Capstone 3

Mark Perez

Overview

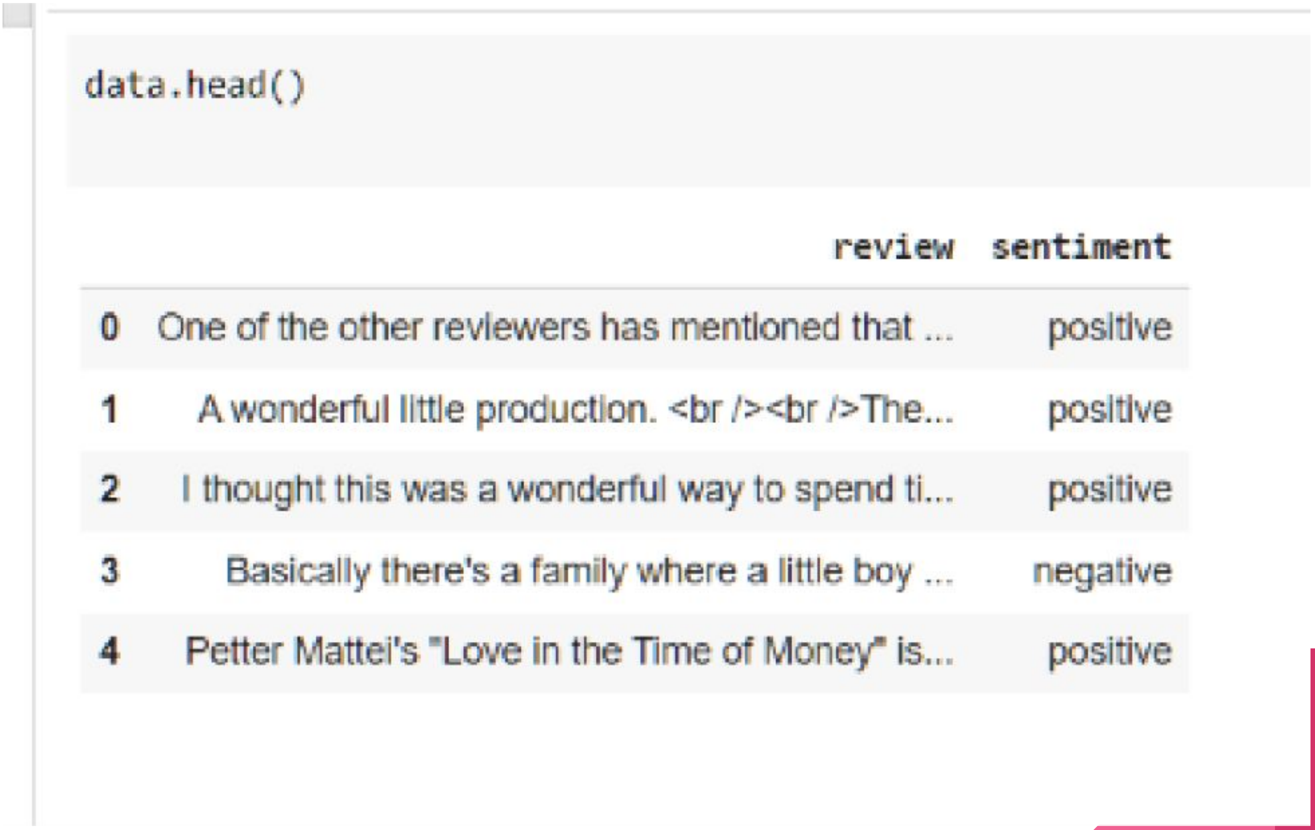
- IMDB Dataset
- 50K reviews
- Use NLP for prediction

kaggle

Why is it important?

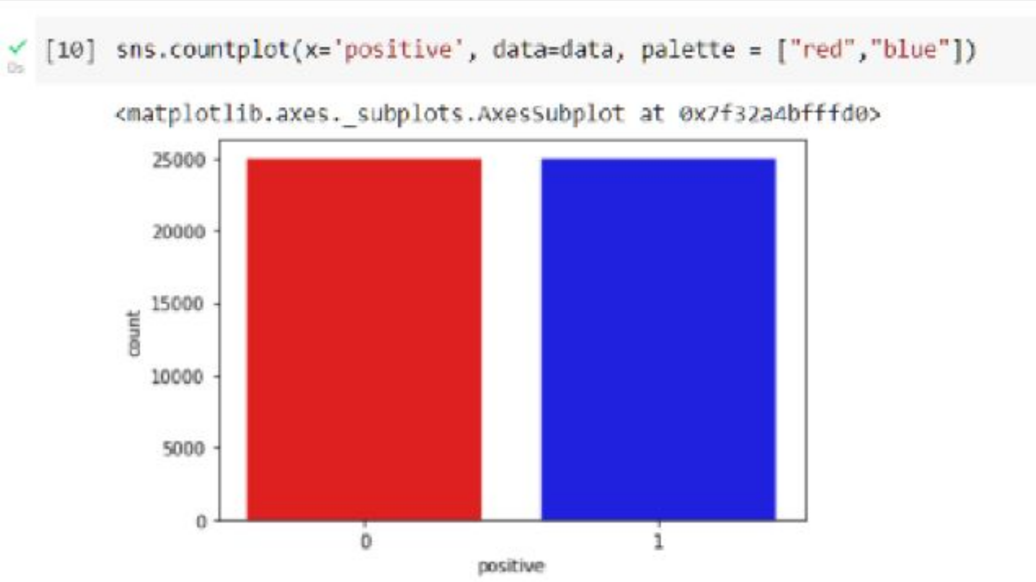
- Sentiment analysis is an important technique for natural language processing
- Models focus on polarity, emotions, or intentions
- Helps businesses make decisions

EDA



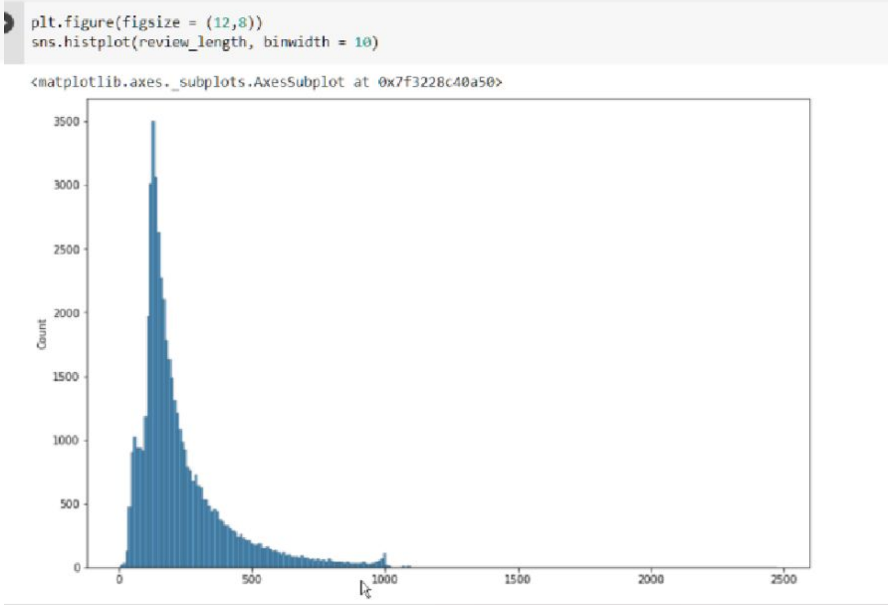
```
data.head()
```

	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. <br /><br />The...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive



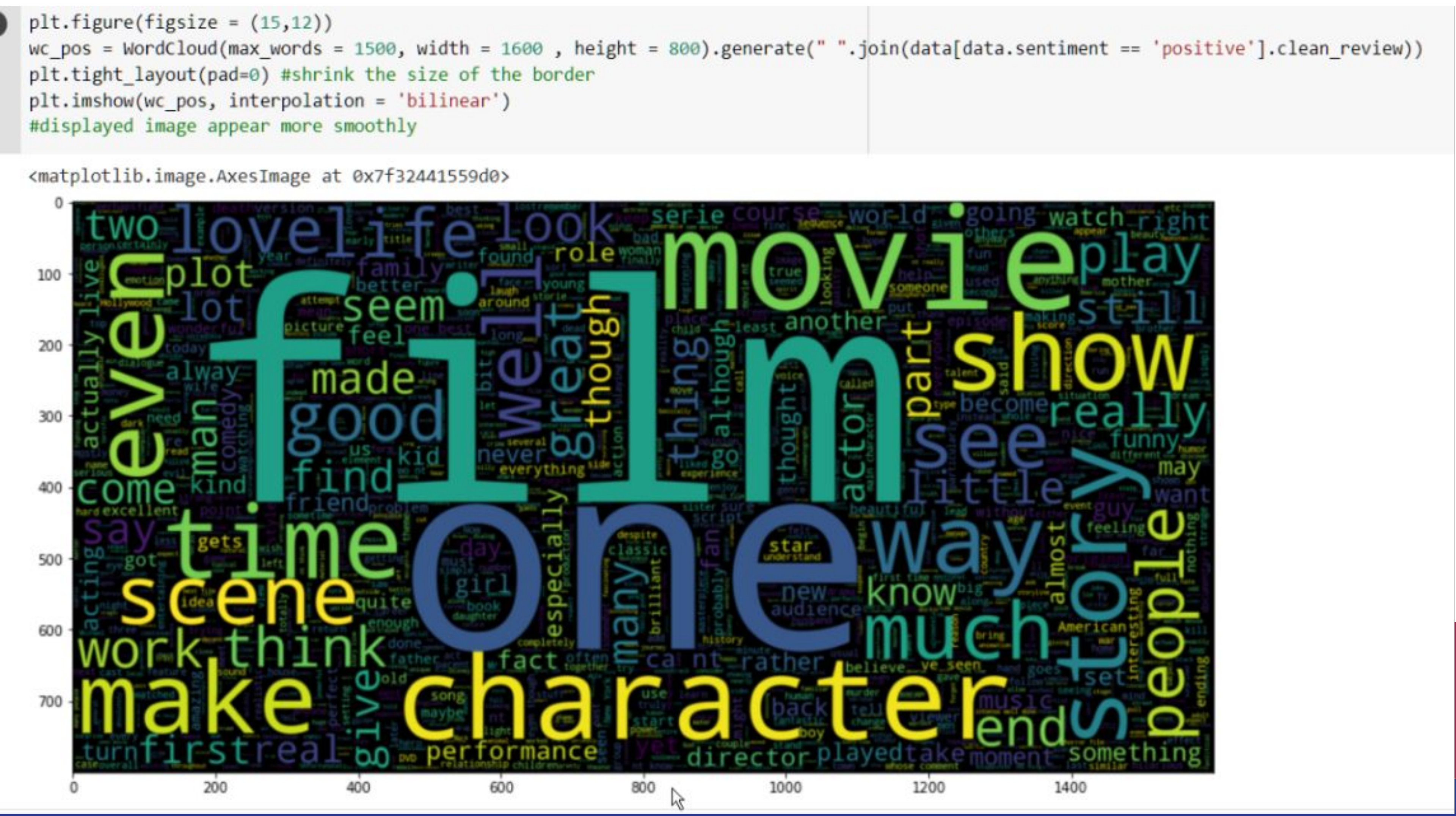
```
[10] sns.countplot(x='positive', data=data, palette = ["red","blue"])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f32a4bfffd0>
```



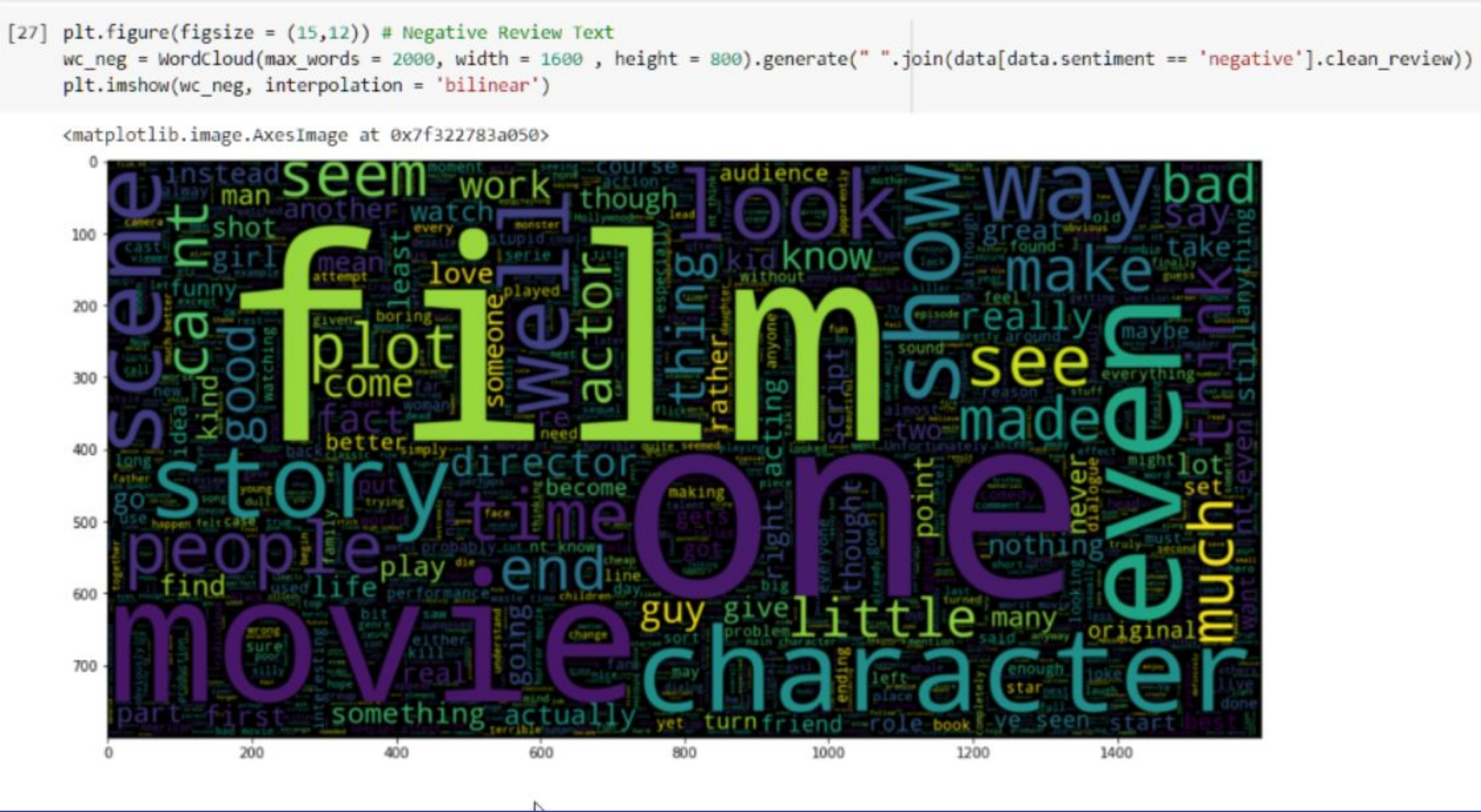
```
plt.figure(figsize = (12,8))
sns.histplot(review_length, binwidth = 10)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3228c40a50>
```



```
plt.figure(figsize = (15,12))
wc_pos = WordCloud(max_words = 1500, width = 1600 , height = 800).generate(" ".join(data[data.sentiment == 'positive'].clean_review))
plt.tight_layout(pad=0) #shrink the size of the border
plt.imshow(wc_pos, interpolation = 'bilinear')
#displayed image appear more smoothly
```

```
<matplotlib.image.AxesImage at 0x7f32441559d0>
```



```
[27] plt.figure(figsize = (15,12)) # Negative Review Text
     wc_neg = WordCloud(max_words = 2000, width = 1600 , height = 800).generate(" ".join(data[data.sentiment == 'negative'].clean_review))
     plt.imshow(wc_neg, interpolation = 'bilinear')
```

```
<matplotlib.image.AxesImage at 0x7f322783a050>
```

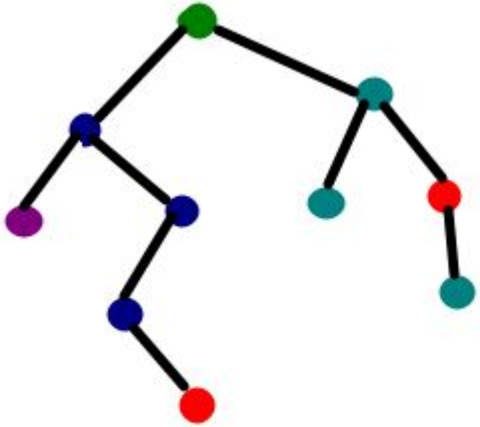
Modeling

- Two models for Binary Classification:

  Decision Tree Classifier & BERT

Decision Tree Classifier

- Simple to prepare and understand

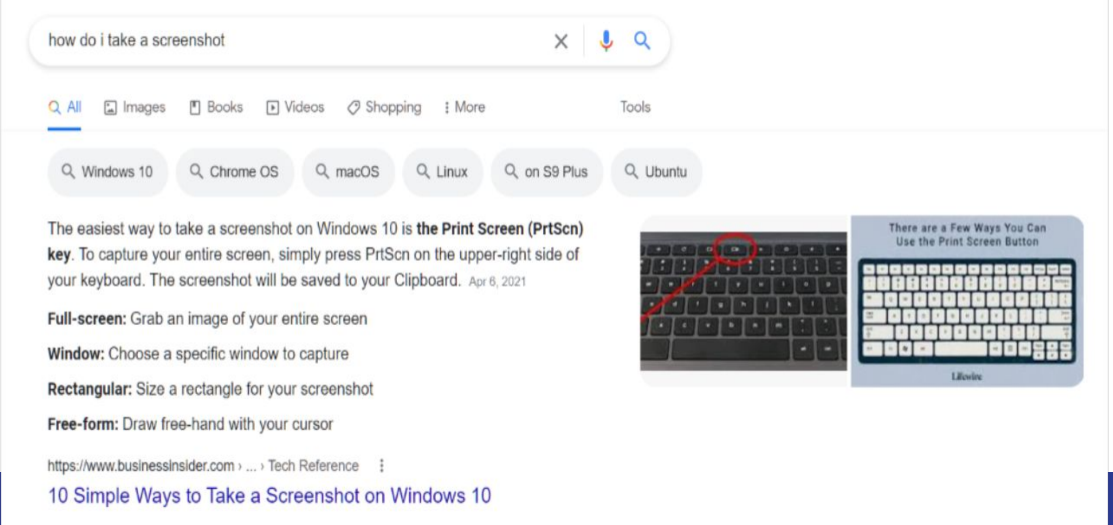


	Precision	Recall	F1-score	Accuracy
0	.82	.6	.69	
1	.47	.73	.57	
				.64

What is BERT?

- Bidirectional Encoder Representations from Transformers
- Each word is contextualized based on the other words in the sentence.
- Different versions of BERT

Text Summarization



how do i take a screenshot

The easiest way to take a screenshot on Windows 10 is **the Print Screen (PrtScn) key**. To capture your entire screen, simply press PrtScn on the upper-right side of your keyboard. The screenshot will be saved to your Clipboard. Apr 6, 2021

Full-screen: Grab an image of your entire screen

Window: Choose a specific window to capture

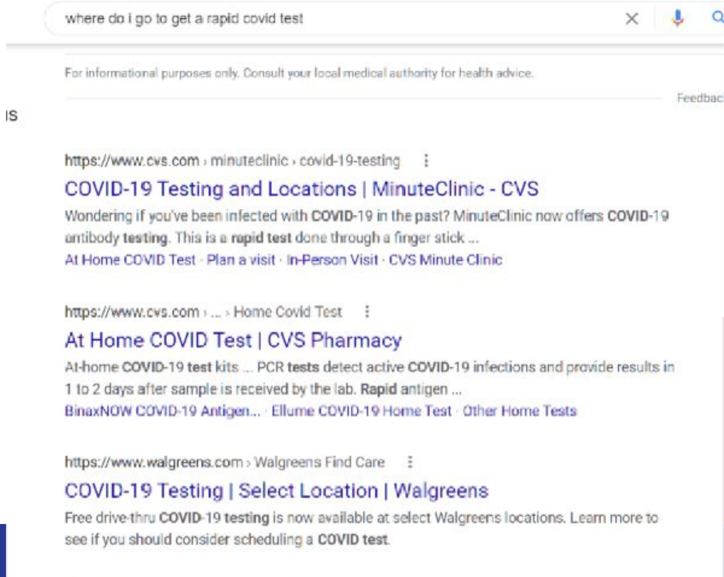
Rectangular: Size a rectangle for your screenshot

Free-form: Draw free-hand with your cursor

https://www.businessinsider.com › ... › Tech Reference

10 Simple Ways to Take a Screenshot on Windows 10

Text Encoding Similarity Retrieval



where do i go to get a rapid covid test

For informational purposes only. Consult your local medical authority for health advice.

https://www.cvs.com › minuteclinic › covid-19-testing

COVID-19 Testing and Locations | MinuteClinic - CVS

Wondering if you've been infected with **COVID**-19 in the past? MinuteClinic now offers **COVID**-19 antibody **testing**. This is a **rapid test** done through a finger stick ...

At Home COVID Test · Plan a visit · In-Person Visit · CVS Minute Clinic

https://www.cvs.com › ... › Home Covid Test

At Home COVID Test | CVS Pharmacy

At-home **COVID**-19 **test** kits ... PCR **tests** detect active **COVID**-19 infections and provide results in 1 to 2 days after sample is received by the lab. **Rapid** antigen ...

BinaxNOW COVID-19 Antigen... · Ellume COVID-19 Home Test · Other Home Tests

https://www.walgreens.com › Walgreens Find Care

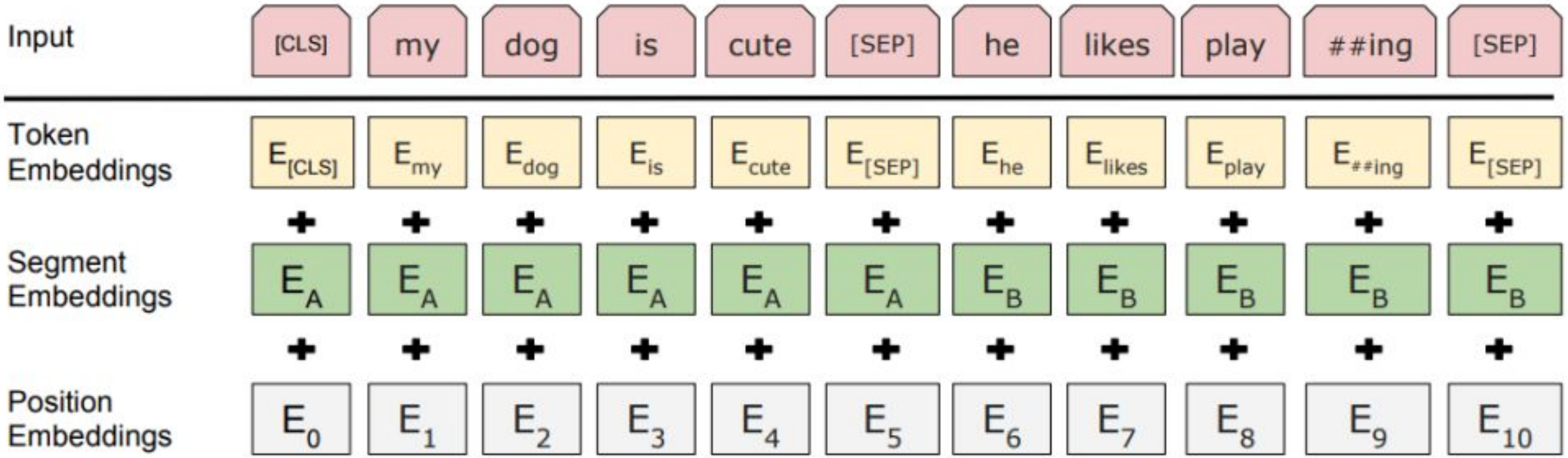
COVID-19 Testing | Select Location | Walgreens

Free drive-thru **COVID**-19 **testing** is now available at select Walgreens locations. Learn more to see if you should consider scheduling a **COVID test**.

How BERT works

- BERT relies on a Transformer
- The input for the encoder are three embedings: Token, Segment, and Positional
- BERT’s language modeling task (MLM) masks 15% of words in the input and asks the model to predict the missing word, as well as predict whether a following sentence (after a SEP token) is random or not.

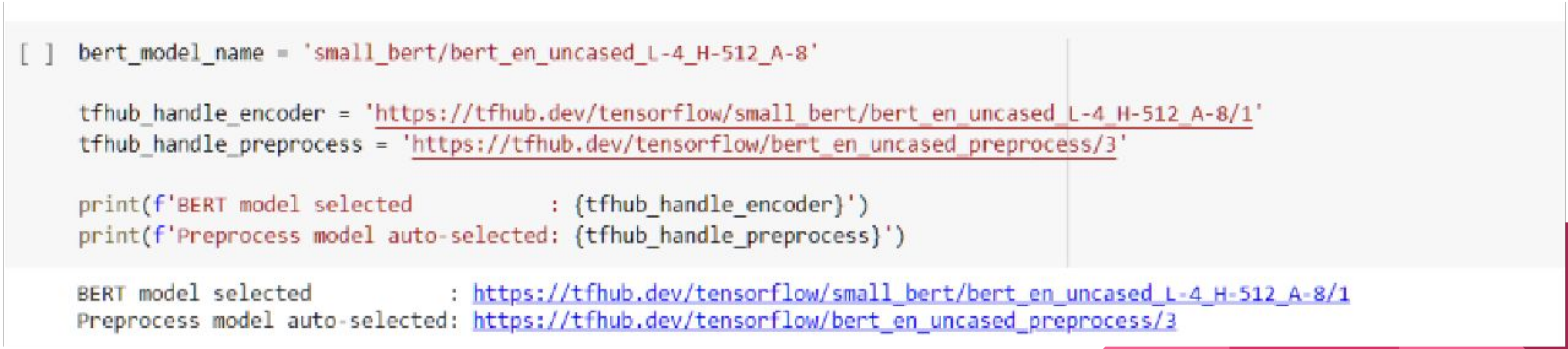
Visual representation of BERT tokens



Input	[CLS]	my	dog	is	cute	[SEP]	he	likes	play	##ing	[SEP]
Token Embeddings	$E_{[CLS]}$	E_{my}	E_{dog}	E_{is}	E_{cute}	$E_{[SEP]}$	E_{he}	E_{likes}	E_{play}	$E_{\#\#ing}$	$E_{[SEP]}$
	+	+	+	+	+	+	+	+	+	+	+
Segment Embeddings	E_A	E_A	E_A	E_A	E_A	E_A	E_B	E_B	E_B	E_B	E_B
	+	+	+	+	+	+	+	+	+	+	+
Position Embeddings	E_0	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}

Beauty of Tensorflow

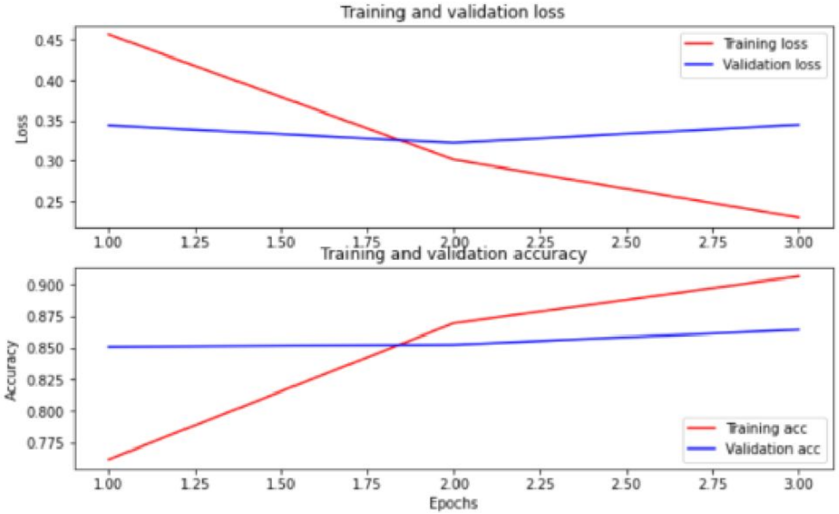
- TensorFlow Hub is a repository for trained machine learning models
- We load the preprocessing model (to prepare the text) and the model (small BERT) from TF Hub.

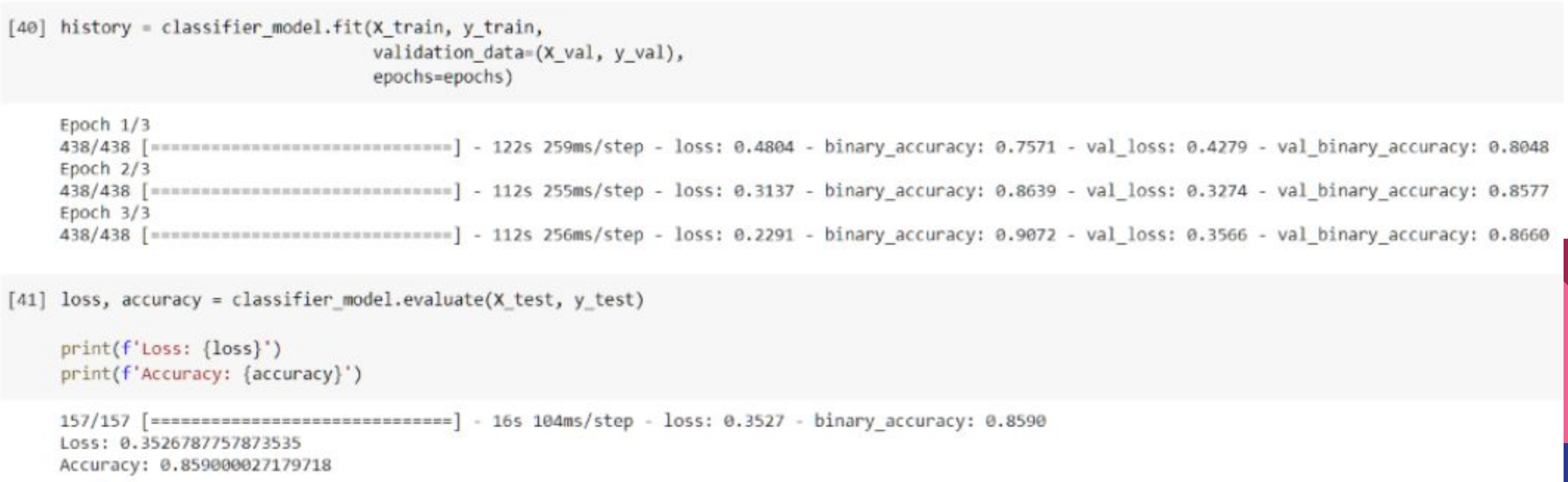


```
[ ]  bert_model_name = 'small_bert/bert_en_uncased_L-4_H-512_A-8'

     tfhub_handle_encoder = 'https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-4_H-512_A-8/1'
     tfhub_handle_preprocess = 'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3'

     print(f'BERT model selected           : {tfhub_handle_encoder}')
     print(f'Preprocess model auto-selected: {tfhub_handle_preprocess}')
```

```
     BERT model selected           : https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-4_H-512_A-8/1
     Preprocess model auto-selected: https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3
```

Results





```
[40] history = classifier_model.fit(X_train, y_train,
                                    validation_data=(X_val, y_val),
                                    epochs=epochs)
```

```
Epoch 1/3
438/438 [==============================] - 122s 259ms/step - loss: 0.4804 - binary_accuracy: 0.7571 - val_loss: 0.4279 - val_binary_accuracy: 0.8048
Epoch 2/3
438/438 [==============================] - 112s 255ms/step - loss: 0.3137 - binary_accuracy: 0.8639 - val_loss: 0.3274 - val_binary_accuracy: 0.8577
Epoch 3/3
438/438 [==============================] - 112s 256ms/step - loss: 0.2291 - binary_accuracy: 0.9072 - val_loss: 0.3566 - val_binary_accuracy: 0.8660
```

```
[41] loss, accuracy = classifier_model.evaluate(X_test, y_test)

     print(f'Loss: {loss}')
     print(f'Accuracy: {accuracy}')
```

```
157/157 [==============================] - 16s 104ms/step - loss: 0.3527 - binary_accuracy: 0.8590
Loss: 0.3526787757873535
Accuracy: 0.859000027179718
```

Conclusion

- BERT Model is great at Sentiment analysis
- Accuracy increased by about 22%
- Accuracy is not always the best metric to measure

Further Analysis and Constraints

- More data
- Compare other models (Random Forest and LSTM)
- Fine tuning