Sentiment Analysis on IMDb Movie Reviews: Word2Vec with Naive Bayes and CNN

Urvi Suwal and Isha Katwal

INTRODUCTION

- People often use film-rating websites like IMDb to decide if a movie is worth viewing
- Sentiment Analysis on IMDb movie reviews is used to identify the sentiment or opinion expressed by a reviewer towards a movie.

- **Problem Statement**
  Our project aims to implement sentiment analysis on IMDb movie reviews using word2vec(a popular method for word embeddings) and text classification using CNN and Naive Bayes to classify IMDb movie reviews as either “Positive” or Negative”.

BACKGROUND

Word2Vec

- A method to generate vector representations, or word embeddings in a text corpus based on the context in which they appear
- The algorithm uses a neural network architecture that consists of two learning models: CBOW (continuous bag of words) and Skip-gram
- Developed by Google researchers in 2013

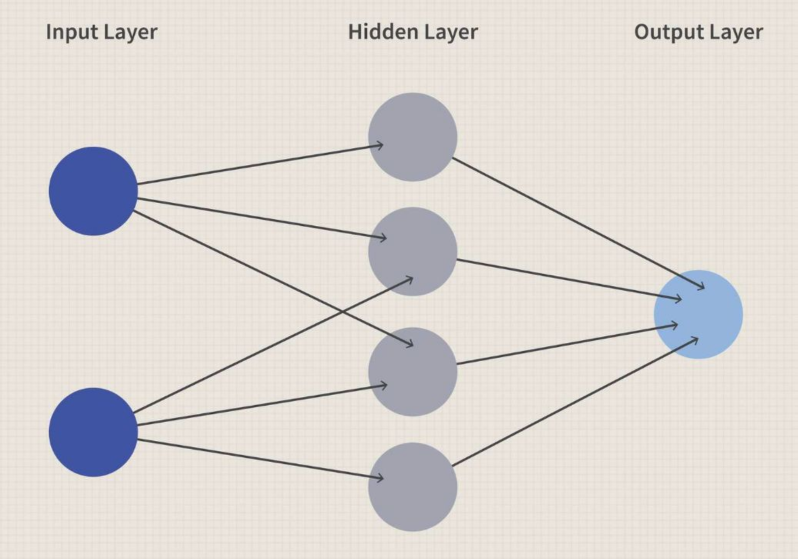
Gaussian Naive Bayes Classifier

- Based on the probabilistic approach and Gaussian distribution
- We assume all the continuous variables associated with each feature to be distributed according to Gaussian Distribution

BACKGROUND

Convolutional Neural Network

- A class of deep neural networks, most commonly applied to analyze visual imagery.
- Three main layers:
  - Convolutional layer
  - Pooling layer
  - Fully-connected (FC) layer



DATA

IMDb movie review dataset that contains 50,000 movie reviews, created by Andrew Maas, Stanford University

- 25,000 highly polar movie reviews for training and 25,000 for testing
- Retrieved a .csv dataset file from kaggle.com

IMDb

MATERIALS

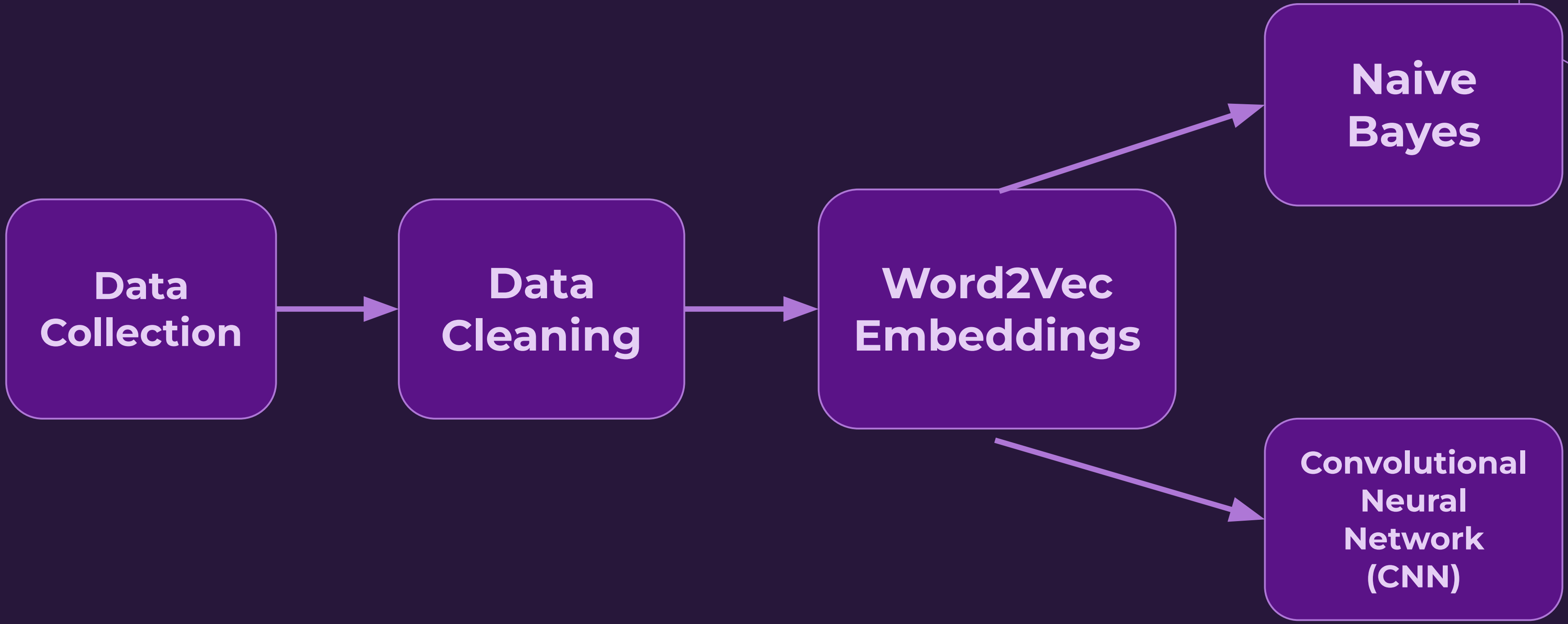
TensorFlow

Keras

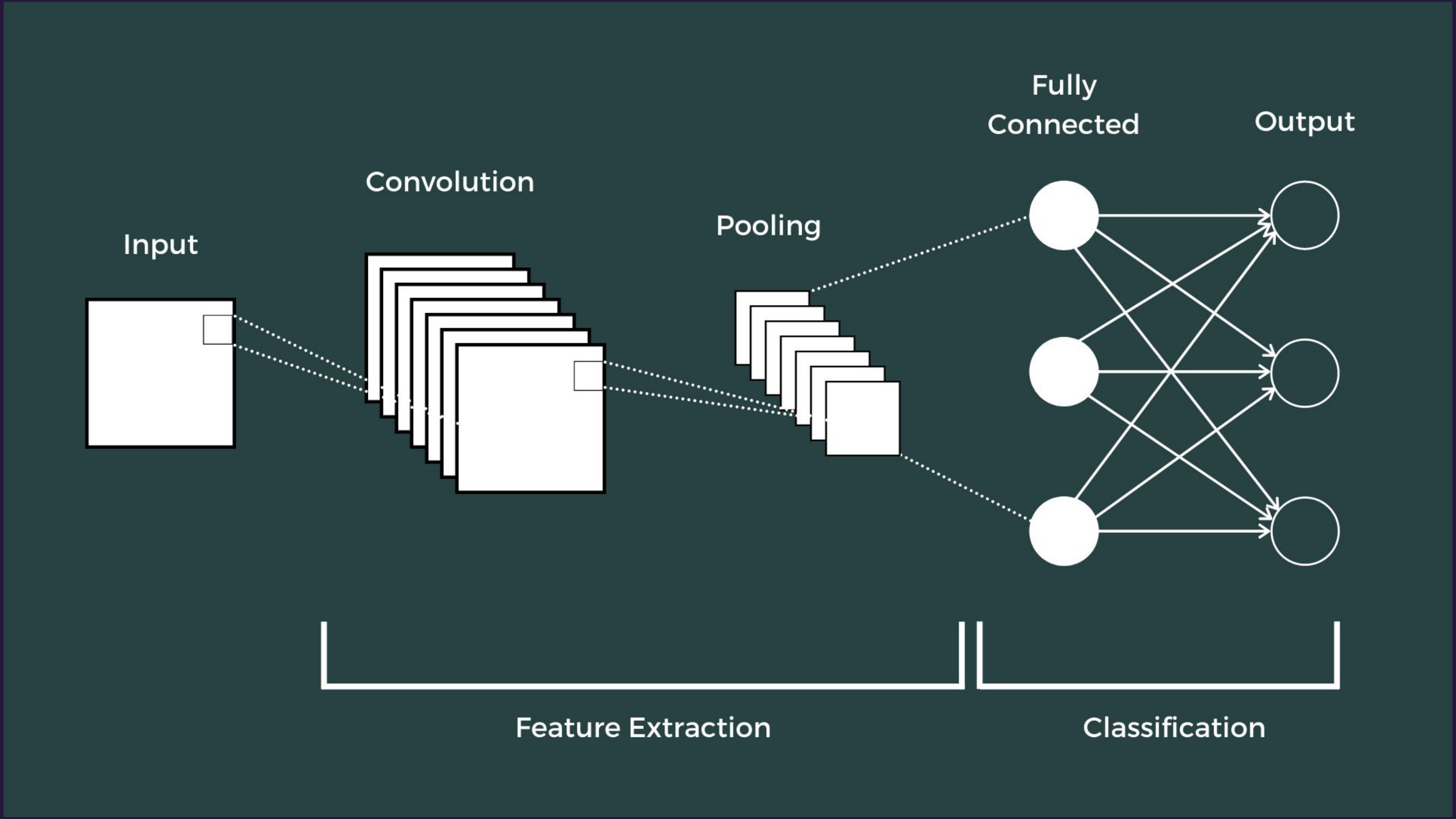
Gensim

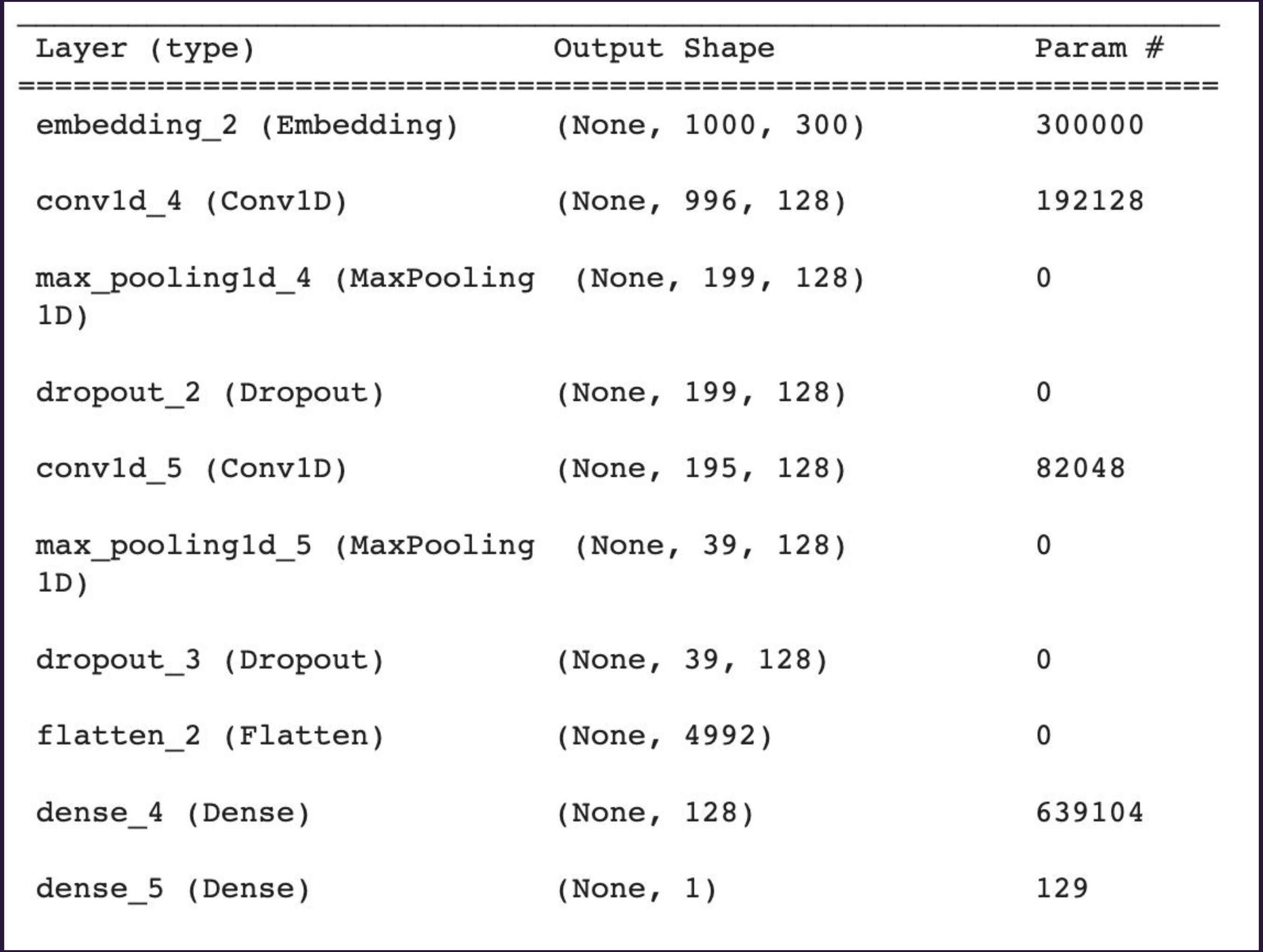
scikit -learn

WORKFLOW


Data Collection
Data Cleaning
Word2Vec Embeddings
Naive Bayes
Convolutional Neural Network (CNN)

CNN Architecture





```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_2 (Embedding)     (None, 1000, 300)         300000

 conv1d_4 (Conv1D)           (None, 996, 128)          192128

 max_pooling1d_4 (MaxPooling  (None, 199, 128)         0
 1D)

 dropout_2 (Dropout)         (None, 199, 128)          0

 conv1d_5 (Conv1D)           (None, 195, 128)          82048

 max_pooling1d_5 (MaxPooling  (None, 39, 128)          0
 1D)

 dropout_3 (Dropout)         (None, 39, 128)           0

 flatten_2 (Flatten)         (None, 4992)              0

 dense_4 (Dense)             (None, 128)               639104

 dense_5 (Dense)             (None, 1)                 129
```

RESULT

Model	**Evaluation Metric**			
	Precision	Recall	F1	Accuracy
Naive Bayes	0.77	0.77	0.77	0.772
CNN (*predicted*)	~0.9	~0.9	~0.9	~0.9

CONCLUSION

- Overall, we believe that the model that uses CNN as a classifier will perform better than Naive Bayes
- This might be due to the fact that CNN utilizes spatial data
  - CNNs are able to capture contextual information, while in Naive Bayes words are independent of each other
- However, there are disadvantages in using CNNs:
  - Time
  - Cost: expensive computation
  - Requires large datasets
  - Sensitive to hyperparameters

Ongoing work

Currently we are working on the error analysis component of our project

- Analyzing incorrect predictions from either of the models (Naive Bayes and CNN) to identify patterns.