Amazon
Fake Reviews Detection Project

Yiting Hu, Yunlong Wang
DSO 560 - NLP Text Analytics

Presentation outline

1. Business Problem Background
2. Project Goal
3. Data Description
4. Implementation Roadmap
5. Modeling
    a. Vectorization
    b. Logistic Regression
    c. Neural Network
6. Potential Return on Investment
7. Future Steps
8. Conclusion

Business Problem Background

In the modern eCommerce shopping experience, customers' reviews play an important role. The reviews on the eCommerce website are useful for users and vendors. According to an online review study, roughly 90% of internet users consult online reviews before buying. Customers also like to write reviews to share their experience with the product. Positive reviews can help vendors increase brand awareness and loyalty. Likewise, a negative review can drive people away. With reviews being one of the decisive factors of purchasing decisions, unfortunately, the review mechanisms can be misused by vendors, users, or other parties for various purposes. There exist fake reviews on eCommerce platforms that mislead customers and impose trust issues.

Over many years, Amazon, our chosen project sponsor, has been devoted resources to prevent fake reviews from being seen by customers. In 2020, Amazon stopped more than 200 million suspected fake reviews, and they declared that they have continued to invent, deploy, and improve sophisticated technology in this area. In other words, it's a constant battle to fight.

Project Goal

In this project, we are working on behalf of Amazon to solve the business problem of fake reviews. Utilizing an existing Amazon review dataset, we aim to develop a machine learning model that seeks to represent and predict the authenticity of reviews.

The analysis is to interpret and classify subjective data (Amazon customer submitted reviews) using natural language processing and machine learning. The model will ideally increase the efficiency and accuracy of detecting unverified reviews that pose otherwise.

Data Description

	verified	reviewText
0	0	Since this is a standup act, i don't think the...
1	1	If you are a Don Adams Fan, this is the series...
2	0	If I could give this a zero, I would. Tom Cru...
3	1	Saw the movie years ago on t.v. when it 1st ca...
4	0	David A.R. White plays an FBI agent who finds ...

The dataset we used is Amazon Review Dataset (Movies and TV) released in 2014.

There are more than 3 million rows in the raw dataset. Due to the scope of this project, we randomly subset 100,000 rows to be used.

There are 12 features in the dataset, including reviewer ID, product ID, reviewer name, helpful votes, product style/format, review text, verified status, ratings, review summary/title, review time, review time in Unix, and image. For analysis purposes, we only use review text and verified status.

Data Description

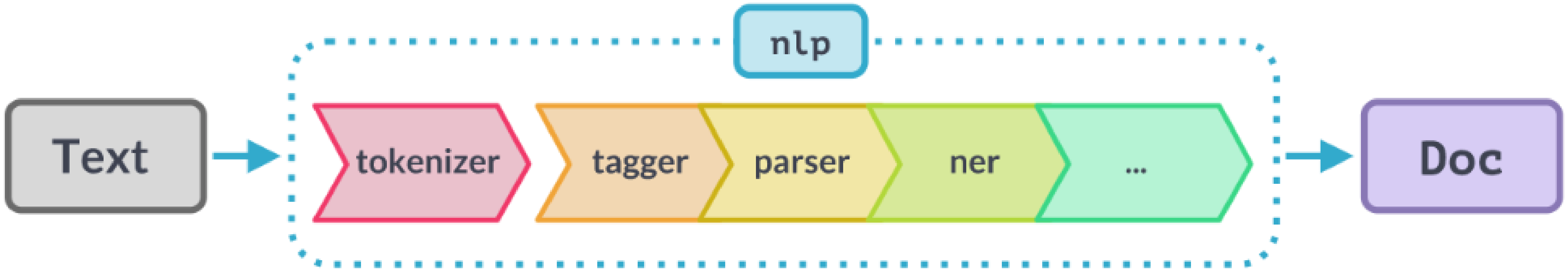
In this analysis, review text is the independent variable, and verified status is the dependent variable.

Our initial exploratory analysis shows that roughly 73% of the reviews are verified reviews, and the rest 27% are labeled as unverified reviews. About 80% of reviewers only provided 1 review and the rest of the customers provided on average 2.8 reviews.

Implementation Roadmap

1. Data collection and cleaning
2. Text Preprocessing
3. Modeling
4. Make predictions
5. Rank reviews on the likelihood to be false
6. Set threshold
7. Potentially censor or flag most suspicious ones

Use Spacy to Vectorize Documents

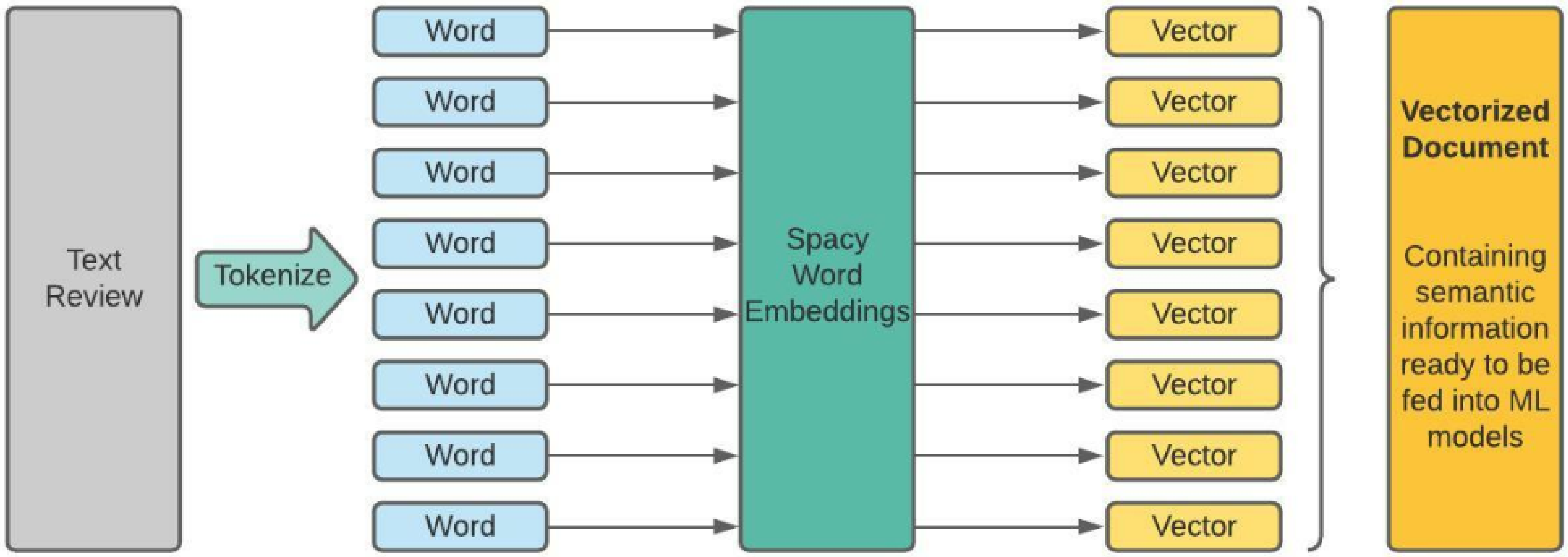


Spacy performs preprocessing and tokenization for individual words.

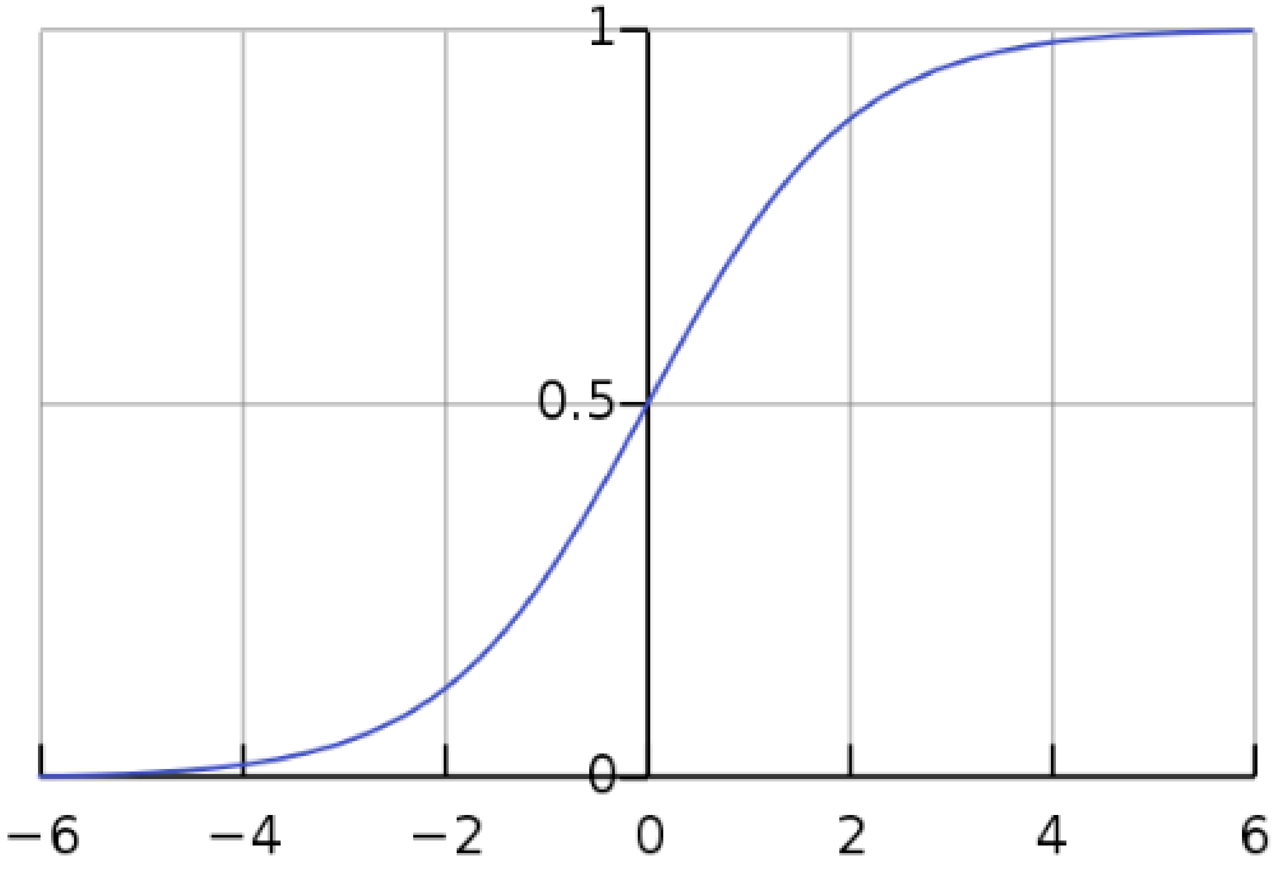
It then vectorizes documents(individual reviews) into [1 x 300] vectors.

Documents are thus represented in a digital format, capturing information for machines to ‘understand’.

Use Spacy to Vectorize Documents



LOGISTICS REGRESSION



A statistical model to predict the likelihood of categorical dependent variables, in our case, verified unverified reviews.

```
from sklearn.linear_model import LogisticRegression

lr = LogisticRegression(max_iter = 1000)
lr.fit(X_train_array, y_train)
```

LOGISTICS REGRESSION

```
y_pred = lr.predict(X_train_array)

np.mean(y_pred == y_train)
```

```
0.7791466666666667
```

```
cv_results
```

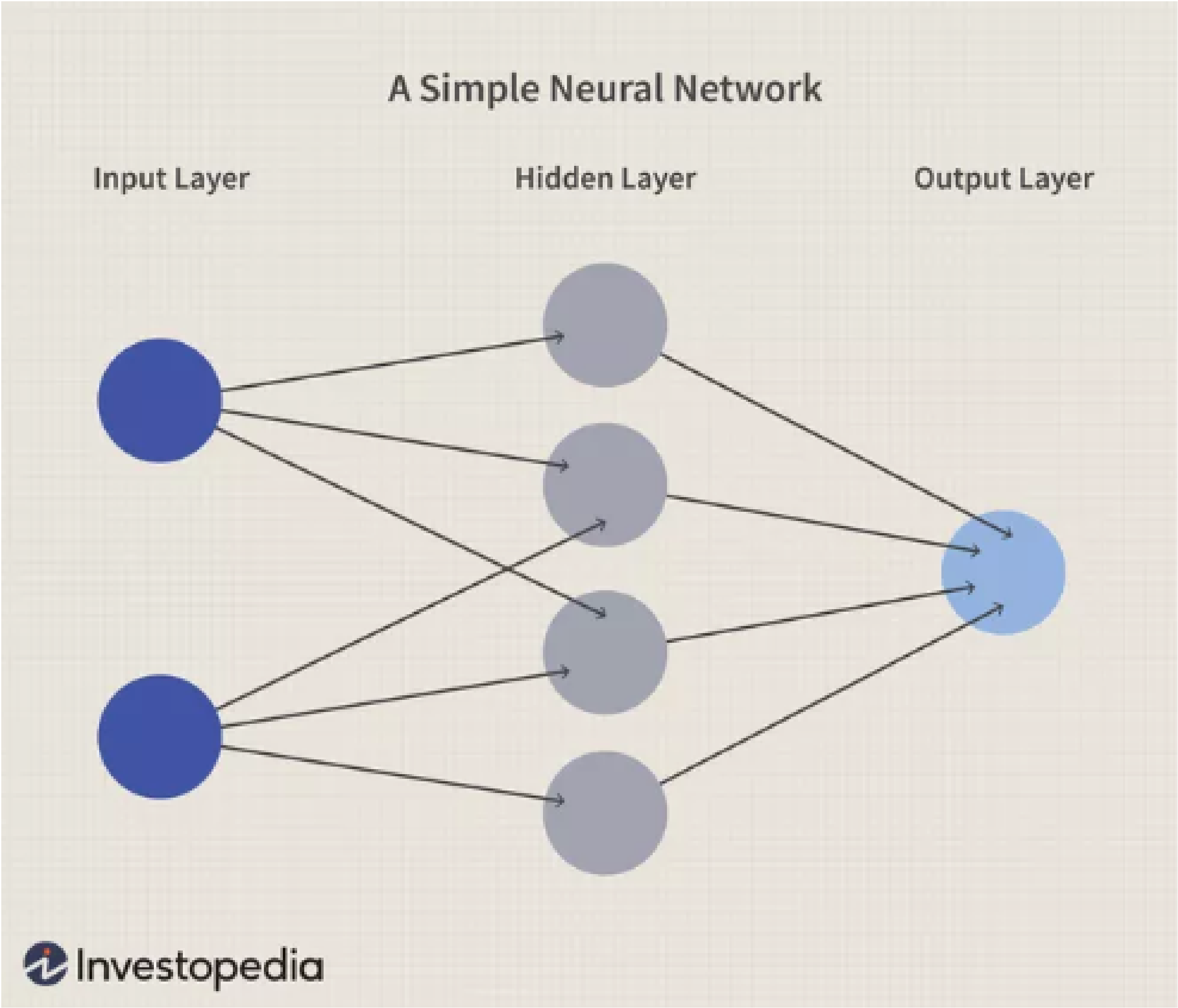
```
{'fit_time': array([13.67145824, 11.56526184, 13.90373492, 13.67203355, 13.13864994,
        13.1452899 , 11.66384983, 12.11314011, 12.64546061,  9.1454432 ]),
 'score_time': array([0.00897551, 0.00797796, 0.00797772, 0.00797844, 0.01595926,
        0.01695514, 0.00698233, 0.0079782 , 0.00797772, 0.00797915]),
 'test_score': array([0.7751, 0.78  , 0.7812, 0.7784, 0.777 , 0.779 , 0.7779, 0.7703,
        0.7755, 0.7802])}
```

Logistic Regression yielded 78% accuracy on the testing set.

10 fold cross-validation yielded similar results between 77.5% to 78.12%

Neural Net

- Mimics the mechanism of the brain.
- Works with high dimensionality.
- Describes non-linear relationships.
- Highly customizable.



Neural Net Evaluation

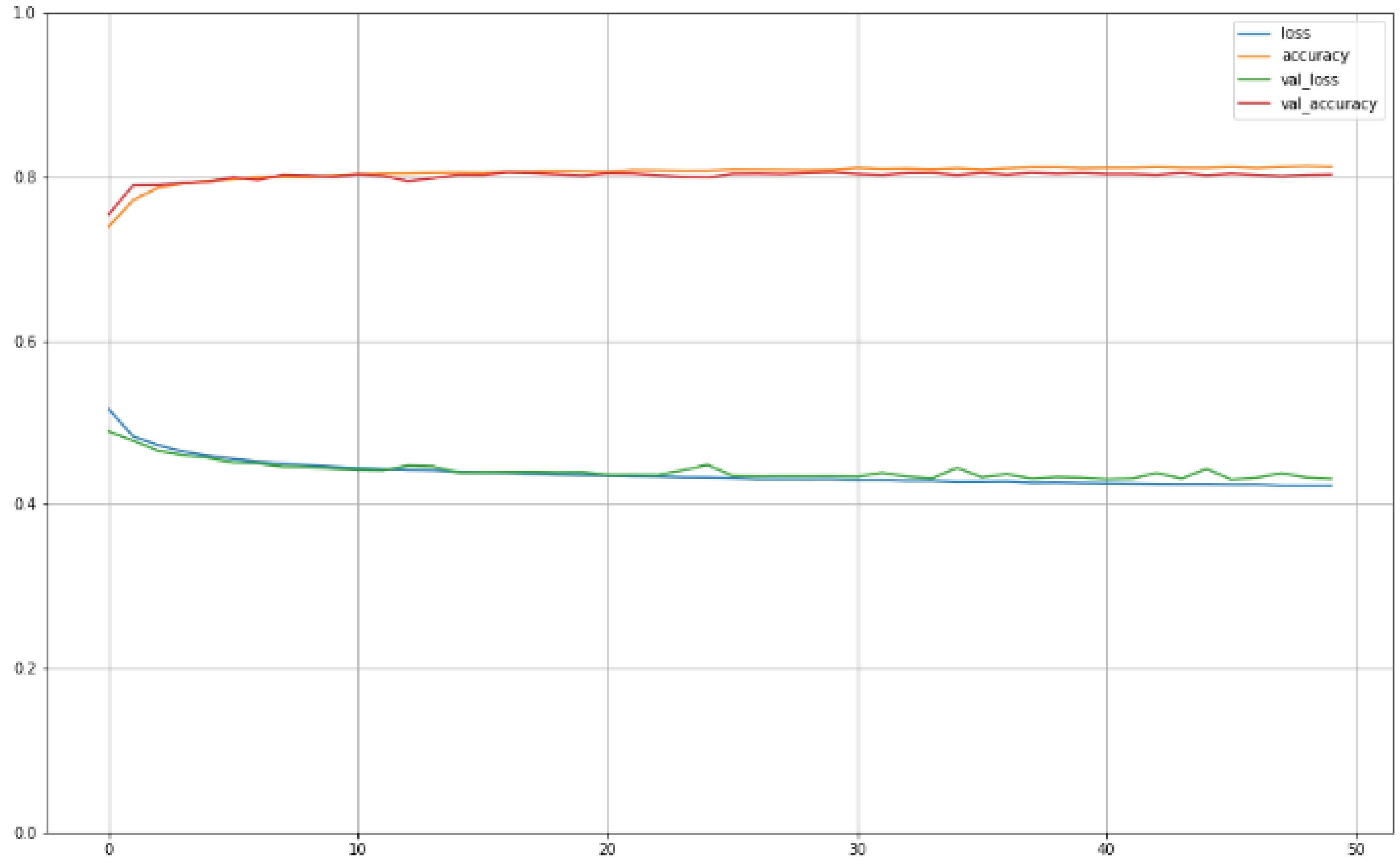
```
model = keras.models.Sequential()
model.add(keras.layers.Dense(300, input_dim = 300, activation = "relu"))
model.add(Dropout(0.2))
model.add(keras.layers.Dense(100, activation = "relu"))
# Add 1 more layer here
model.add(keras.layers.Dense(1, activation = "sigmoid"))

model.compile(loss = "binary_crossentropy",
             optimizer = "sgd",
            metrics = ["accuracy"])

es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience = 10)
```

- While simple, the neural net takes significantly longer than the logistic regression to run.
- Slight improvement in result.

Neural Net Evaluation



- Accuracy Plateaued around 80%
- No sign of overfitting on the validation set

Neural Net Evaluation

```
model.evaluate(X_test_array, y_test)
```

```
782/782 [==============================] - 3s 4ms/step - loss:
0.4334 - accuracy: 0.8096

[0.43342652916908264, 0.8096399903297424]
```

Far from Ideal, but 3% significant improvement over Logistic Regression.

	verified	prediction	probability
30613	1	1	0.976659
55296	0	0	0.141027
16327	1	1	0.647360
38471	1	1	0.961357
1593	0	0	0.236146

```
data.iloc[[80613]]['reviewText']
```

```
80613    ok
Name: reviewText, dtype: object
```

Neural Net Evaluation

unverified_percent	probability	all_percent
0.01	0.104847	0.00272
0.03	0.122591	0.00896
0.05	0.138008	0.01556
0.10	0.163524	0.03136
0.20	0.214259	0.06436
0.30	0.269709	0.10040
0.40	0.338754	0.14032
0.50	0.424574	0.18888

The current model picks up 10 % of all unverified reviews by labeling 3% from the most suspicious of all reviews.

The model becomes marginally less effective

But picks up 50% of all unverified reviews by labeling the most suspicious 18.8% of all reviews.

Business Impact

The estimated range of fake reviews on e-comm platforms is from 4% to 39%. According to the World Economic Forum, it translates into $152 billion of global purchases every year. Having a false-review detection that labels fake reviews in time will help customers make more informed purchasing decisions, decreasing the chances and dollar amounts of fraud.

For vendors/business owners, faking reviews not only violates the trust that consumers place on reviews but also might incur legal costs and hefty fines in thousands, even in millions. A false-review detection mechanism can decrease incentives for fake reviews.

For platforms like Amazon, the presence of fake reviews causes a loss of confidence from customers. Eliminating fake reviews will help maintain a strong trust among existing and future customers. From a long-term perspective, it increases brand loyalty and maintains a stable customer base.

Potential Return on Investment

Unverified reviews may result from emerging merchants trying to gain an edge over established vendors by giving out free products or simply paying for them.

Imposters hurt all parties involved and create negative externalities and inefficiencies in the market.

Specific enumeration can be difficult to quantify but the following assumptions and estimation can shed some light.

Potential Return on Investment

- Amazon ships out 1.6 million packages per day.
- Assuming $4.50 USPS standard shipping costs one way.
- Assuming 10% of all returns are caused by false reviews.
- Assuming Linear relationship: each 1% reduction in the false review means a 0.01% reduction in returns.

- Accurately removing 50% of false reviews saves costs in shipping alone:

0.01% * 50 * 1.6 million * $4.5 * 2(round trip) = $72,200 per day.

With a mildly accurate model, one removes far less than 50% of all reviews to remove 50% of false reviews; e.g. just the most suspicious 18.8%.

Future Steps

About Amazon Verified Purchase Reviews

An "Amazon Verified Purchase" review means we've verified that the person writing the review purchased the product at Amazon and didn't receive the product at a deep discount.

Reviews that are not marked "Amazon Verified Purchase" are valuable as well, but we either can't confirm that the product was purchased at Amazon or the customer did not pay a price available to most Amazon shoppers.

Long reviews tend to be predicted as unverified, and short reviews are verified.

Possible explanations:

- Overly extensive elaboration may have distorted vectors to the point they lose information.
- Most people leave short reviews.
- Build models taking into account the length of the document, count of special characters, etc.

Citations

1. https://www.nbshare.io/notebook/595607887/Understanding-Word-Embeddings-Using-Spacy-Python/
2. https://en.wikipedia.org/wiki/Logistic_regression
3. https://www.investopedia.com/terms/n/neuralnetwork.asp
4. https://www.qualtrics.com/blog/online-review-stats/
5. https://www.aboutamazon.com/news/how-amazon-works/creating-a-trustworthy-reviews-experience?tag=cnet-buy-button-20&ascsubtag=d9a2d269-8295-4125-8ea3-ce997e30caab%7C%7Cdtp
6. https://www.reviewtrackers.com/blog/fake-reviews/
7. https://www.weforum.org/agenda/2021/08/fake-online-reviews-are-a-152-billion-problem-heres-how-to-silence-them/