Project 3 - Group 5 (Ben Paffrath)

Fake news detection

Project 3

Executive summary

- Good results with simple machine learning models
- Linear SVC and Logistic Regression were the best

- Embedding Models showed good scores
- but the prediction of the real data does not seem to be correct

Linear SVC
Train Accuracy: 0.972
Test Accuracy: **0.907**

Embedding Model ?
Train Accuracy: 0.982
Test Accuracy: **0.949**

Preprocessing & Data cleaning

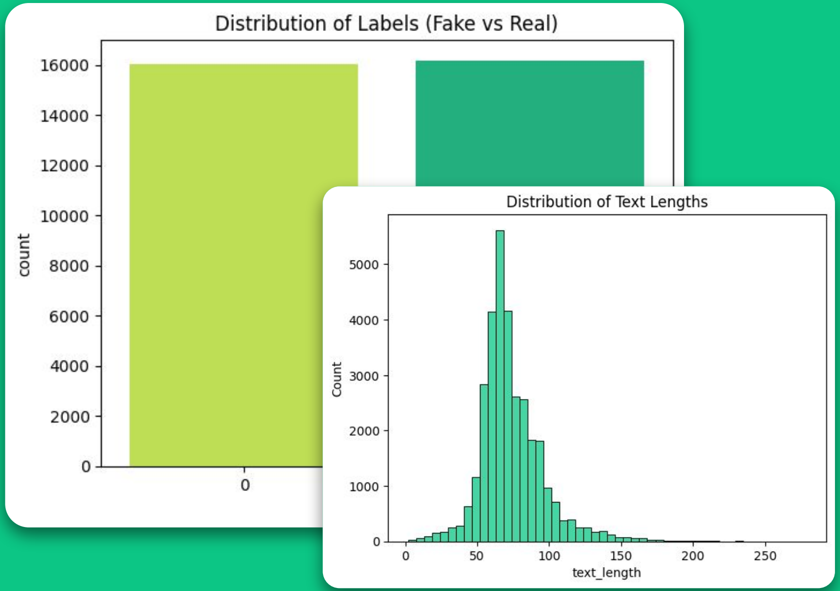
Preprocessing & Feature extraction

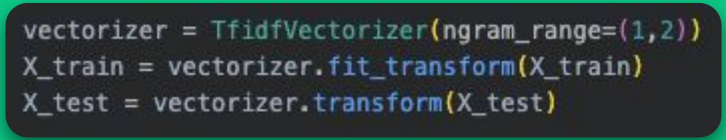
CSV data was pretty clean, so just some simple steps to prepare the data.

- Dropped duplicates (5.7 %)
- Removed stop words and punctuation
- Used Lemmatization

Feature extraction

- Classical ML Models: TfidfVectorizer
- Embedded Based: Tokenizer



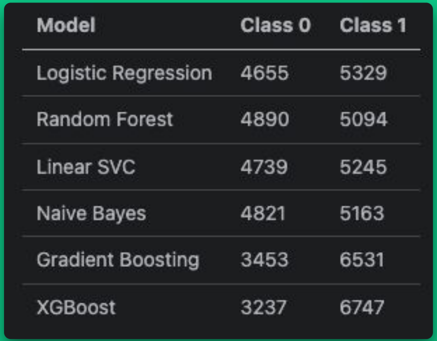


```
vectorizer = TfidfVectorizer(ngram_range=(1,2))
X_train = vectorizer.fit_transform(X_train)
X_test = vectorizer.transform(X_test)
```

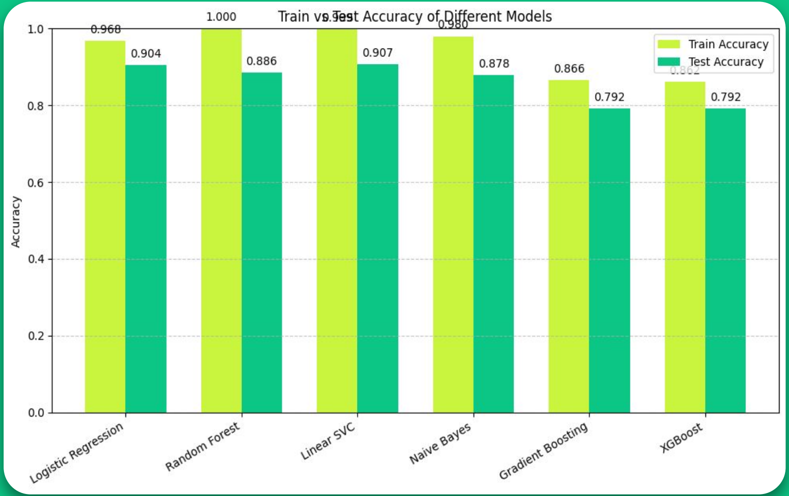
Trained Models

Classical Machine learning Models

- Used models with default parameters
- Simple models worked surprisingly well
- Gradient Boosting and XGBoost performed not as good as expected
- Boosting models tend to recognize significantly more fake news than the other models



Model	Class 0	Class 1
Logistic Regression	4655	5329
Random Forest	4890	5094
Linear SVC	4739	5245
Naive Bayes	4821	5163
Gradient Boosting	3453	6531
XGBoost	3237	6747



Learned / pre-trained Embeddings

Learned embedding:

An embedding layer trained from scratch during model training, without using any external pre-trained vectors.

```
Embedding(
     max_words=10000,
     output_dim=50,
     input_length=30
)
```

Pre-trained embedding (GloVe):

An embedding initialized with vectors pre-trained on large text datasets using the GloVe algorithm.

- Downloaded the GloVe file [50d, 100d, ...]
  `years 0.16962 0.4344 …`
- Used the word dictionary of the tokenizer
- Map vectors of GloVe
- Embedding matrix

```
weights=[embedding_matrix]
trainable=False
```

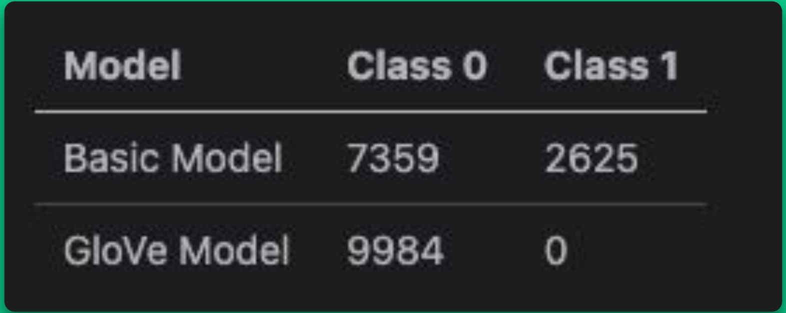
Trained Models

Embedding-based Neural Models

- Both models were good on the labeled test set
- But not good on real data

```
Sequential([
    Embedding(...),
    Bidirectional(LSTM(150,
...)),
    Dropout(0.2),
    LSTM(100),
    Dense(128,
activation='relu'),
    Dense(1,
activation='sigmoid')
])
```

?



Model	Class 0	Class 1
Basic Model	7359	2625
GloVe Model	9984	0

Project 3

Recap and Takeaways

I built different models, from simple ones to those using pre-trained embeddings, and achieved good results quickly with easy machine learning models.

- Start with easy models - they can get good results
- Neuronal Models are not easy to debug
- High accuracy score does not mean it predicts real life data well

FROM
SIMPLE
TO →
COMPLEXE

If there is still time...

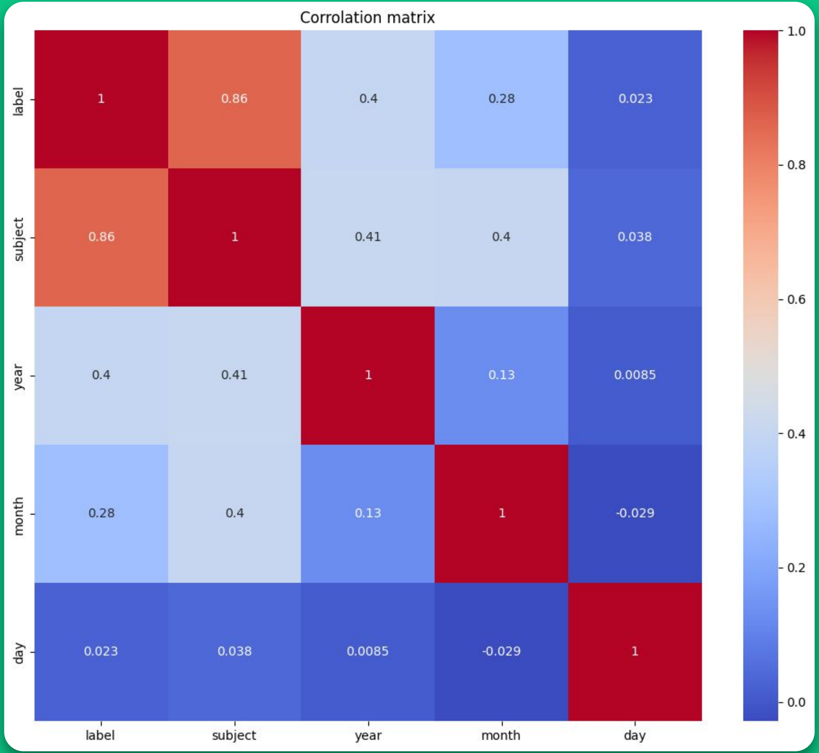
Working with wrong CSV-Data

label, title, text, subject, date

Problem:
Very high accuracy on test data (100%)

Finding:
Correlation matrix does not always show a correlation

Solution:
Removed nearly all columns



	label	subject	year	month	day
label	1	0.86	0.4	0.28	0.023
subject	0.86	1	0.41	0.4	0.038
year	0.4	0.41	1	0.13	0.0085
month	0.28	0.4	0.13	1	-0.029
day	0.023	0.038	0.0085	-0.029	1

If there is still time...

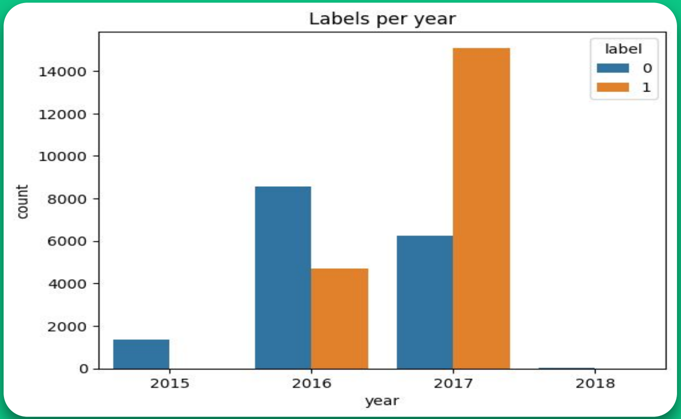
Working with wrong CSV-Data

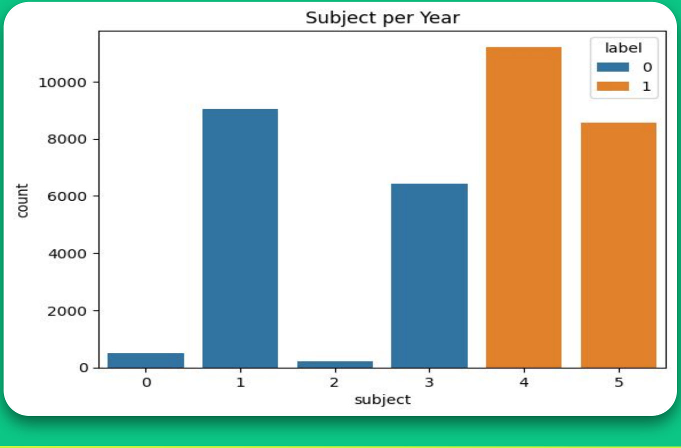
label, title, text, subject, date

Problem:
Very high accuracy on test data (100%)

Finding:
Correlation matrix does not always show a correlation

Solution:
Removed nearly all columns





Any Questions?

Group 5 (Ben Paffrath)