BIG DATA PROGRAMMING – CSC6760

Disneyland Reviews Sentimental Analysis

Abstract— Data is an important aspect of any business for faster decision making. The way we can organize the data in the warehouse to the day-to-day updates improves these decisions. The famous business enterprises like Walmart, Amazon…etc. use the warehouses to meet the demand of the delivery of the goods on a timely manner for a bigger population especially in cities. There are many startups recently that introduced the concept of delivery in minutes. The main strategy that runs these ideas is the warehouse. We are going to show how the databases can be used to perform the warehouse management and retrieve the information for various business insights and implement the ML Models to get the performance of the best ML Model. The maintaining of the warehouses in the Relational databases gives an easy way to take the insights on where the demand is! This gives the companies to sell their products easily without squandering the stock. The basic idea is to develop a web application with the user interface to perform the ML Algorithms and display the insights of items spread across different warehouses.

Keywords **-** Relational databases, decision making, ML Algorithms, Data Exploration, Text Preparation, Sentimental Analysis.

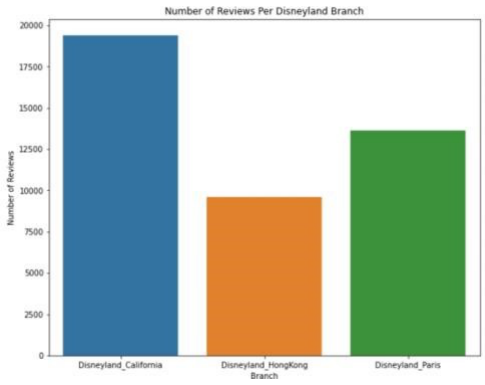
I. PROBLEM STATEMENT

To predict customer sentiments regarding attractions at three Disneyland locations (California, Paris, and Hong Kong) using sentimental analysis of reviews. Also, to identify keywords using topic modelling that can help Disney determine visitor pain points and improve park experience.

II. DATA DESCRIPTION

The dataset contains 42,656 reviews from TripAdvisor about three Disneyland branch locations: California, Paris, Hong Kong. There are 19,406 reviews about California, 13,630 about Paris, and 9,620 about Hong Kong.

Data variables include Review_ID, Rating, Year_Month, Reviewer_Location, Review_Text, and Branch.



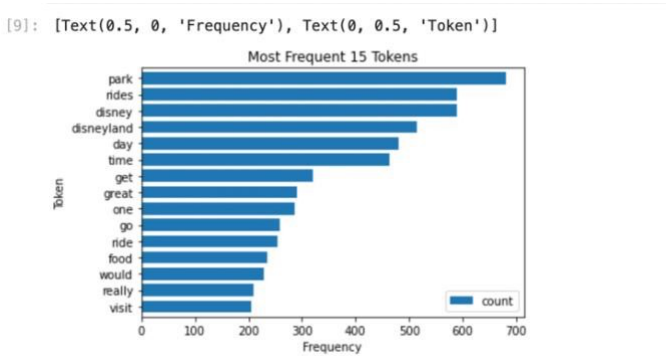
[6]:

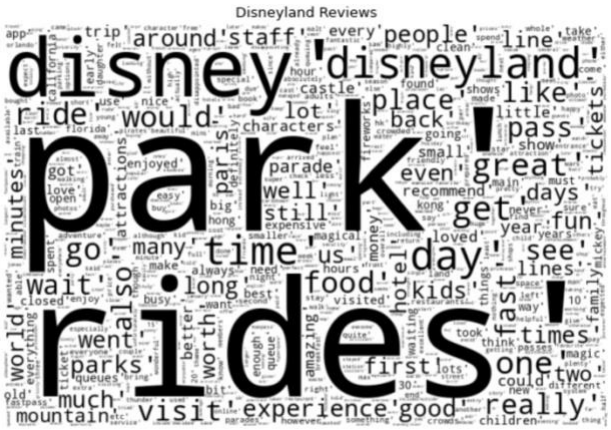
	review_id	rating	rating_binary	year_month	reviewer_location	review_text	branch	reviews_cleaned
0	666542833	5	1	2019-1	United States	Love this place very clean and safe have it ow...	Disneyland_HongKong	[love, place, clean, safe, mrt, get, night, ti...
1	666297074	5	1	2019-1	India	I would recommend to go through below tips to ...	Disneyland_HongKong	[would, recommend, go, tips, navigate, easily,...

III. TEXT EXPLORATION

Preparing the text for analysis involves two stages: data **pre-processing** and data **exploration**. Perform **Tokenization** to break up the text into smaller pieces to easily assign meaning to each of them. Explore the tokens using unique token count, total token count, most frequent and least used words, etc. to understand the **distribution of the text.**

There are in total 41668 tokens and, 5517 unique tokens.





IV. TEXT PREPARTION

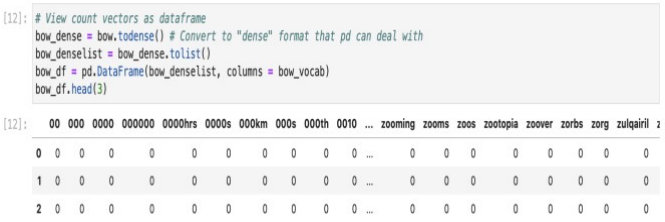
Convert into lowercase for ease of analysis and to avoid errors.

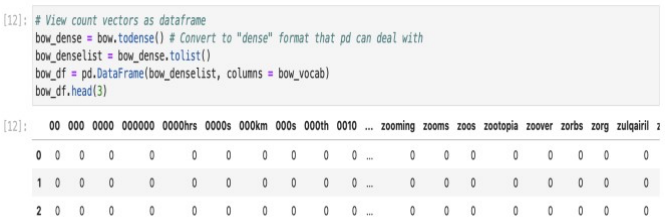
Drop punctuation and numbers, if they aren't necessary.

- **Text cleaning** using regular expression to avoid missing out on words like don't and wouldn't that have important punctuation marks to keep the meaning of the words intact.
- Perform **Tokenization** to break up the text into smaller pieces to easily assign meaning to each of them.
- Explore the tokens using unique token count, total token count, most frequent and least used words, etc. to understand the **distribution of the text**
- Convert into lowercase for ease of analysis and to avoid errors • Drop punctuation and numbers, if they aren't necessary
- Check for the possibility of reviews involving languages other than English and translate them to English for further analysis
- Drop **stop words** using **NLTK or Spacy** to declutter the text
- Select a frequency threshold and remove words with a frequency below the threshold to retain uniform significance
- Text cleaning using regular expression to avoid missing out on words like don't and wouldn't that have important punctuation marks to keep the meaning of the words intact.
- Stemming and Lemmatization using NLTK packages to avoid the occurrence of words in different forms that would otherwise give the same purpose or context.

V. TEXT REPRESENTATION

Bag-of-Words is the representation used for information retrieval. In this model, a token of text within the document (either query or review) is represented as the bag(multi-set) of its words disregarding grammar and even word order but keeping multiplicity.





TF-IDF (Term Frequency- Inverse Document Frequency) is a representation where the ranking function is defined such that it rewards words that frequently appear in a selected document (high Term Frequency) but are rare among other documents (Inverse Document Frequency), essentially representing the importance of a word in a document.

```
[ ] # Exploring TF-IDF vectors: sort row 0 by TF-IDF values
    row_0 = tfidf_df.iloc[[0]].copy()
    row_0 = row_0.sort_values(by=0, ascending=False, axis=1)
    row_0
```

	mrt	safe	beautiful	own	night	things	clean	love	too	here	many	place	do	to	very	get	have
0	0.446716	0.386841	0.274267	0.274267	0.239046	0.236378	0.217254	0.21165	0.194437	0.193075	0.181386	0.159965	0.157967	0.154938	0.147781	0.145791	0.13043

VI. SENTIMENT ANALYSIS(CLASSIFICATION)

Sentiment analysis is a powerful marketing tool that enables product managers to understand customer emotions in their marketing campaigns. It is an important factor when it comes to product and brand recognition, customer loyalty, customer satisfaction, advertising and promotion's success, and product acceptance.

- Support Vector Machine (SVM) In the SVM algorithm, we plot each data item as a point in n-dimensional space with the value of each feature being the value of a particular coordinate.

- BERT
  Binary classification Using BERT embeddings (inputs) and BERT (model)

- Logistic Regression
  It is used to make a prediction about a categorical variable versus a continuous one. A categorical variable can be true or false, yes or no, 1 or 0, et cetera.

- Naïve Bayes
  A Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

- Random Forest Classifier
  Random forest, like its name implies, consists of many individual decision trees that operate as an ensemble.

- XGBoost
  XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework.

```
[ ] # Set up model and tokenizer
    model = TFBertForSequenceClassification.from_pretrained("bert-base-uncased")
    tokenizer = BertTokenizer.from_pretrained("bert-base-uncased")

Downloading: 100% 570/570 [00:00<00:00, 17.5kB/s]
Downloading: 100% 536M/536M [00:11<00:00, 62.3MB/s]
All model checkpoint layers were used when initializing TFBertForSequenceClassification.
```

```
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=3e-5, epsilon=1e-08, clipnorm=1.0),
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=[tf.keras.metrics.SparseCategoricalAccuracy('accuracy')]) # Can choose another measure if y

model.fit(train_data, epochs=2, validation_data=validation_data)

Epoch 1/2
100/100 [==============================] - 112s 879ms/step - loss: 0.2721 - accuracy: 0.8950 - val_loss: 0.2534 -
Epoch 2/2
100/100 [==============================] - 90s 897ms/step - loss: 0.1011 - accuracy: 0.9719 - val_loss: 0.3033 - v
<keras.callbacks.History at 0x7f1782766550>
```

VII. PERFORMANCE METRICS

We have given the input Bag of words and TF-IDF and checked the performance of the ML Models.

[29]:

	bow x SVM	TF-IDF x SVM	TF-IDF x NB	TF-IDF x XGB	BERT x BERT
Precision	0.935484	0.916031	0.875912	0.929688	0.959016
Recall	0.966667	1.000000	1.000000	0.991667	0.975000
Accuracy	0.912409	0.919708	0.875912	0.927007	0.941606
F1 score	0.950820	0.956175	0.933852	0.959677	0.966942
MCC	0.558063	0.568599	0.000000	0.615135	0.718889
AU PRC score	0.933498	0.916031	0.875912	0.929239	0.956939
AU ROC score	0.748039	0.676471	0.500000	0.731127	0.840441

VIII. CONCLUSION

The best performing model from the above performance metrics is BERT and then followed by SVM with Bag of Words input and then XGBoost with TF-IDF input.