



PROJECT MLE

Damien Le Dirach - August 2021

Presentation

- I. Introduction
- II. Modeling / Notebook
- III. API
- IV. API test
- V. Docker
- VI. Kubernetes
- VII. Conclusion

I - Introduction

1) Problem

- Put a sentiment analysis model into production
- Models are not retrained in production
- Develop an API that allows:
 - ✓ To question the models
 - ✓ Access the performance of models
 - ✓ Authenticate users

2) Presentation of the dataset

3) Development environment

I - Introduction

1) Problem

2) Presentation of the dataset

- Kaggle
- The dataset includes 42,000 reviews of 3 Disneyland parks (Paris, California and Hong Kong)
- Columns:
 - ✓ Review_ID: unique id given to each review
 - ✓ **Rating**: ranging from 1 (unsatisfied) to 5 (satisfied) => **target**
 - ✓ Year_Month: when tea reviewer visited tea theme park
 - ✓ Reviewer_Location: country of origin of visitor
 - ✓ **Review_Text**: comments made by visitor => **feature**
 - ✓ Disneyland_Branch: location of Disneyland Park

3) Development environment

I - Introduction

1) Problem

2) Presentation of the dataset

3) **Development environment**

- Linux experience with mini projects
- Transposition in a Windows environment with the IDE **Visual Studio Code**:
 - ✓ intelliSense (completion code ..)
 - ✓ Debugging (step by step, breakpoints ...)
 - ✓ Docker integration (with Docker Desktop)
 - ✓ Azure Cloud Integration
 - ✓ Many extensions (snippets...)

II - Modeling / Notebook

1) **Notebook Supplied**

- Partitioned data in training / test set
- **CountVectorizer**
 - Trained / applied on training set
 - To be applied to the test sets (ie column " Review_Text ")
 - Representation of text as a vector
- **4 Models**
 - LogisticRegression() applied to all training data
 - RandomForestClassifier(n_estimators= 20, max_depth= 5) applied respectively to HK / California / Paris parks

2) Adaptation of the notebook to the problem

II - Modeling / Notebook

1) Notebook Supplied

2) **Adaptation of the notebook to the problem**

- 2 pickle files generated for each model within the API:
 - **CountVectorizer**
 - **Pre-trained models**
- 2 other global files:
 - NLTKWordTokenizer.pkl
 - stopwords.pkl

III - API

1) Endpoints



"/ : returns {score: 1} if the API is working

"/get_username" : return it username currently used [authentication *]

"/get_performance" : returns the model score

"/text_to_sentiment/" : returns the prediction (score from 1 to 5) associated with the text entered [authentication *]

2) Class ModelFromFiles

3) Using a Type Enum

```
class EnumModel(IntEnum):  
    AllBranch = 1  
    HK = 2  
    California = 3  
    Paris = 4
```

- **Charger un modèle** (parmi les 4 existants) à partir des fichiers pré-enregistrés « count_vectorizer{i}.pkl » et « model{i}.pkl »
⇒ def _load_from_pickles_files(self)
- **Pré-traiter un texte** afin de le rendre exploitable par le modèle
⇒ def preprocess(self, text, pkl_stopwords, pkl_tokenizer)
- **Effectuer une prédiction** à partir d'un texte
⇒ def predict(self, text, pkl_stopwords, pkl_tokenizer)

IV - API test

1) Manual test [FastAPI - Swagger UI \(disneyreviews.azurewebsites.net\)](https://disneyreviews.azurewebsites.net)

2) Description of the test script



3) Tests carried out

```
#test du endpoint « / » => [code HTTP attendu = 200, score attendu = 1]
    '####': (200, '1')

#test => bad username, 401 attendu
    'text_to_sentiment#{sentence}#1#alice1#wonderland'.format(sentence=lst_sentences[0]): (401, '')
```

V - Docker

1) API Docker



2) Test script docker




3) Docker Compose



```
=====
API test
=====
Request done at "/text_to_sentiment"
| sentence=Visited 21 5 2014. This park is a joke, three main rides were closed in one park...
| model_index=1
| username="alice"
| password="wonderland"
=> Test(expected vs actual) / HTTP Status: 200 vs 200 / Score: [1] vs [1] ==> success
```

VI - Kubernetes

1) Deployment 

2) Service 

3) Ingress 

VII - Conclusion

- <https://disneyreviews.azurewebsites.net/docs#/>
- Outlook:
 - GitHub Actions workflow
 - Container with MySQL DB for username/password
 - Deployment Kubernetes Azure
 - Logging
- Questions / Comments?