# Data science for developers


- Nicolas Vo
- Ulziiburen Dorjpurev
- Abhijeet Tiwari

http://esiee-imc.westeurope.cloudapp.azure.com/

# Introduction


Project: sentiment analysis for movie reviews

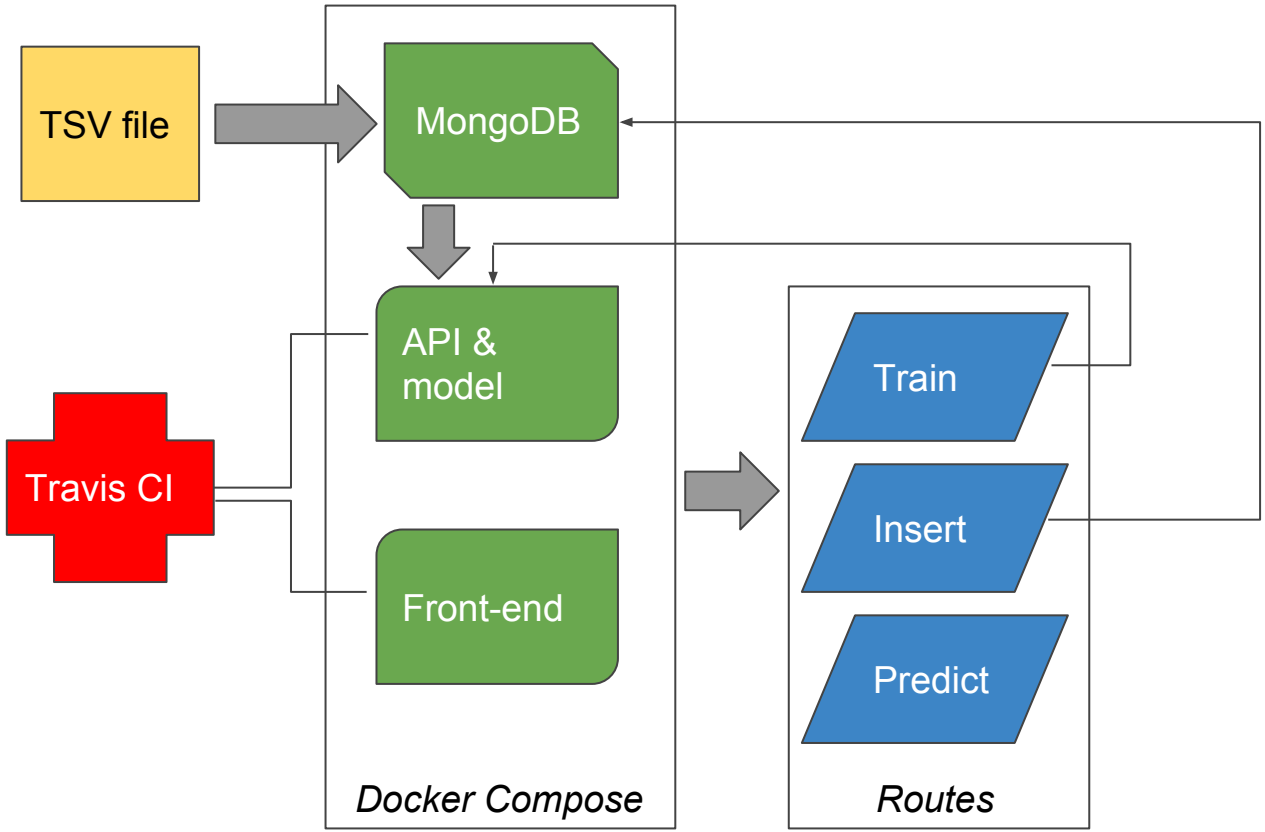
Uses:

- Understand user movie preferences and sell the data to Netflix
- Help users find movies they like/dislike based on the reviews they received

Routes:

- Predict if a sentence is positive or negative
- Insert new sentences and polarities
- Retrain the model

# Project architecture




Train:
- No parameter

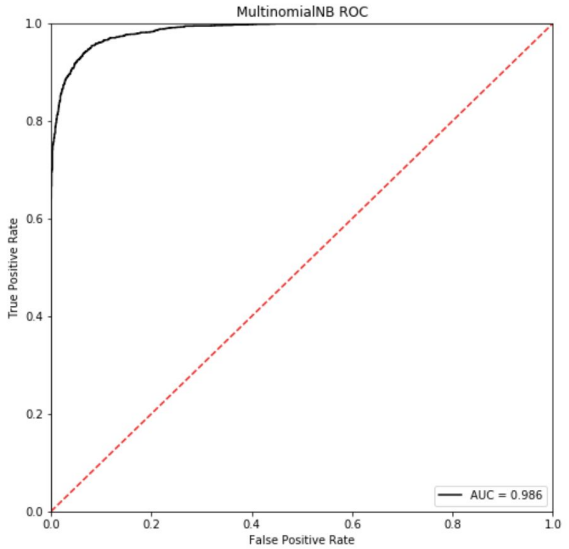
Insert:
- Sentence
- Polarity

Predict:
- Sentence

# Data and model

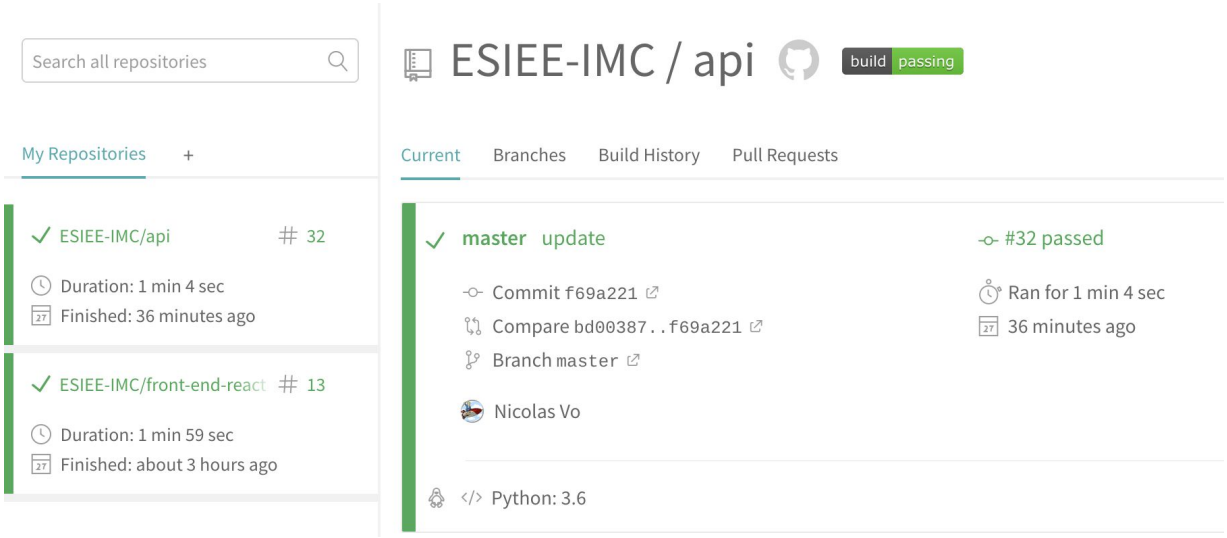
- 156 000 sentences
- Polarity between 0 and 4 (negative, neutral, positive)
- Preprocessing: lowercase, accentuation, stopwords
- Vectorization: TF-IDF
- Classifier: Naive Bayes
- Evaluation: ROC

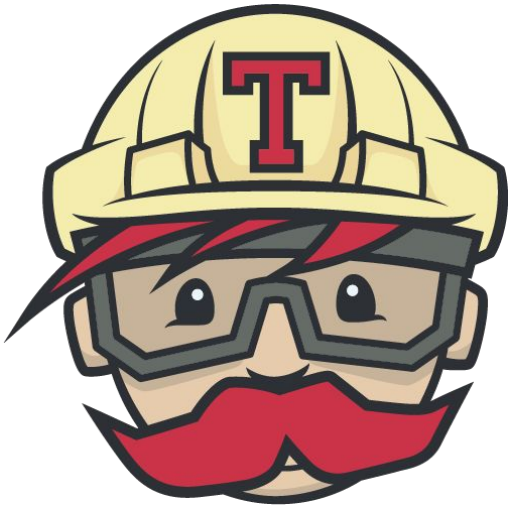
Source: Kaggle



# CI/CD

- Travis CI
- Free and no personal servers required
- Builds API and front-end on push and pull request

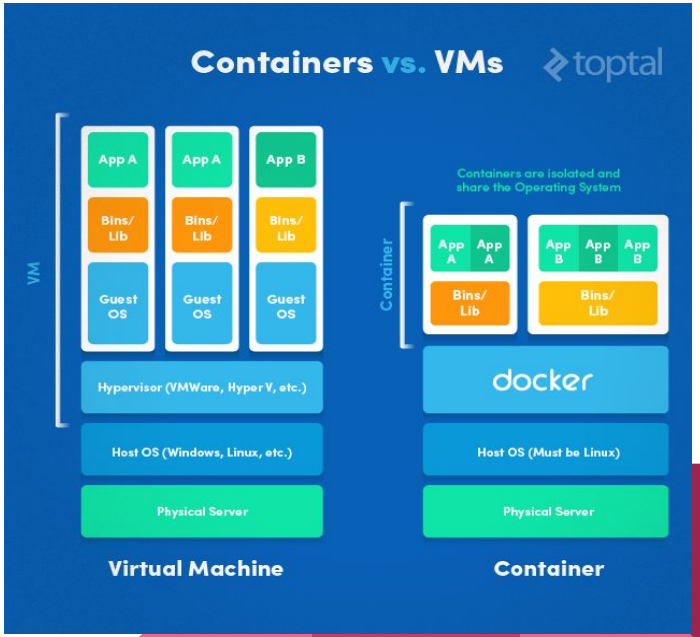


# Containerization

Aim: isolation, security, ease of deployment, scalability

- Docker Compose for container orchestration
- Microsoft Azure VM
- Flask API, Mongo database, ReactJS front-end



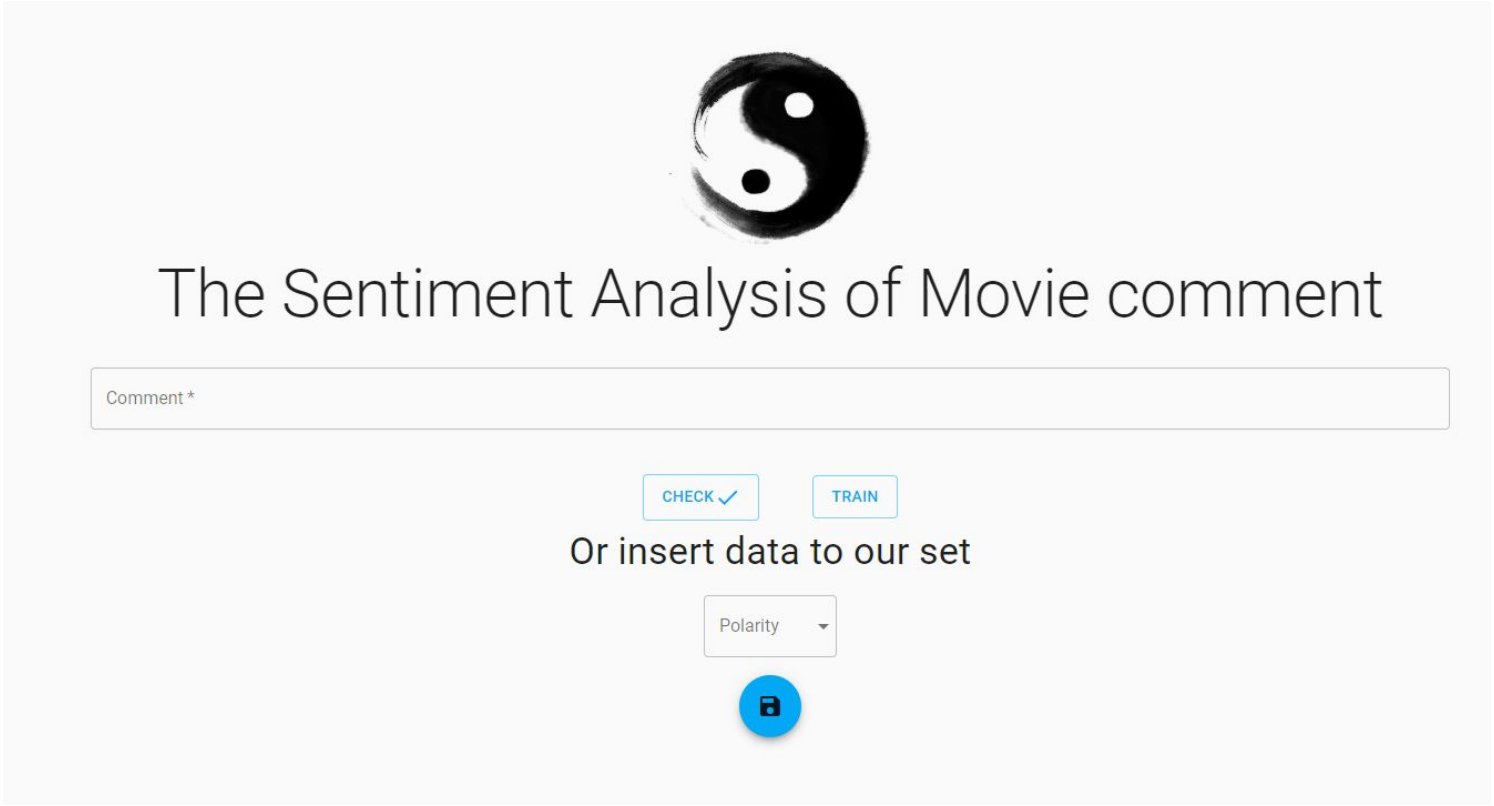
# API

1) Predict a sentence
2) Insert a sentence and its polarity
3) Retrain the model

Future improvements:

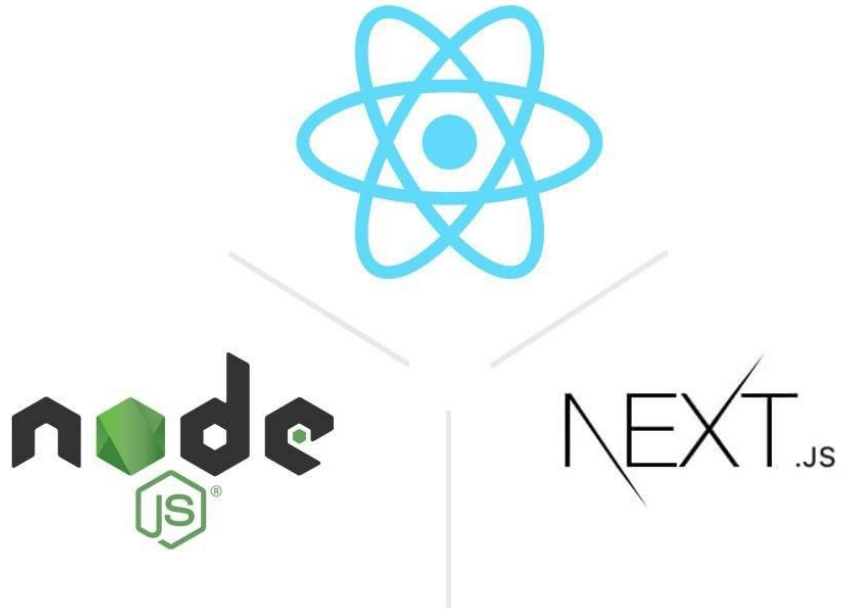
- Delete, update
- Database exploration (tables, graphs)
- Archive classifiers and vectorizers
- Evolution: API versioning

# UI


The Sentiment Analysis of Movie comment
Comment *
CHECK
TRAIN
Or insert data to our set
Polarity

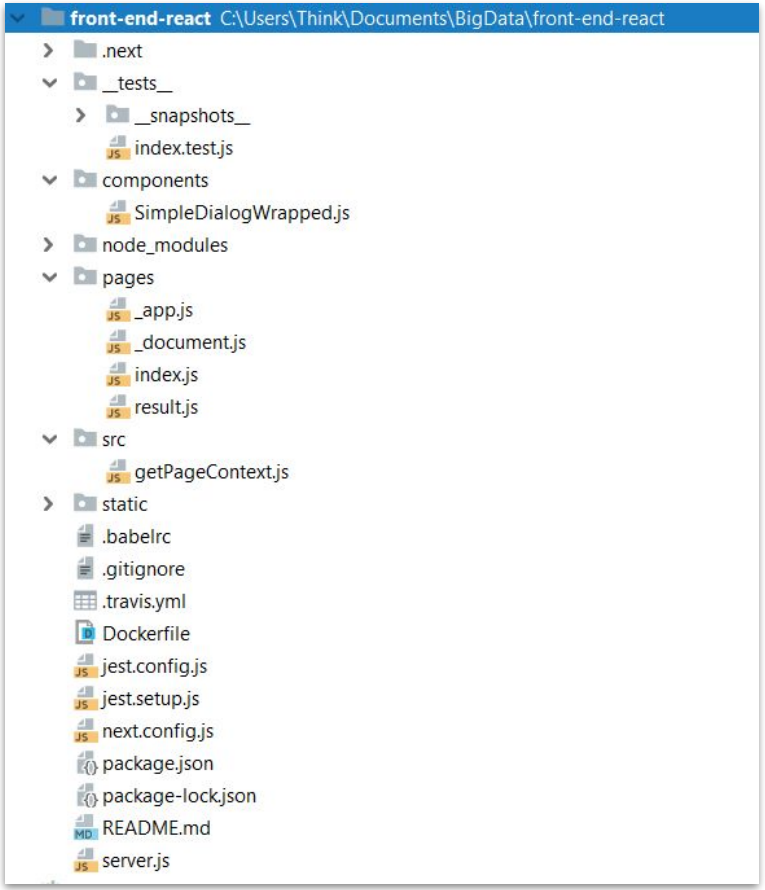
# Front-end

- ReactJS
- NextJS
- ExpressJS
- Material UI



# Project structure

- Unit test with Jest
- Travis CI
- Docker
- ES6



# Conclusion

1) Data and model with a business value
2) API using Flask RESTful
3) UI with ReactJS
4) CI/CD and Agile methodology
5) Containerization and cloud deployment