Classification
Fake and True News

Dataset

Dataset downloaded from kaggle

The procedure of collecting is unknown

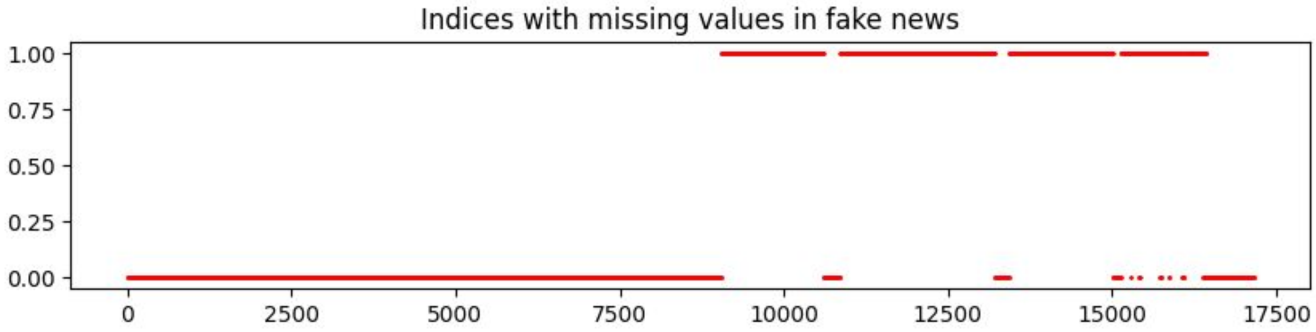
25% of duplicates in fake.csv, 5% in true.csv

99% of true data comes from reuters, 25% of fake data comes from twitter (comments to the post are included)

Scraping of fake dataset replaces apostrophes with white space (removal of lone letters)

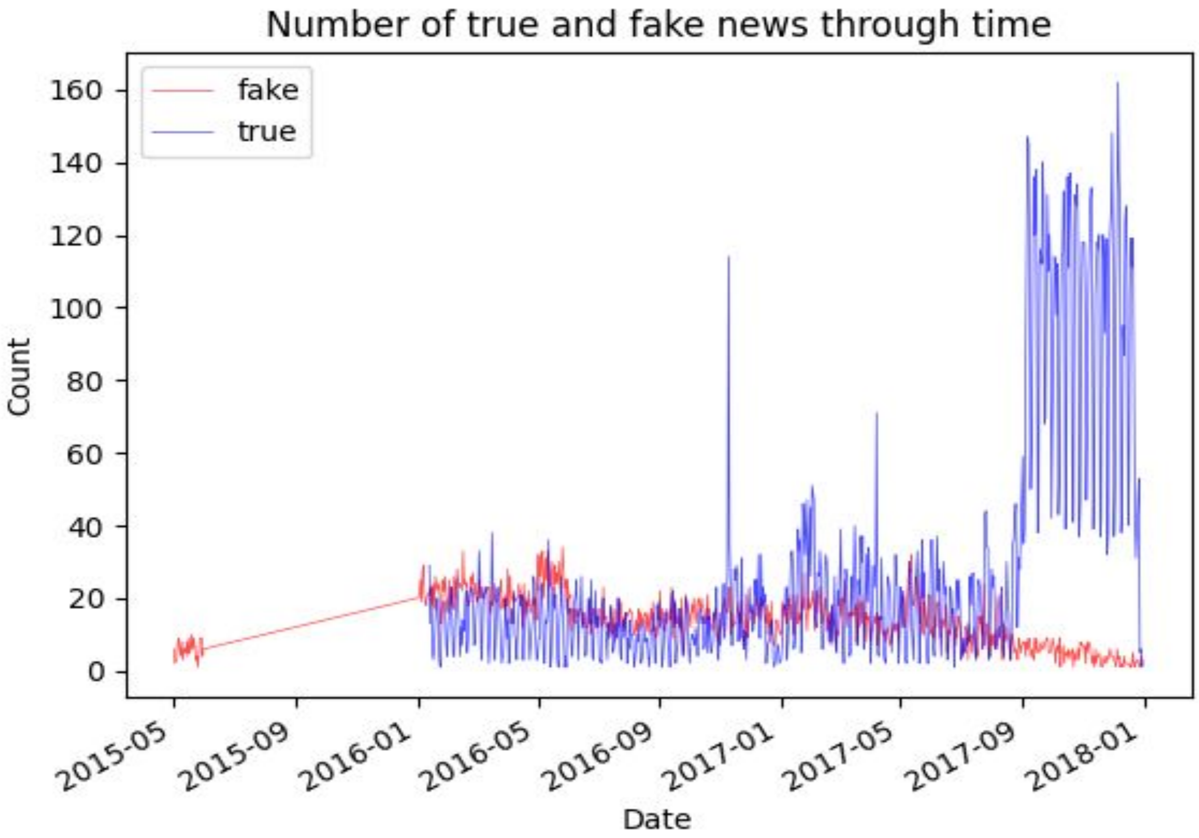
Dataset

40% of dates in fake news are missing, we can't confidently fill in with standard techniques




```
First value in fake news for date:
date    2017-12-31 00:00:00
Name: 0, dtype: object


Last value in fake news for date:
date    2016-01-02 00:00:00
Name: 17165, dtype: object
```

Dataset

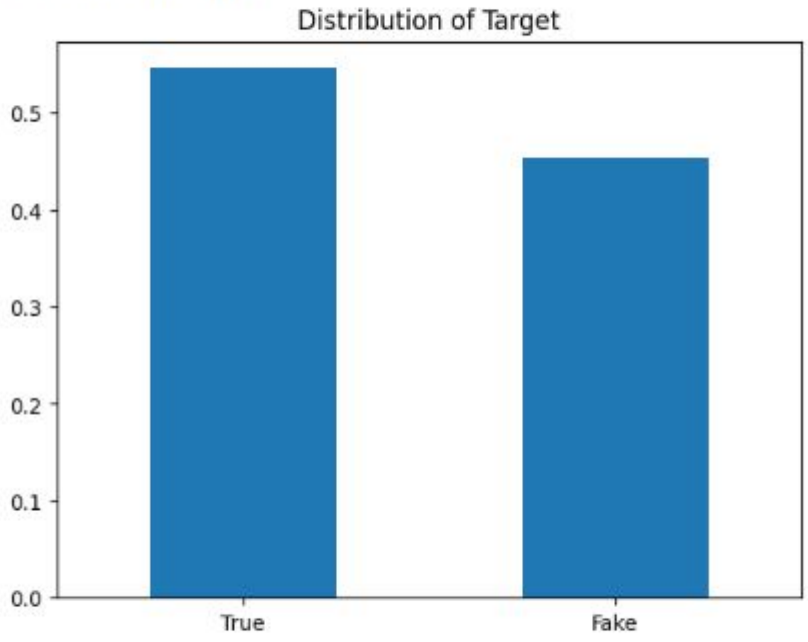


Number of true and fake news through time
fake
true
160
140
120
100
80
60
40
20
0
Count
2015-05
2015-09
2016-01
2016-05
2016-09
2017-01
2017-05
2017-09
2018-01
Date

Data Split

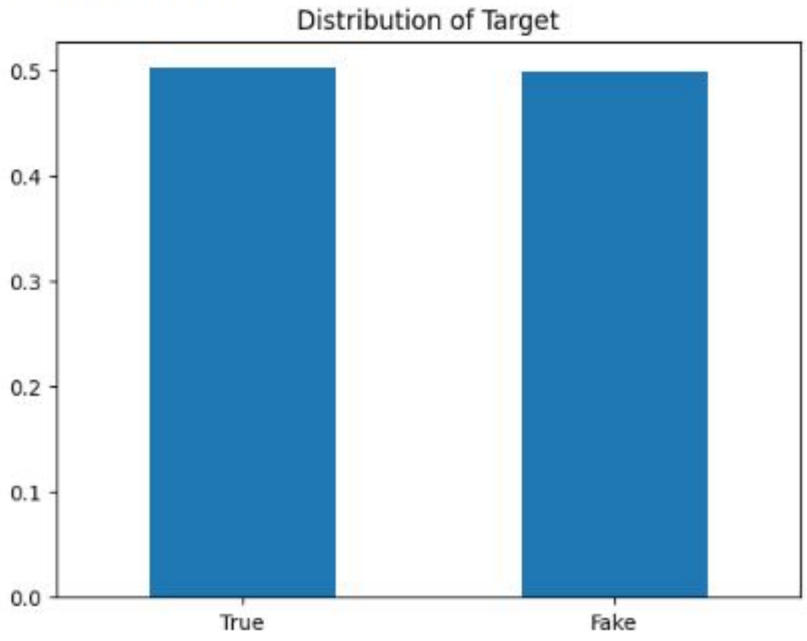
Stratified

```
Rows of training set: 29601
Rows of validation set: 3700
Rows of test set: 3701
```

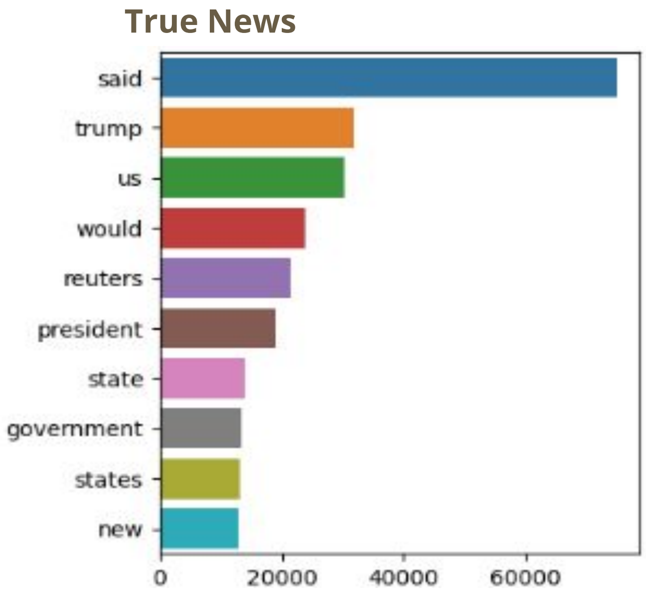


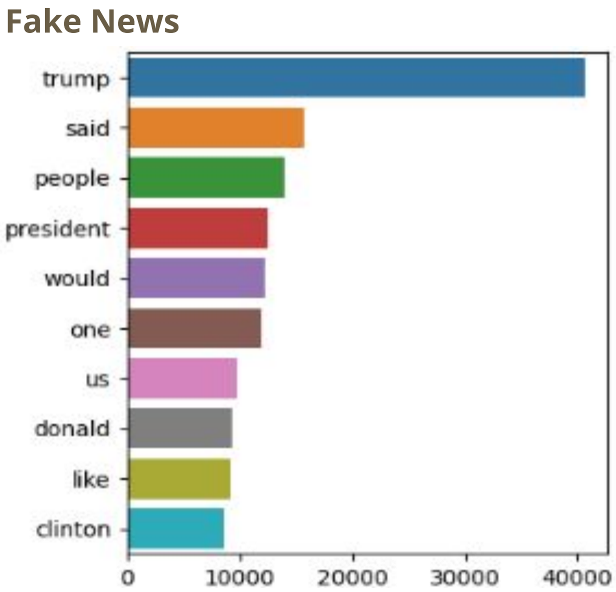
By Date

```
Rows of training set: 19690
Rows of validation set: 494
Rows of test set: 494
```



Most Common Unigrams in True and Fake News





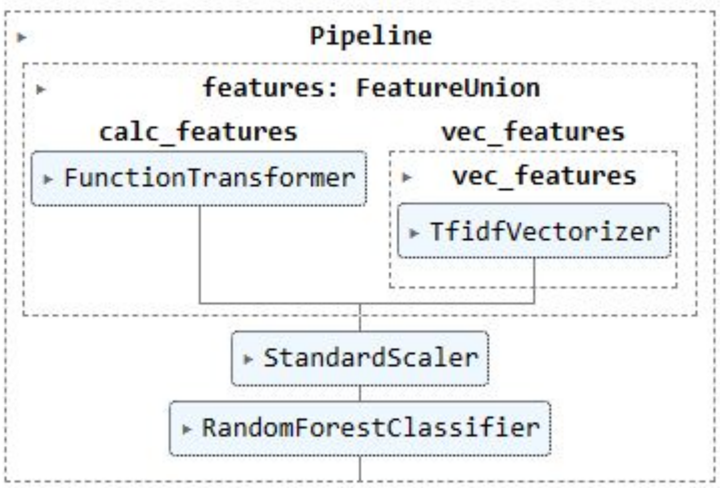
Most common words in both corpuses:

said, trump, us, would, president, people, reuters, one, state, new, also, states, house, government, donald, republican, could, united, told, clinton, obama, white, campaign, last, election, two, like, party, time, year

Analysis of Text Structure

Fake News	True News
Higher proportion of uppercase letters to grab attention	Consistent average word length
Higher variance in length	Slightly longer
Higher frequency of exclamation marks	Longer words - more sophisticated language
More outliers	Higher frequency of dots

Feature Engineering



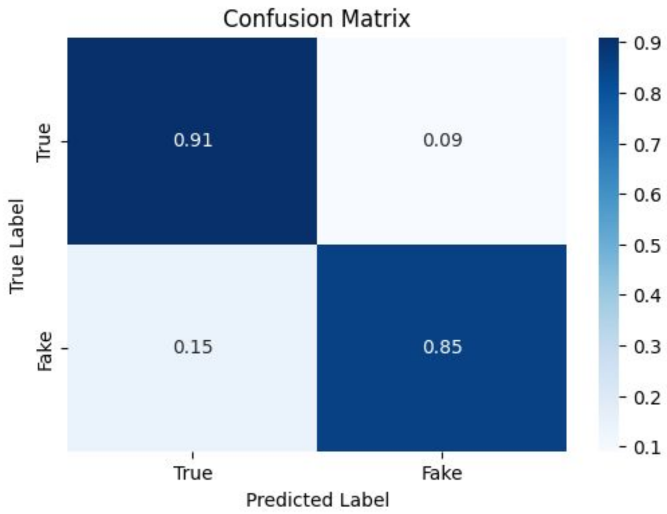
calculated features:

- Proportion of upper case letters
- Average word length
- Number of characters in news
- Count of words
- Proportion of punctuation
- Proportion of exclamation marks
- Proportion of dots

vectorized words:

- Said
- Would
- People
- One
- Also
- Could
- Last
- Two
- Like
- Time
- Year

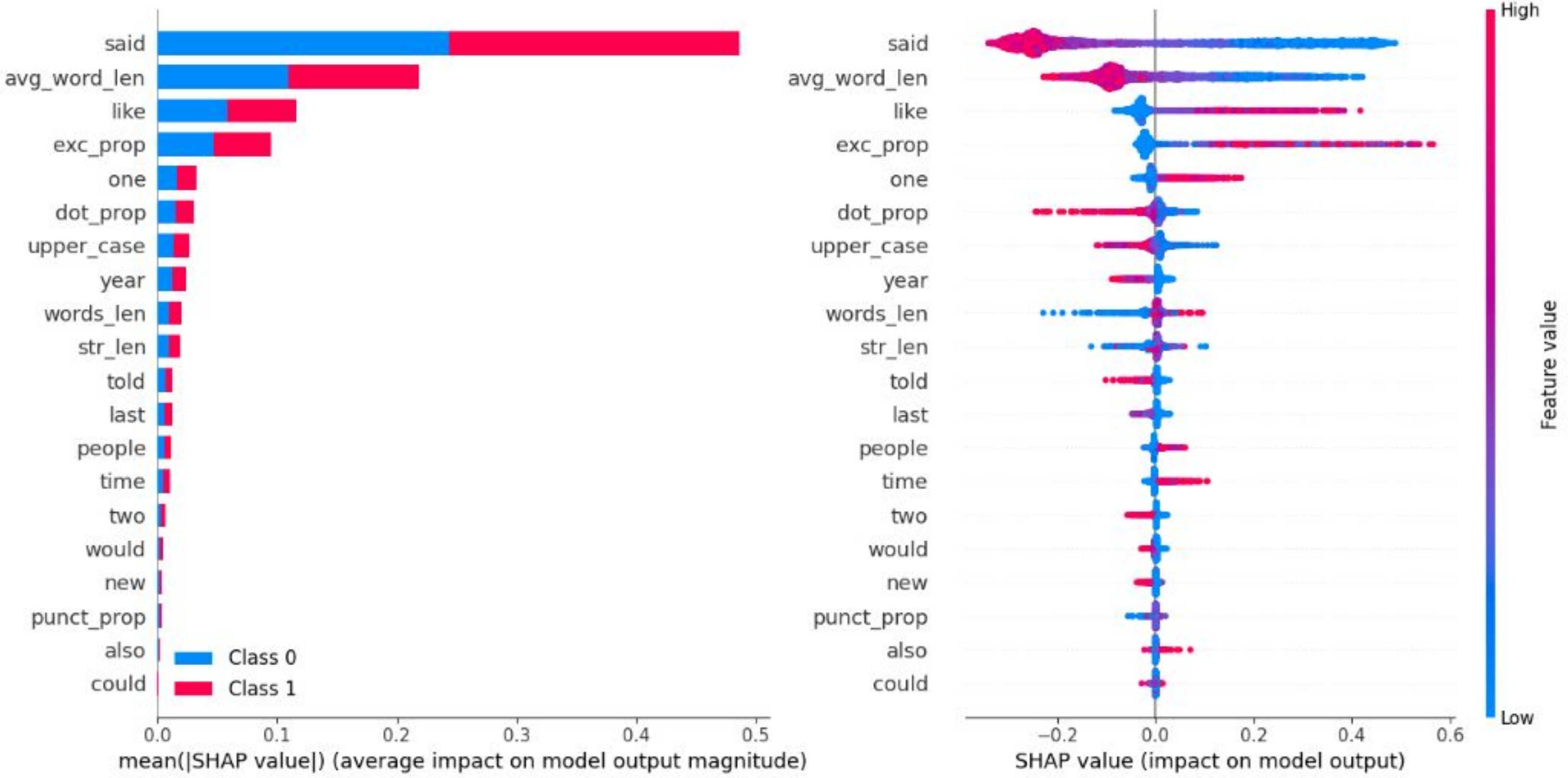
Modeling with Algorithms from Sklearn



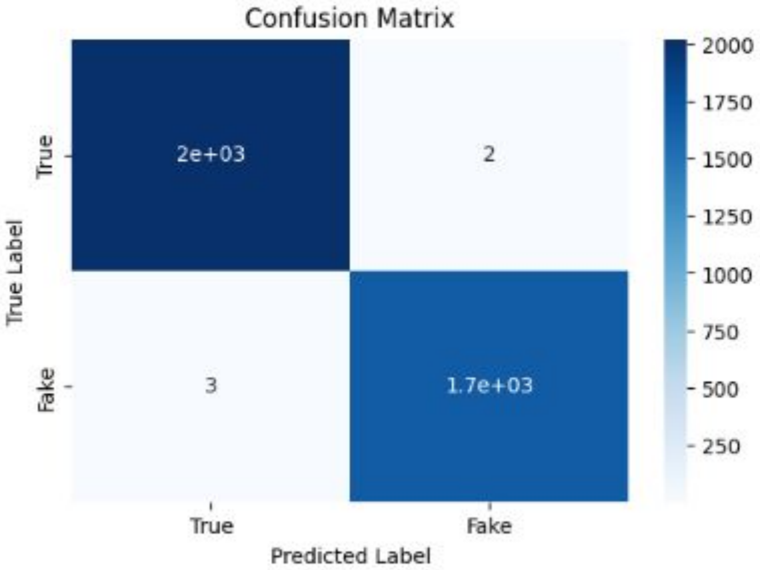

```
Classification Report:
              precision    recall  f1-score   support

        True       0.88      0.91      0.90      2023
        Fake       0.89      0.85      0.87      1678

    accuracy                           0.88      3701
   macro avg       0.88      0.88      0.88      3701
weighted avg       0.88      0.88      0.88      3701
```

- Best performing model - RandomForestClassifier
- Weighted f1_score on calculated features 0.77
- Weighted f1_score using all features on test set 0.88
- Adding more vectorized words didn't improve the outcome

Interpretation of Random Forest Classifier Predictions



Modeling *title* with Pretrained distilbert-base-cased



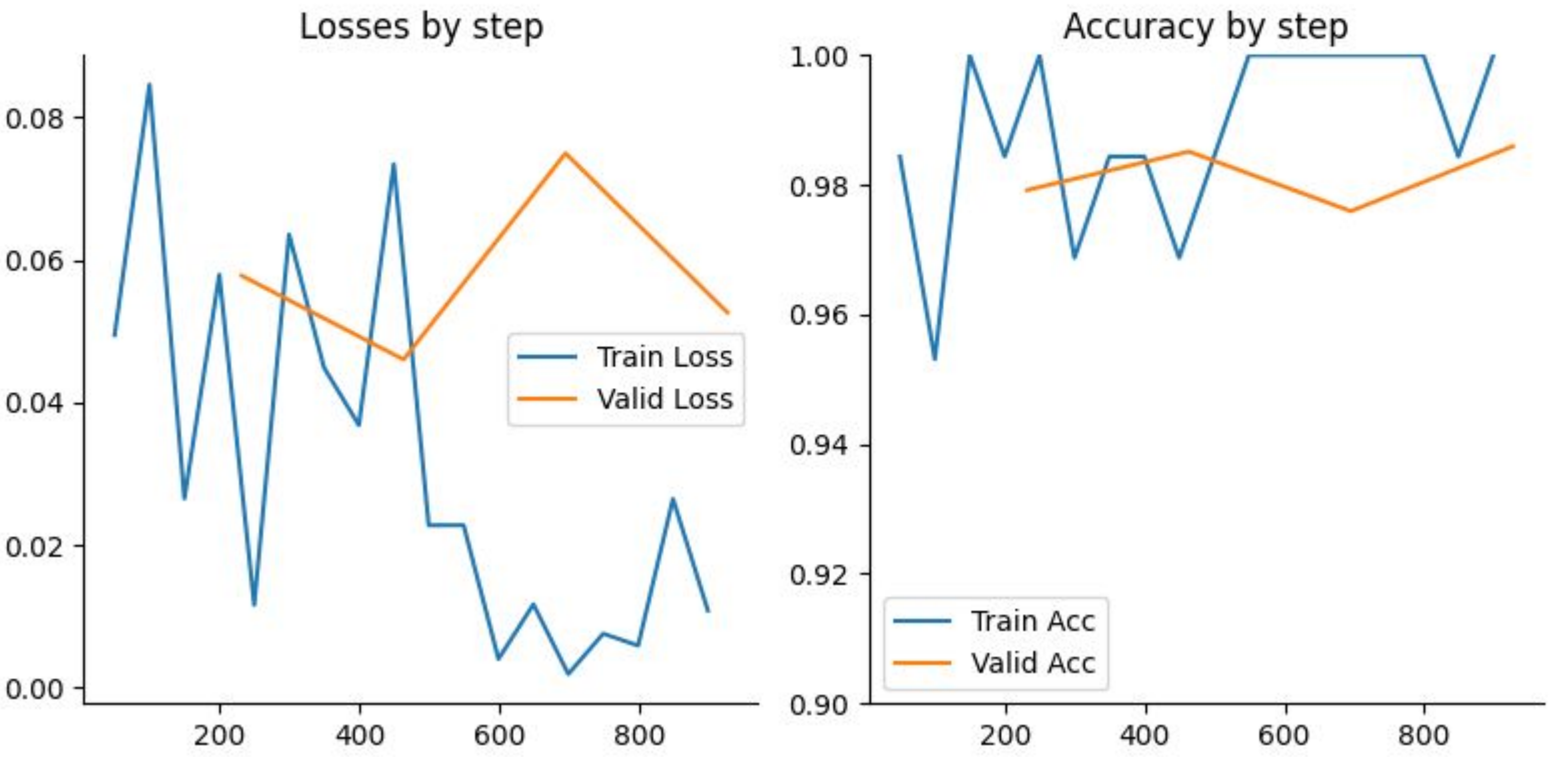
Classification Report:

	precision	recall	f1-score	support
True	1.00	1.00	1.00	2023
Fake	1.00	1.00	1.00	1678
accuracy			1.00	3701
macro avg	1.00	1.00	1.00	3701
weighted avg	1.00	1.00	1.00	3701

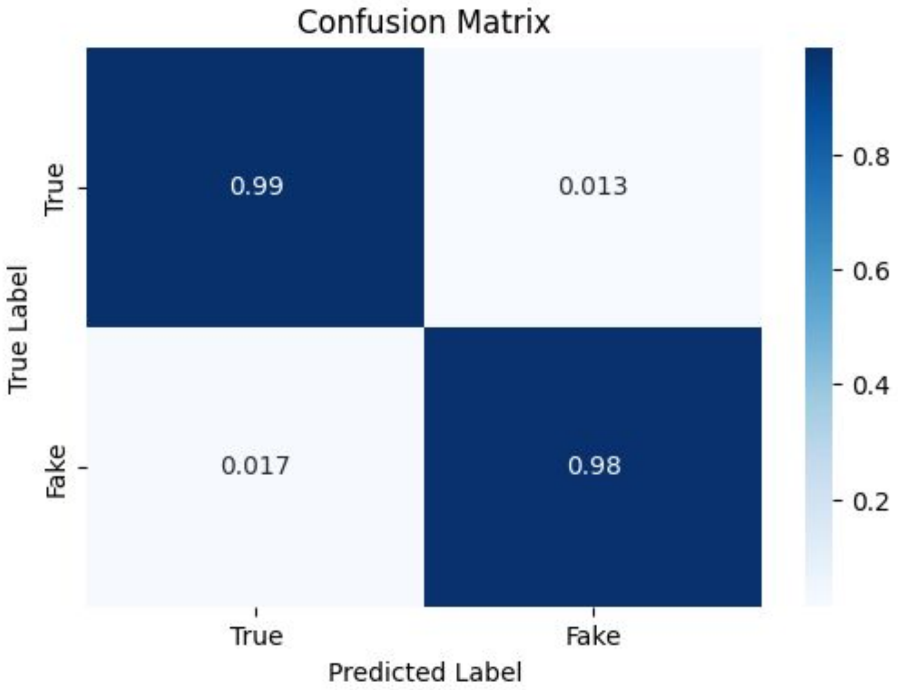
- The model was mistaken only 0.1% of the time.
- Training reached accuracy 1.00 on validation set after 1 epoch.

Examples of Wrongly Predicted Titles

Modeling *text* with Pretrained distilbert-base-cased



Performance of the model



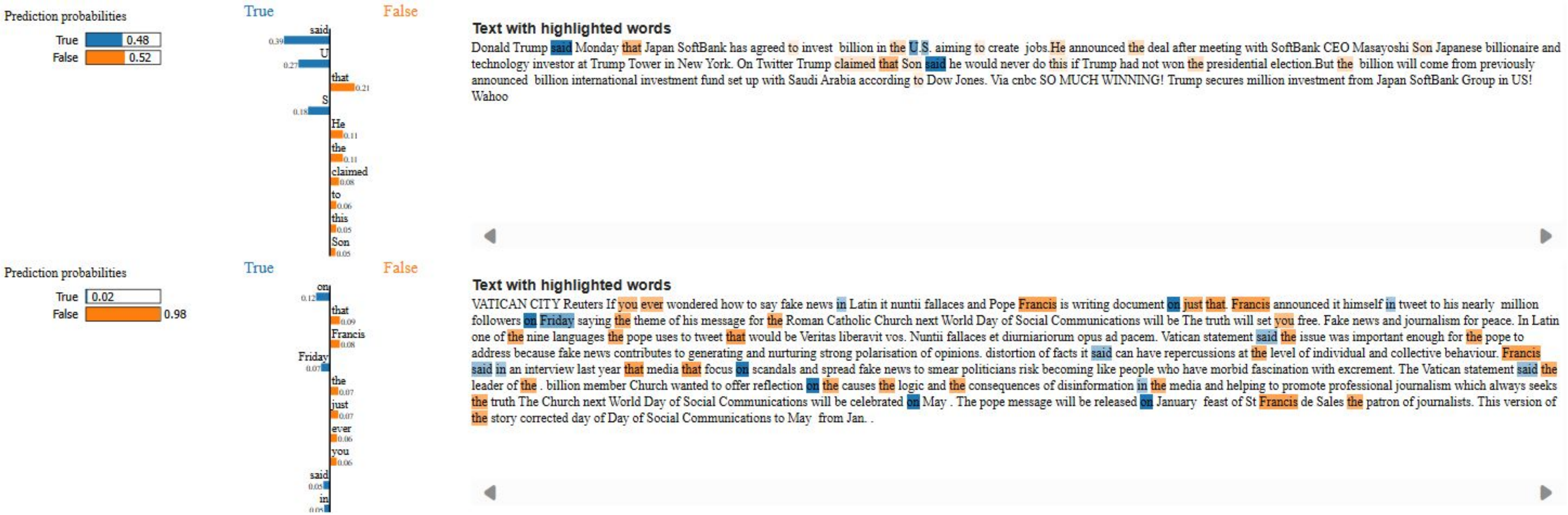
Classification Report:

	precision	recall	f1-score	support
True	0.99	0.99	0.99	2023
Fake	0.98	0.98	0.98	1678
accuracy			0.98	3701
macro avg	0.98	0.98	0.98	3701
weighted avg	0.98	0.98	0.98	3701

The model is wrong every 3 in 100 news.

Problems with correctly classifying fake news

Examples of Wrong Predictions



Conclusions

- Models rely on writing style of the writer
- The structure of title seems to be the most important factor of predictions
- True news have more reporting style of writing
- The size and performance of classic ML and NN models differs by 0.1 in weighted f1 score
- The source of dataset is not known.
- Estimating model performance on examples that are consistent with business are crucial
- Deployment of the model could depend on business integrity, ethics and financial resources

Improvements

- Checking for features that can leak data about the predictions.
- Further feature engineering and optimization of model hyperparameters.
- Estimating the performance of different sets of data to establish if models are performing as good as in the notebook.
- Experimenting with uncased models.
- Building LSTM model which would be smaller but hopefully performed similar.
- Estimating time of getting results and cost of deployment.