2020 ECE SUMMER HONORS RESEARCH PROGRAM -

Industrial Recommender Systems in Media & Entertainment

- *By Siddharth Mandgi*
  under Prof Hong Man

WEEK-3

Table of Contents

MOVIE RECOMMENDER ENGINE

Sentiment Based Recommendations (Feature Based Recommendations)

Amazon Dataset

In this approach we make use of ***textual reviews*** of movies from the ***Amazon Movie Dataset*** and apply natural language processing to predict sentiments which is then used to generate recommendations for users.

	reviewerID	asin	reviewerName	helpful	unixReviewTime	reviewTime	reviewText	summary	overall	sentiment
0	ADZPIG9QOCDG5	0005019281	Alice L. Larson "alice-loves-books"	[0, 0]	1203984000	02 26, 2008	This is a charming version of the classic Dick...	good version of a classic	4	Positive
1	A35947ZP82G7JH	0005019281	Amarah Strack	[0, 0]	1388361600	12 30, 2013	It was good but not as emotionally moving as t...	Good but not as moving	3	Neutral
2	A3UORV8A9D5L2E	0005019281	Amazon Customer	[0, 0]	1388361600	12 30, 2013	Don't get me wrong, Winkler is a wonderful cha...	Winkler's Performance was ok at best!	3	Neutral
3	A1VKW06X1O2X7V	0005019281	Amazon Customer "Softmill"	[0, 0]	1202860800	02 13, 2008	Henry Winkler is very good in this twist on th...	It's an enjoyable twist on the classic story	5	Positive
4	A3R27T4HADWFFJ	0005019281	BABE	[0, 0]	1387670400	12 22, 2013	This is one of the best Scrooge movies out. H...	Best Scrooge yet	4	Positive

spaCy

We use spaCy for NLP techniques such as importing stop words, tokenizations and transformation.

```
stopwords = list(STOP_WORDS)
stopwords
```

```
['whenever',
 "'m",
 'whither',
 'very',
 'is',
 'still',
 'see',
 '’m',
 'put',
 'first',
 'himself',
 'toward',
 'latterly',
```

```
def spacy_tokenizer(sentence):
    mytokens = parser(sentence)
    mytokens = [word.lemma_.lower() if word.lemma_ != "-PRON-" else word.lower_ for word in mytokens]
    mytokens = [word for word in mytokens if word not in stopwords and word not in punctuations]
    return mytokens
```

Pipeline

We then build a pipeline using spacy functions and sklearn classifiers through which our training data (review comments) and labels (sentiments) are passed and predictions are generated for the test dataset.

```
from sklearn.linear_model import LogisticRegression
classifier =  LogisticRegression()

pipe = Pipeline([('cleaner', predictors()),
                 ('tfidfVect', tfdifVect),
                ('classifier',classifier),])
```

IMDB - Sentiments

Our trained model has also been used to train, predict sentiments and generate recommendations for the IMDB dataset.

```
IMDB.loc[IMDB['sentiment'] == "Positive"].tail(100)
```

	review	sentiment
49860	One of the best and most exciting of all consp...	Positive
49861	OK OK, it might be hard to put the entirety of...	Positive
49862	I saw this series when it world premiered at t...	Positive
49863	It's a testament to Gosha's incredible film-ma...	Positive
49864	Thanks to the helpfulness of a fellow IMDb mem...	Positive
...	...	...
49992	John Garfield plays a Marine who is blinded by...	Positive
49993	Robert Colomb has two full-time jobs. He's kno...	Positive
49995	I thought this movie did a down right good job...	Positive
49997	I am a Catholic taught in parochial elementary...	Positive
49998	I'm going to have to disagree with the previou...	Positive

100 rows × 2 columns

IMDB - Ratings

Recommendations for IMDB based on Ratings: -

	review	sentiment	ratings
0	One of the other reviewers has mentioned that ...	Positive	5
1	A wonderful little production. The...	Positive	5
4	Petter Mattei's "Love in the Time of Money" is...	Positive	5
5	Probably my all-time favorite movie, a story o...	Positive	5
6	I sure would like to see a resurrection of a u...	Positive	5
...	...	...	...
208	I just started watching The Show around July. ...	Positive	5
209	This film is well cast, often silly and always...	Positive	5
213	Normally I don't like series at all. They're a...	Positive	5
216	I saw this at the London Film Festival last ni...	Positive	5
218	This movie really woke me up, like it wakes up...	Positive	5

MUSIC RECOMMENDER ENGINE

Spotify EDA (Feature Based Recommendations)

In this approach of music recommendation engine, we focus our research on understanding how Spotify analyses different features of music listed below: -

Features

Danceability: Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.

Energy: Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy.

Instrumentalness: Predicts whether a track contains no vocals. The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content.

Liveness: Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live.

Loudness: The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track. Values typical range between -60 and 0 db.

Speechiness: Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value.
Tempo: The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.

Valence: A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).

Custom Function

We first use Spotipy API and create a custom function to import playlists from different artists and sources.

```
def analyze_playlist(creator, playlist_id):

    # Create empty dataframe
    playlist_features_list = ["artist","album","track_name",  "track_id","danceability","energy","key","loudness"

    playlist_df = pd.DataFrame(columns = playlist_features_list)

    # Loop through every track in the playlist, extract features and append the features to the playlist df

    playlist = sp.user_playlist_tracks(creator, playlist_id)["items"]
    for track in playlist:
        # Create empty dict
        playlist_features = {}
        # Get metadata
        playlist_features["artist"] = track["track"]["album"]["artists"][0]["name"]
        playlist_features["album"] = track["track"]["album"]["name"]
        playlist_features["track_name"] = track["track"]["name"]
        playlist_features["track_id"] = track["track"]["id"]

        # Get audio features
        audio_features = sp.audio_features(playlist_features["track_id"])[0]
        for feature in playlist_features_list[4:]:
            playlist_features[feature] = audio_features[feature]

        # Concat the dfs
        track_df = pd.DataFrame(playlist_features, index = [0])
        playlist_df = pd.concat([playlist_df, track_df], ignore_index = True)

    return playlist_df
```

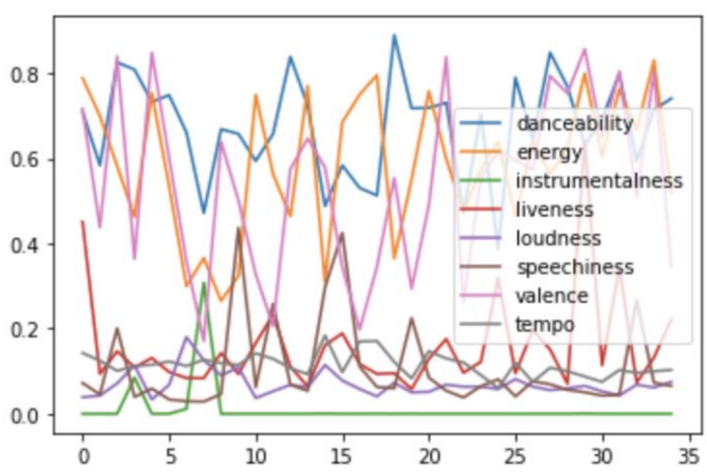
Playlist

A sample Spotify playlist and it’s features.

	artist	album	track_name	track_id	danceability	energy	key	loudness	mode	speechiness	instrumentalness	liveness
0	Bazzi	COSMIC	Mine	7uzmGiiJyRfuViKKK3lVmR	0.710	0.789	4	-3.874	1	0.0722	0.000003	0.4510
1	ZAYN	Mind Of Mine (Deluxe Edition)	PILLOWTALK	0PDUDa38GO8lMxLCRc4lL1	0.584	0.700	11	-4.275	1	0.0456	0.000000	0.0939
2	Alex Aiono	Work The Middle	Work The Middle	42rB0s3mv8BpSKVBVkULvY	0.826	0.582	10	-7.162	0	0.2010	0.000005	0.1470
3	Drake	More Life	Passionfruit	5mCPDVBb16L4XQwDdbRUpz	0.809	0.463	11	-11.377	1	0.0396	0.085000	0.1090
4	Charlie Puth	Voicenotes	Slow It Down	6TapyDFIRUIZ4cmlaqJx4k	0.733	0.754	0	-3.392	0	0.0593	0.000000	0.1310
5	Charlie Puth	Cheating on You	Cheating on You	0ClPleT6MSgfSgQ9ZrJbAq	0.749	0.535	4	-6.785	1	0.0336	0.000036	0.0982
6	Chris Isaak	Heart Shaped World	Wicked Game	34u3Bj1KVxKWfA07yDJ2vG	0.661	0.300	9	-18.055	1	0.0294	0.011800	0.0839
7	Pink Floyd	The Wall	Comfortably Numb	5HNCy40Ni5BZJFw1TKzRsC	0.472	0.366	11	-12.595	0	0.0286	0.308000	0.0837
8	Alex Aiono	Work The Middle (Acoustic Version)	Work The Middle - Acoustic	5DHvDSGFyNI02zDeQSJ07V	0.669	0.265	4	-9.018	0	0.0463	0.000000	0.1430

EDA

Analyzing this playlist, we find the respective results.



Results

Analyzing these features, we derive the following results: -

- This playlist suggests that the user likes music that has a high danceability and energy along with songs which have a high tempo.

- Valence suggests if the user has a positive outlook or negative outlook to music.

- In this case the user likes music that is more positive and livelier.

- Loudness values in this playlist suggests that the user likes softer music.

All these features can really help generate personalized recommendations to user.

Conclusion

In this week's report we generated recommendations based on Natural Language Processing of Movie Reviews and EDA of features such as danceability, loudness, valence, tempo, etc.
In the coming weeks we will apply traditional Recommender Systems such as content-based recommender systems and collaborative recommender systems along with implementing deep learning algorithms.