



Summative Capstone

Data Analysis of Universal Studios Theme Parks

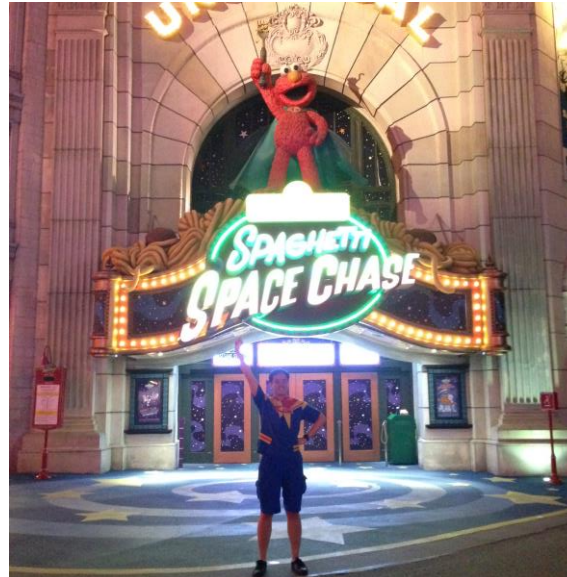
Contents:

- 1. Introduction
- 2. Objective
- 3. ER Diagram
- 4. Methodology
- 5. Process Workflow
- 6. Results
- 7. Conclusions
- 8. Appendix



1. Introduction

- Who are you? :
 - Ex-Operations Team Member in Universal Studios Singapore
 - Analyst for Universal Studios Theme Parks

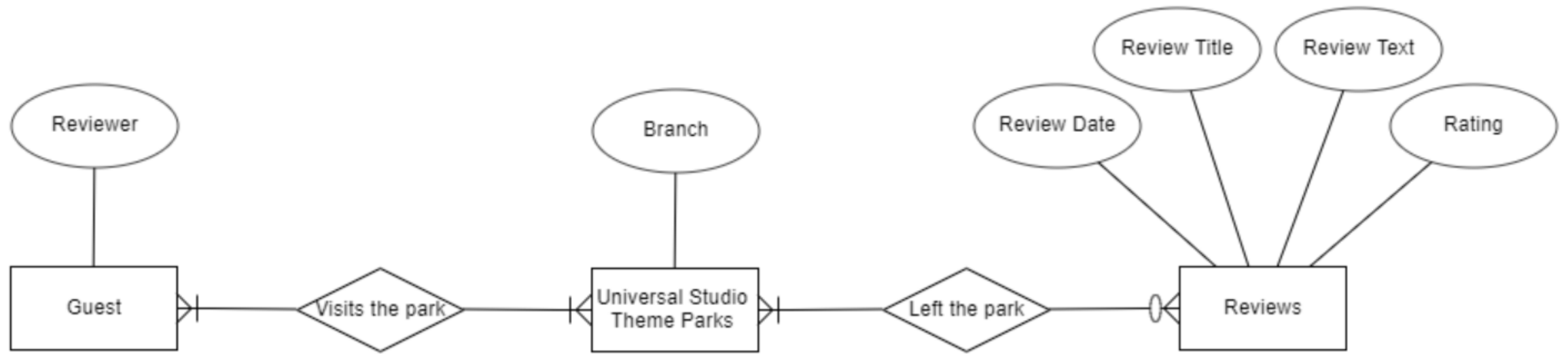


2. Objective

- Target audience:
 - Business Development Managers
 - Guest Service Managers
- Business case and Goals:
 - Constantly improve theme park quality
 - To continue attract visitors
 - Increase financial profit
 - To know which park merchandise is currently popular
- How will your prediction work help?
 - Give insights by analysing guest reviews and their links to higher or lower rating results



3. ER Diagram:



4.1. Methodology

- Datasets:
 - Kaggle
 - <https://www.kaggle.com/dwiknrd/reviewuniversalstudio>
 - 50,000+ reviews
 - 3 Universal Studios branches (Florida, Singapore and Japan)
- Data Analysis Tools:
 - MS SQL
 - Python
 - Power BI



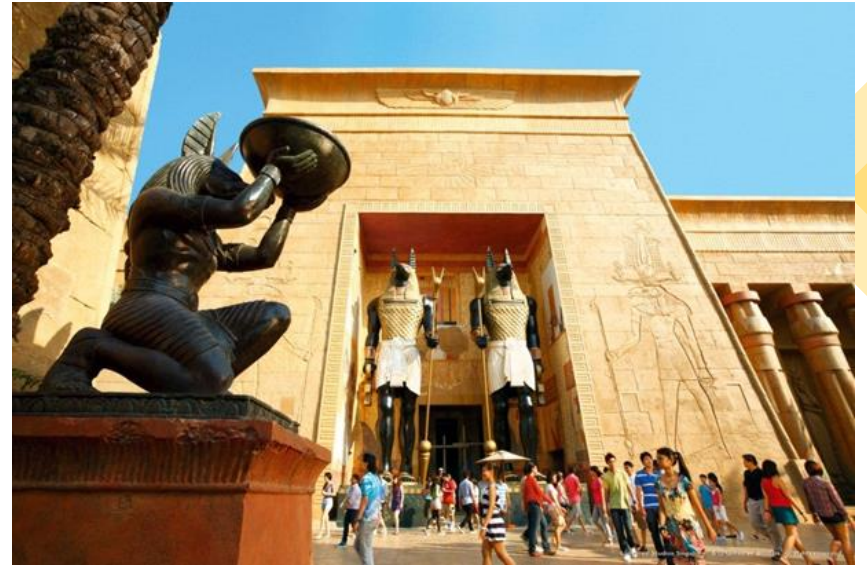
4.2. Methodology

- Models:
 - Supervised Machine Learning Model
 - Random Forest
 - Logistic Regression
 - Support Vector Machines (SVM)
 - Naive Bayes
 - DecisionTree
- Metrics:
 - F1 - Score



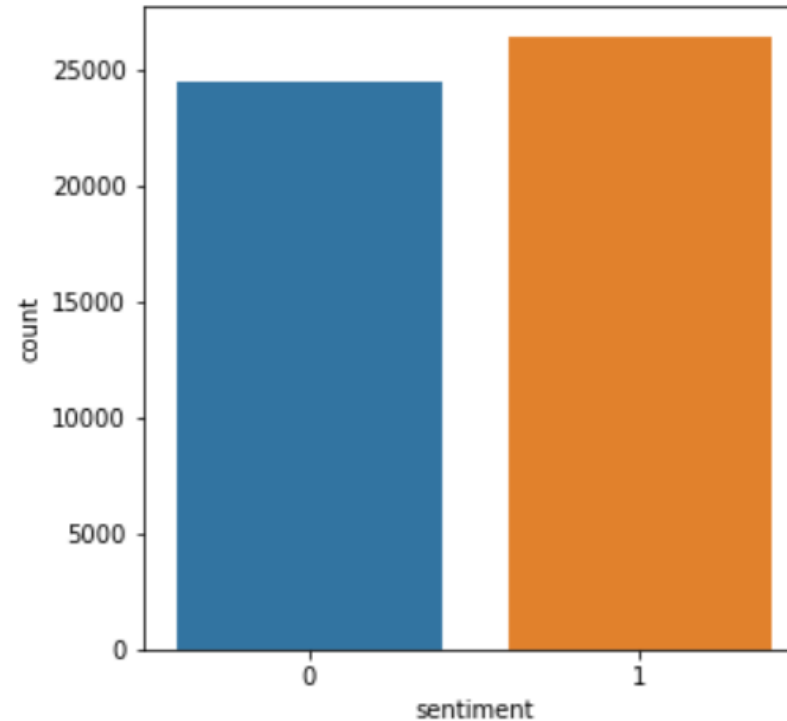
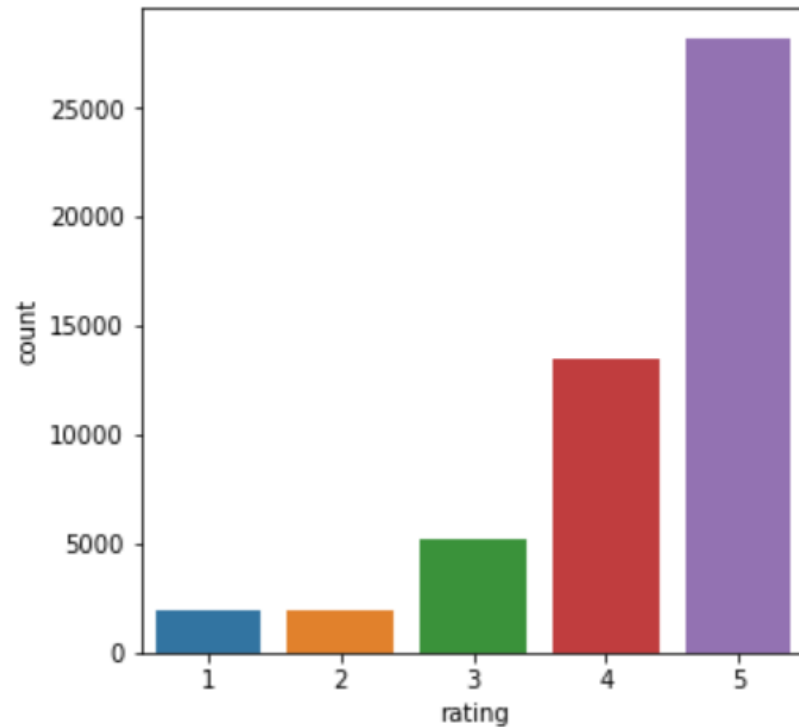
5.1. Process Workflow

- Data Preparation
 - Data Pre-Processing and Cleaning
 - Remove punctuations, rare and common occurring words, etc
 - Natural Language Process (NLP)
 1. **Tokenizing**
 - Break text into sentences, words, or other units
 2. **Removing Stop Words**
 - e.g. “if,” “but,” “or,” etc
 3. **Normalizing words**
 - Condensing all forms of word into a single form
 - Using Stemming or Lemming method
 4. **Vectorizing**
 - Turning text into numerical representation



5.2. Process Workflow

- EDA
 - Guest rating are very skewed sentiment analysis
 - Conducted sentiment analysis
 - Used mean polarity score from sentiment analysis to split



- 0 = Negative Sentiment
- 1 = Positive Sentiment

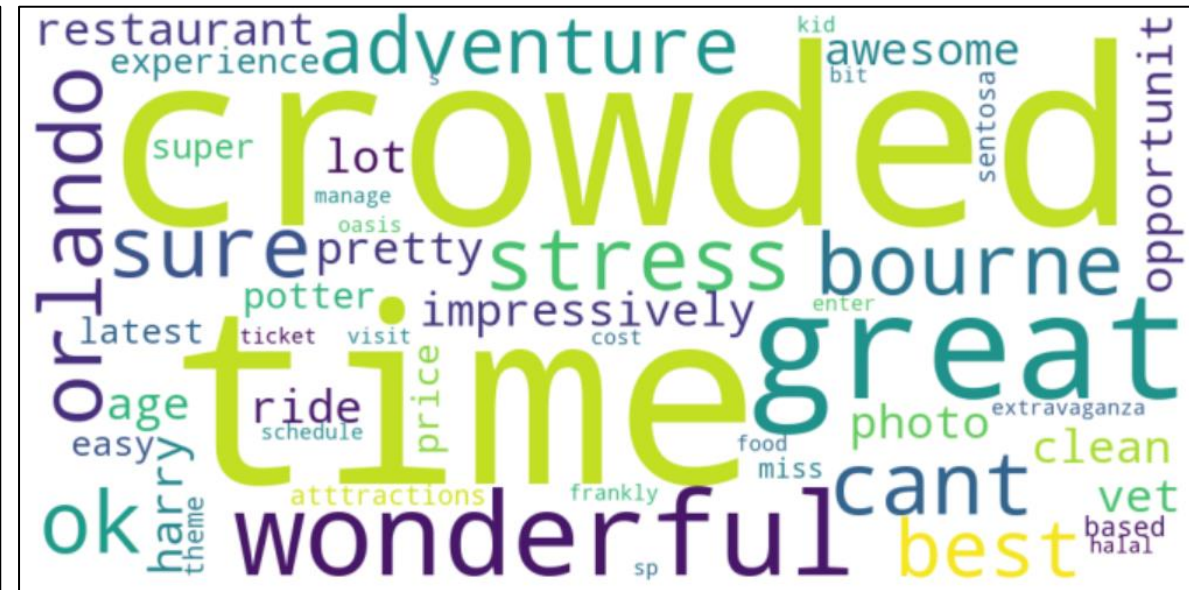
5.3. Process Workflow

- Data Analysis (Word Cloud)

Negative Sentiment:

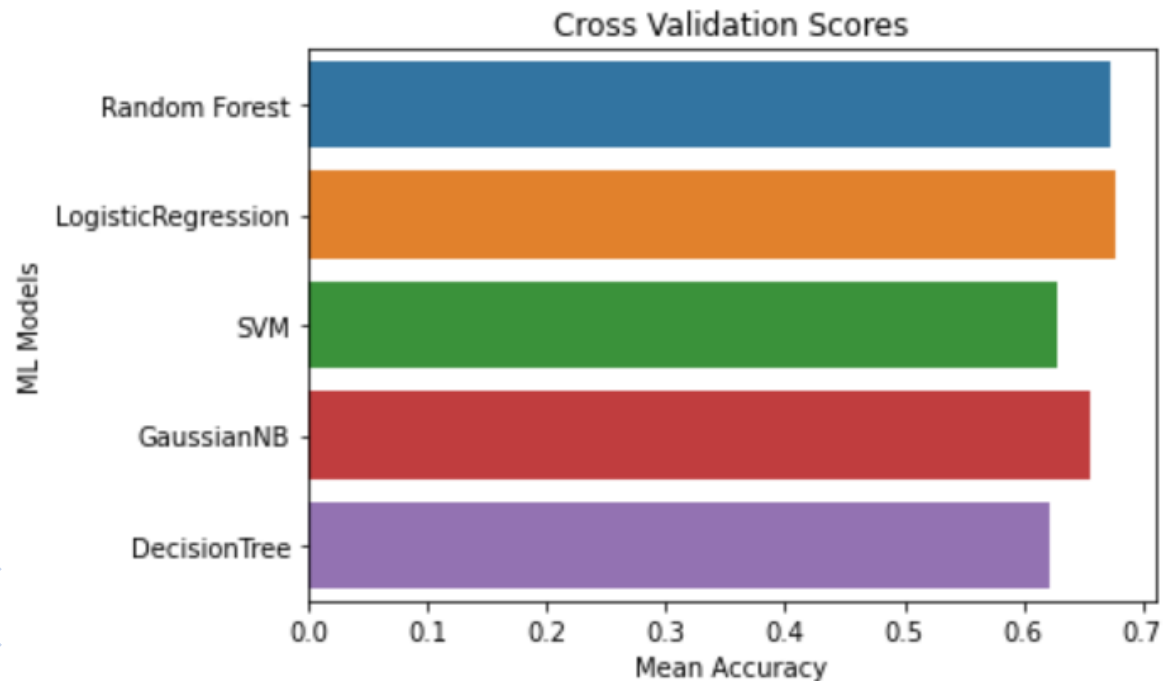


Positive Sentiment:



5.4. Process Workflow

- ML Model Training & Rvaluation
 - **Logistic Regression** - Best Model



Score	
Random Forest	0.673081
LogisticRegression	0.676659
SVM	0.628023
GaussianNB	0.656906
DecisionTree	0.621227

5.5. Process Workflow

- Hyperparameter fine-tuning for Logistic Regression:

```
# Hyperparameter fine-tuning for Logistic Regression on multi-class dataset
parameters = {'penalty': ['l1', 'l2', 'elasticnet', 'none'],
              'C': np.logspace(-2, 2, 5)}

gs_clf = RandomizedSearchCV(LogisticRegression(multi_class='ovr', max_iter= 100000),
                             parameters,
                             cv=5,
                             # scoring='f1_macro',
                             scoring='roc_auc_ovr',
                             n_jobs=-1)
_ = gs_clf.fit(X_test_clean, y_test)

print(gs_clf.best_estimator_)
print(gs_clf.best_params_)
print(gs_clf.best_score_)

print("'{}' gives the best F1-score at: {:.2%}".format(gs_clf.best_params_, gs_clf.best_score_))
```


6.1. Results

- ML model Test & Evaluation
 - F1 - Score: **68.09 %**

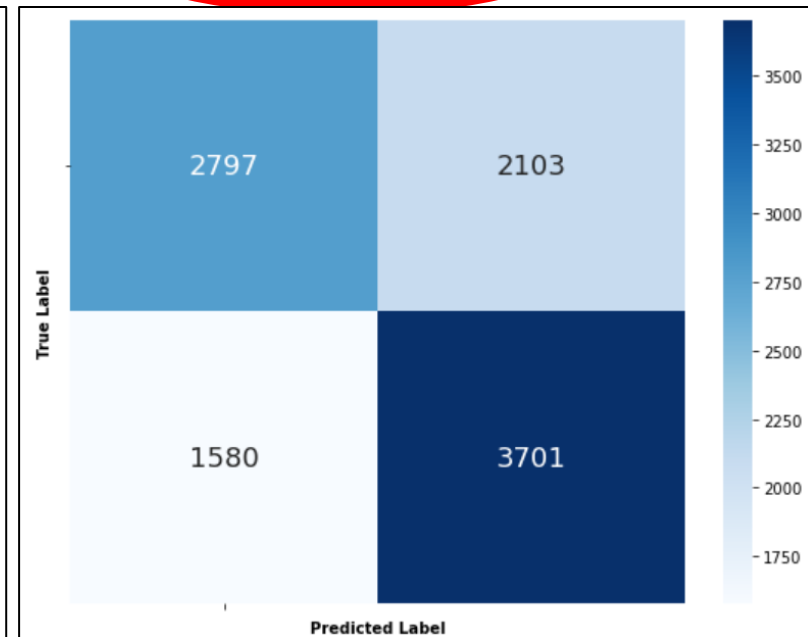
```
LogisticRegression(C=0.1, max_iter=100000, multi_class='ovr', penalty='none')
{'penalty': 'none', 'C': 0.1}
0.6809420490856961
{'penalty': 'none', 'C': 0.1} gives the best F1-score at: 68.09%
```

Classification report:

	precision	recall	f1-score	support
0	0.64	0.57	0.60	4900
1	0.64	0.70	0.67	5281
accuracy			0.64	10181
macro avg	0.64	0.64	0.64	10181
weighted avg	0.64	0.64	0.64	10181

Confusion Matrix:

```
array([[2797, 2103],
       [1580, 3701]], dtype=int64)
```



6.2. Results

- Prediction using observations from Test Model

	Desired Output (Actuals)	Predicted Output
40686	1	1
14285	0	1
25004	0	0
38742	1	1
16573	1	0
5748	0	0
15714	1	1
34726	0	0
21394	0	1
18968	1	1
50064	1	1
33921	1	1
43051	1	0
9798	1	1
7515	1	1
33942	0	1

```
In [131]: x = ['unfortunate weekend stressful experience horrible food worst service outdated park']
          x = clean_text(x)
          vec = Tfidf.transform([x])
          gs_clf.predict(vec)

Out[131]: array([0], dtype=int64)

In [132]: x = ['Best wonderful park']
          x = clean_text(x)
          vec = Tfidf.transform([x])
          gs_clf.predict(vec)

Out[132]: array([1], dtype=int64)
```


7.1. Conclusions

- What are the results and insights from the data analysis?
 - Rides are the main draw for all parks
 - In Florida and Japan, the Harry Potter themed area is most popular
 - In Singapore, Transformer ride is most popular
 - Main reason for low rating is long wait time in queues and lines



7.2. Conclusions

- Recommendations :
 - Constant update, revamp of rides
 - Higher prices and more variety of Harry Potter and Transformer merchandises
 - Entertainment and Hai Di Lao style customer services for long queue



- Power BI Dashboards screenshots:

