrences of each token in each document, creating a “document-term matrix” with integer values representing the frequency of each token.

Cosine similarity is a metric used to measure how similar two items are. In the context of AI, cosine similarity evaluation is utilized to assess the similarity between different datasets.

```
cv = CountVectorizer(stop_words="english")  # creating new CountVectorizer() object
count_matrix = cv.fit_transform(
    movies_credits_df["combine_feature"]
)  # feeding combined strings(movie contents) to CountVectorizer() object
cosine_sin = cosine_similarity(count_matrix, count_matrix)
```

Now we will be defining a function which takes movie title as input and outputs most similar movies.

```python
def get_movie_recommendation(title, cosine_sim=cosine_sin):
    try:
        # Get the index of the movie that matches the title
        idx = indices[title]

        # Get the pairwsie similarity scores of all movies with that movie
        sim_scores = list(enumerate(cosine_sim[idx]))

        # Sort the movies based on the similarity scores
        sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)

        # Get the scores of the 10 most similar movies
        sim_scores = sim_scores[1:11]

        # Get the movie indices
        movie_indices = [i[0] for i in sim_scores]

        # Return the top 10 most similar movies

        print("-----------Movies-Recommended-----------------")
        print("---" * 15)
        return movies_credits_df["original_title"].iloc[movie_indices]
    except Exception:
        return "I'm not sure how to respond to that."
```

Till now we have developed our recommendation model which will be using content-based filtering.

As of now inputs will be in 2 ways, i.e. text and voice.

As of now we will be creating speech SDK, which will take input from microphone and converting speech to text so that it can be used via our recommender.

```python
speech_config = speechsdk.SpeechConfig(subscription=ss_key, region=region)
speech_recognizer = speechsdk.SpeechRecognizer(speech_config=speech_config)
```

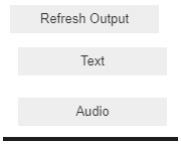
0.6s

```python
def speech_input(b):
    print("Listening....")
    result = speech_recognizer.recognize_once()

    if result.reason == speechsdk.ResultReason.RecognizedSpeech:
        recognized_text = result.text.rstrip(
            "." "."
        )  # Remove dot from the end of recognized text

        print("Recognized: {}".format(recognized_text))
        print(get_movie_recommendation(recognized_text))
    elif result.reason == speechsdk.ResultReason.NoMatch:
        raise Exception(
            "No speech could be recognized: {}".format(result.no_match_details)
        )
    elif result.reason == speechsdk.ResultReason.Canceled:
        cancellation_details = result.cancellation_details
        raise Exception(
            "Speech Recognition canceled: {}".format(cancellation_details.reason)
        )

def on_enter_key_pressed(change):
    if change["name"] == "value" and change["new"]:
        user_input = change["new"].replace("\n", "")  # Remove the newline character
        print("Recognized: {}".format(user_input))
        print(get_movie_recommendation(user_input))
        text_input.value = ""  # Clear the text input after processing the input
```

Defining BOT, that will be used by users to get the recommendation

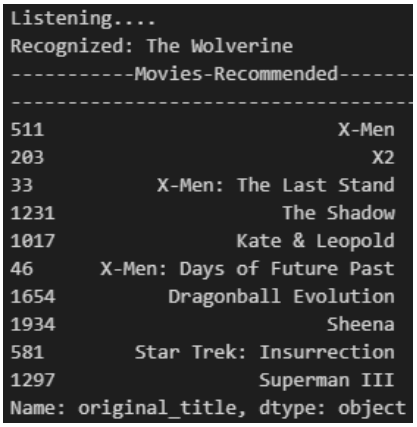
```
def bot():

    def refresh_output(b):
        clear_output(wait=True)
        bot()

    def get_text(b):
        display(text_input)

    refresh_button = widgets.Button(description="Refresh Output")
    button1 = widgets.Button(description="Text")
    button2 = widgets.Button(description="Audio")
    text_input = widgets.Text(placeholder="Enter your input", continuous_update=False)

    # Assign the function to be called when the button is clicked
    refresh_button.on_click(refresh_output)
    button1.on_click(get_text)
    button2.on_click(speech_input)
    text_input.observe(
        on_enter_key_pressed
    )  # Assign the function to be called when Enter is pressed in the text input box

    # Attach the function to the button's click event
    display(refresh_button)
    display(button1)
    display(button2)


bot()
```

Output will be like:

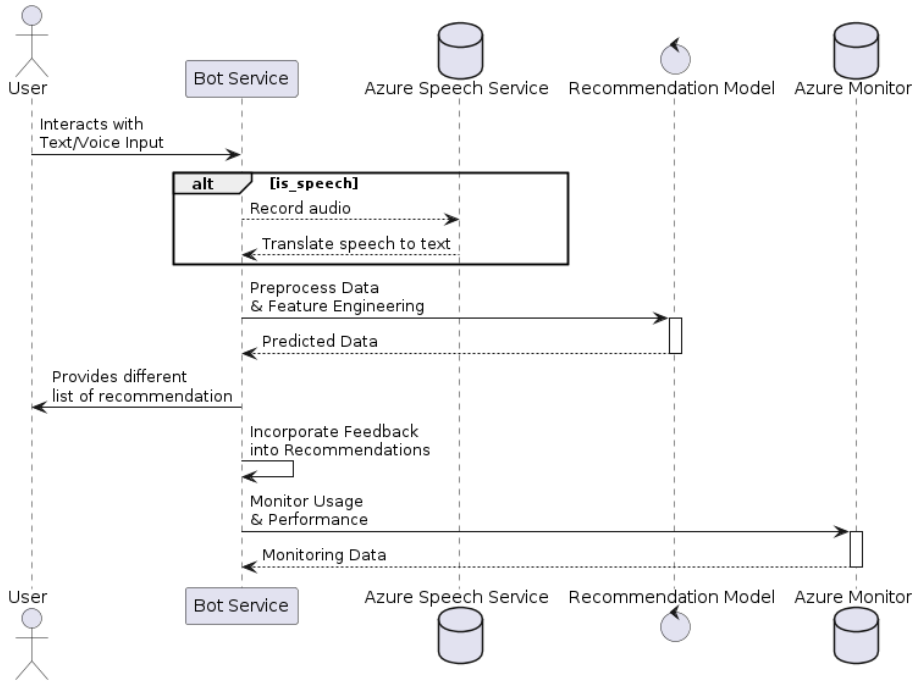


- user selected “Text” as input:

- user selected “Audio” as input:



Diagram



Resource Creation Clean-up

Once task has been completed, will start resource clean-up activity. We can directly clean the resource from our “Resource groups” section, which will remove all resources created under this group, like bot, language service, storage, workspace and more.

Summary

In this task we have learnt about natural language processing, recommendation system and the usage of Azure with speech service. In this BOT, we can upgrade many different techniques and can make a far better model that can be used on more recommendation over movies with providing more detailed results.

References

github.com, n.d. *cognitive-services-speech-sdk.* [Online]
Available at: https://github.com/Azure-Samples/cognitive-services-speech-sdk/tree/master/quickstart/python/from-microphone

Great Learning, n.d. *SIG788 - Engineering AI solutions Content.* [Online]
Available at: https://olympus.mygreatlearning.com/courses/109578?module_id=747613

learn.microsoft.com, n.d. *Speech to text documentation.* [Online]
Available at: https://learn.microsoft.com/en-us/azure/ai-services/speech-service/index-speech-to-text