State-Of-The-Art Text Classification with ULMFiT & RoBERTa

Pooja Umathe

Haihong Ma

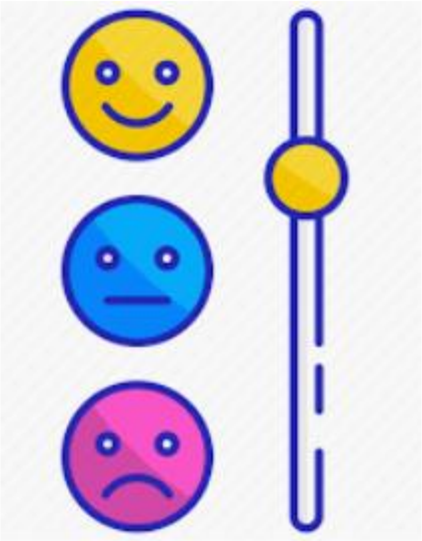
Summary

- Project/Business Requirement
- Data Selection
- Data Cleaning and Exploratory Data Analysis
- Classification
- ULMFiT
- RoBERTa
- Comparing ULMFiT and RoBERTa Results
- Prediction Demo

Project/Business Requirement

Project/Business Requirement

- Develop a Scale to Measure a Business using Social Media Data



Data Selection

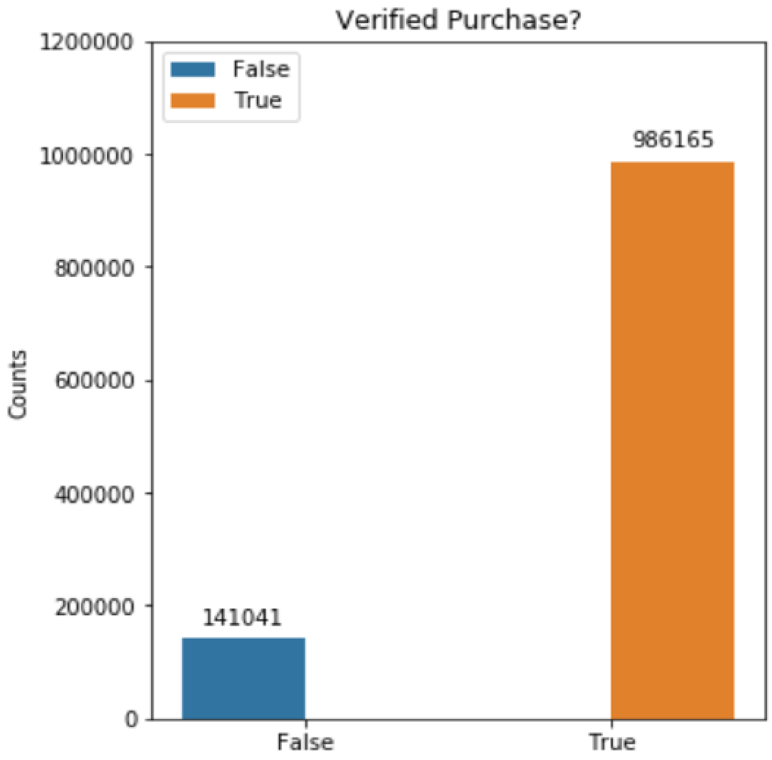
Data Selection

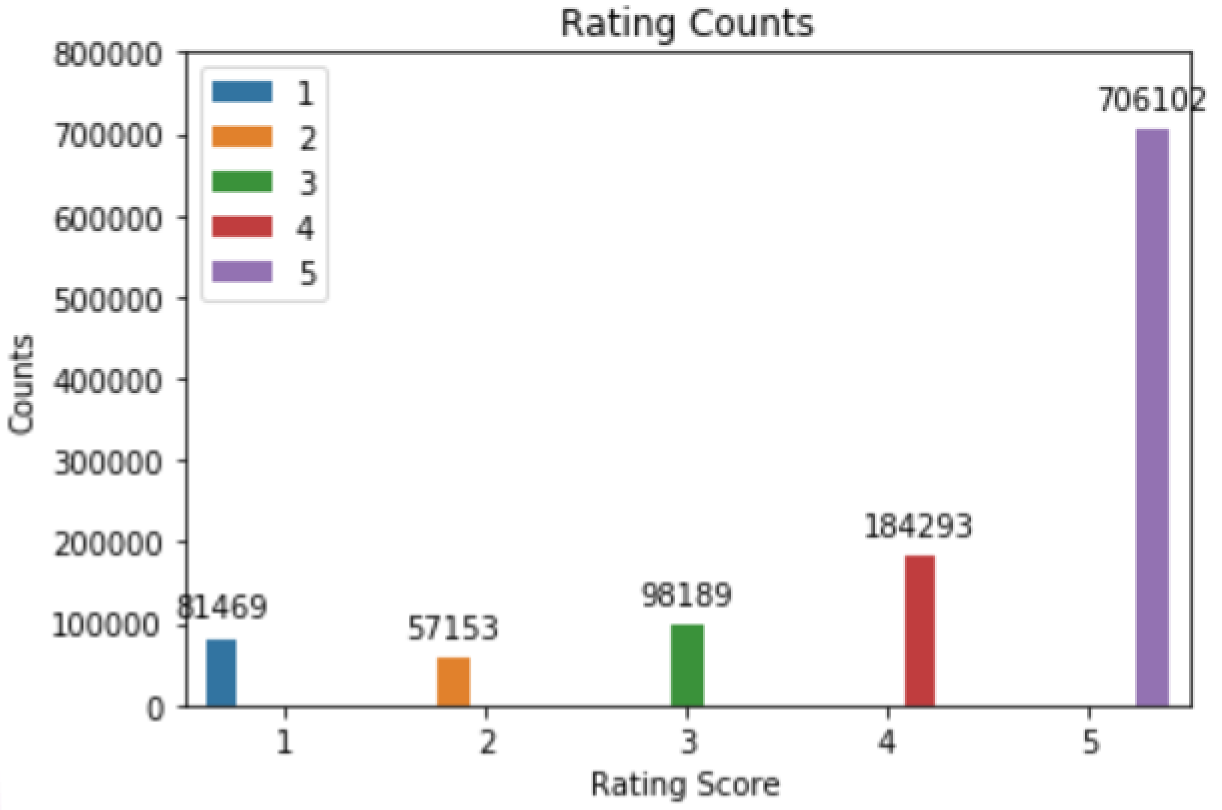
- ➢ amazon labelled data – Product Reviews for Electronic Shopping and Mail-Order Houses (NAICS code 454110)

- ➢ What's inside the data?
  - Data is organized by Product Category
  - Rating score (1- 5) along with product reviews.

Data Cleaning and Exploratory Data Analysis

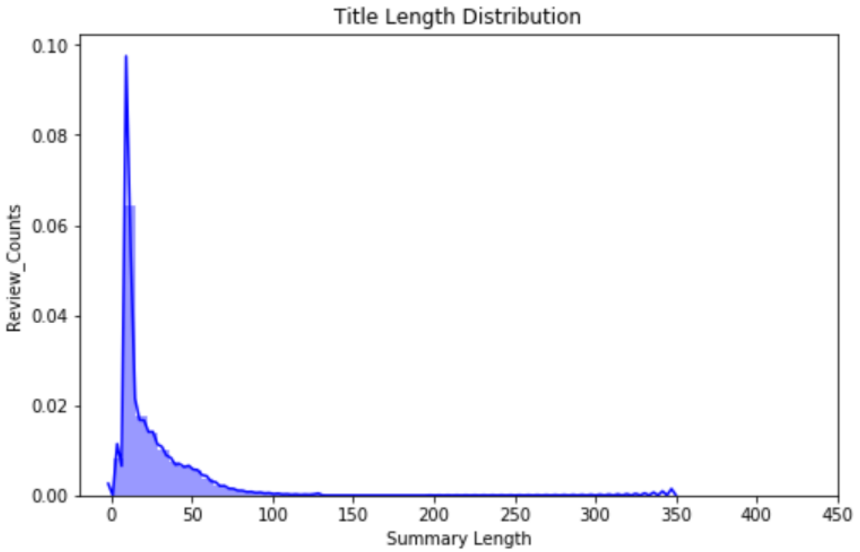
EDA:
Counts by rating

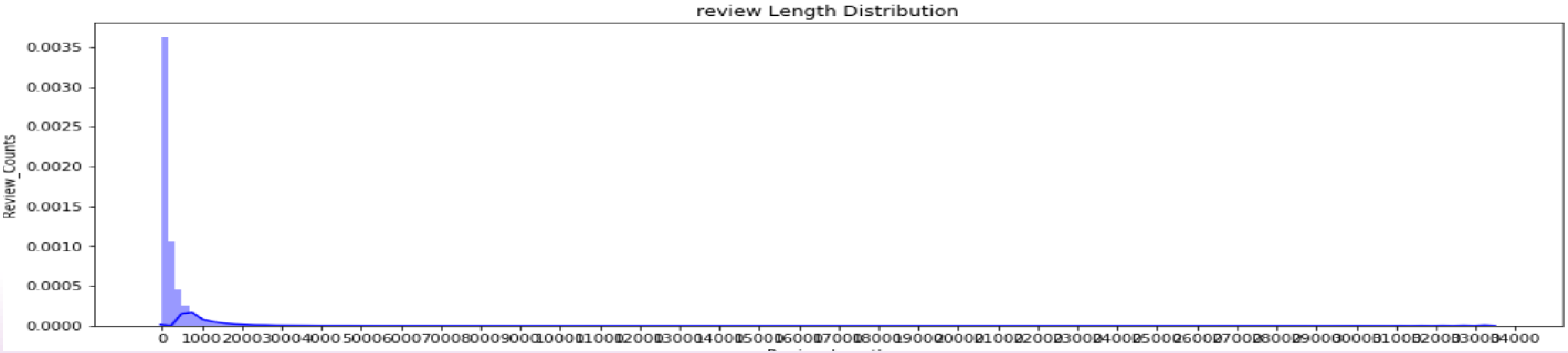




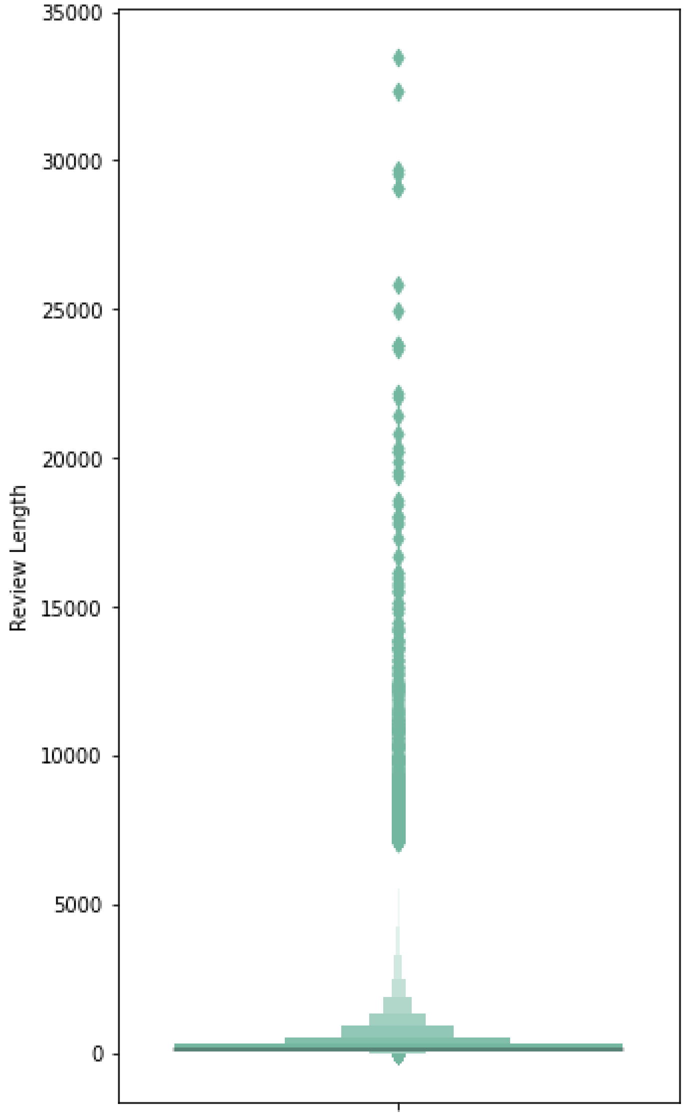
EDA:

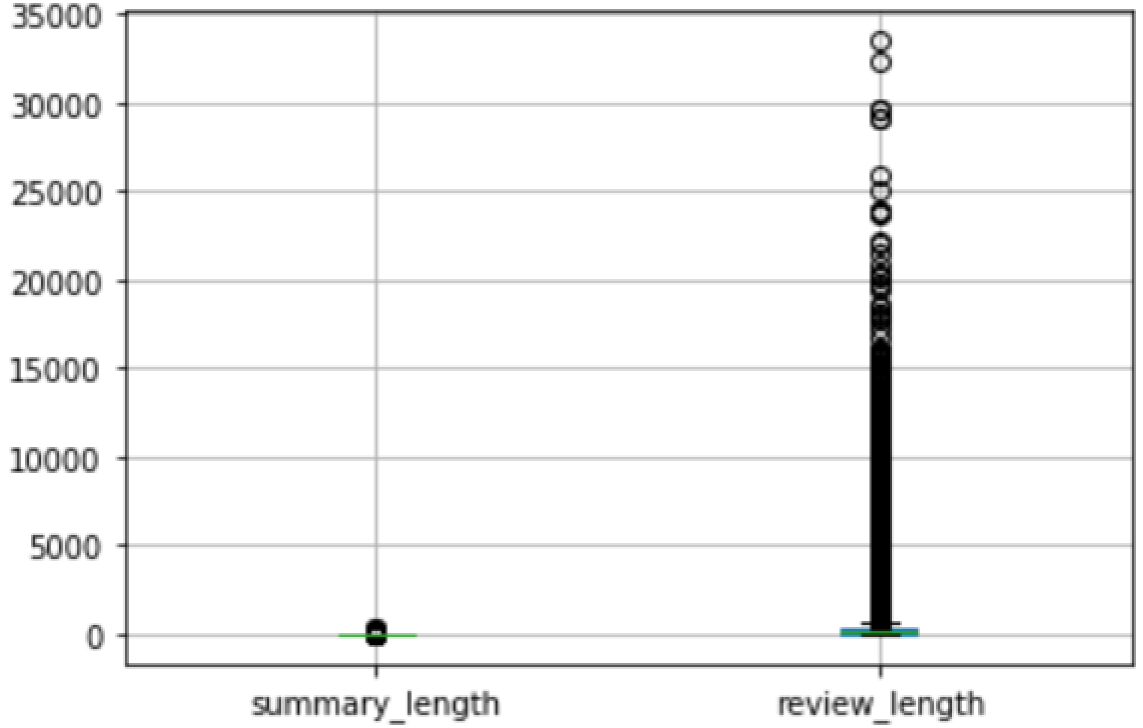
Title and Review Length Distribution



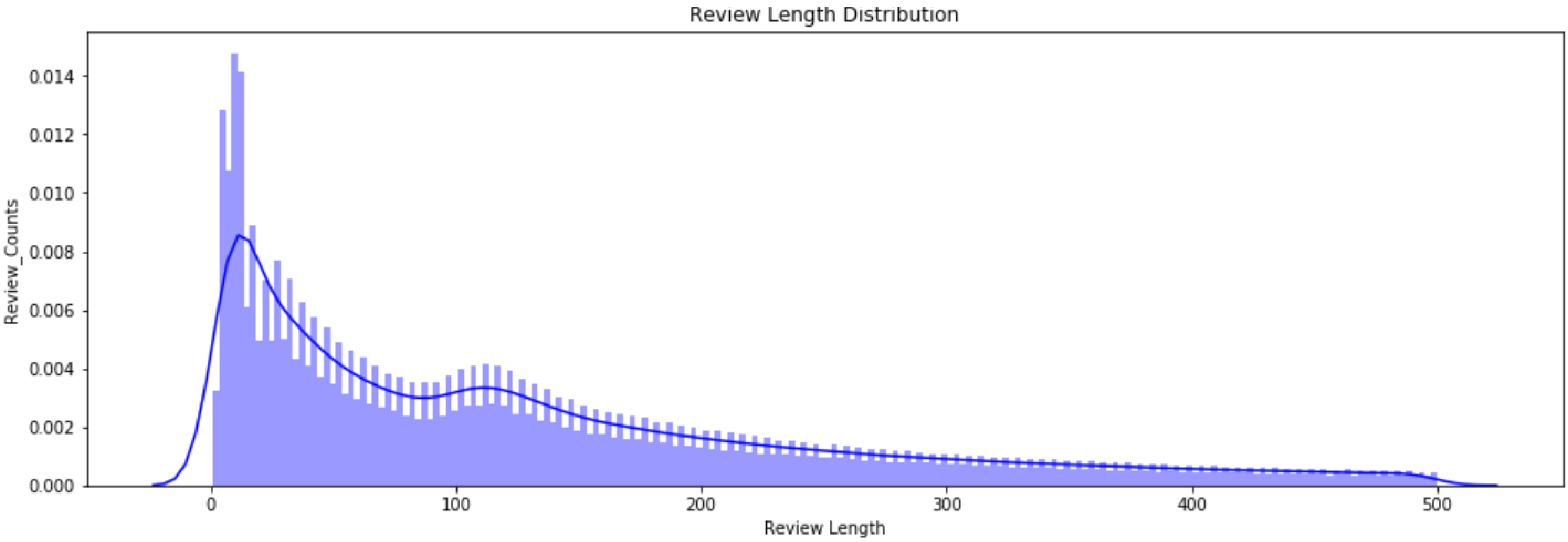


EDA:
Outliers Analysis

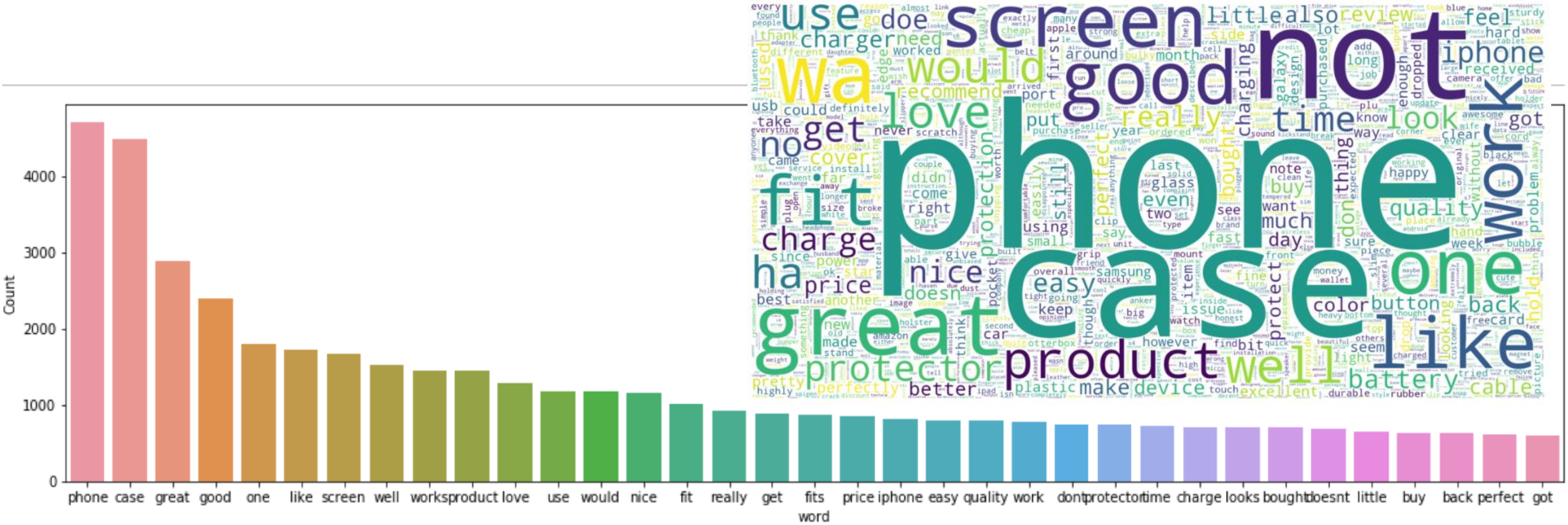




EDA:
Rolled Out the Outliers by Keeping Max Review Length 500



EDA:
Most Frequent Words

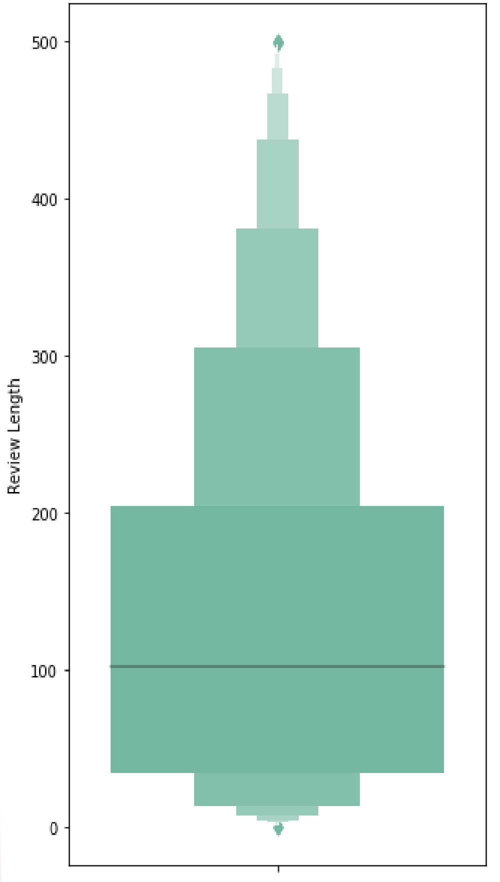


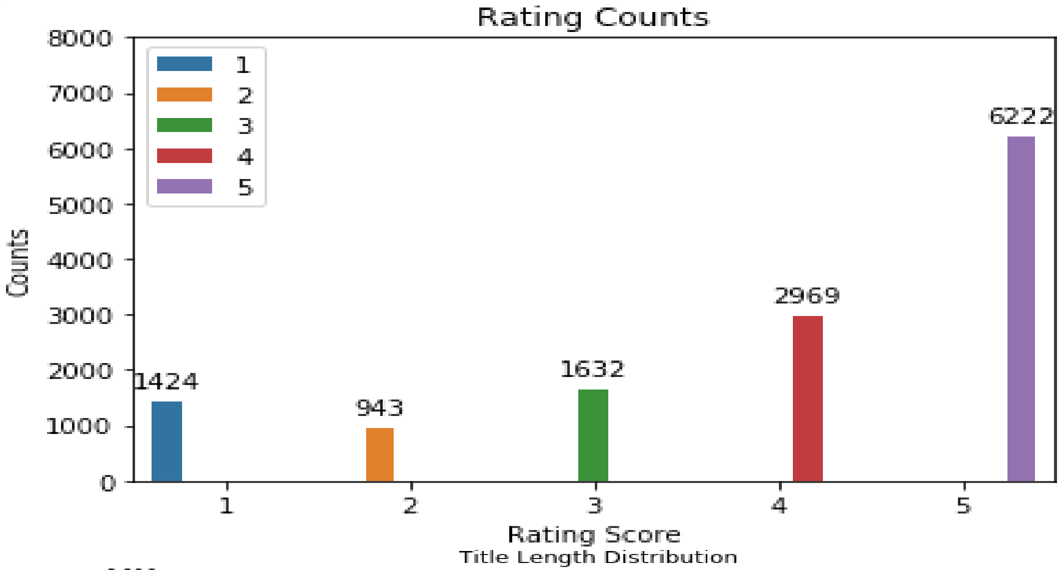
Data Sampling

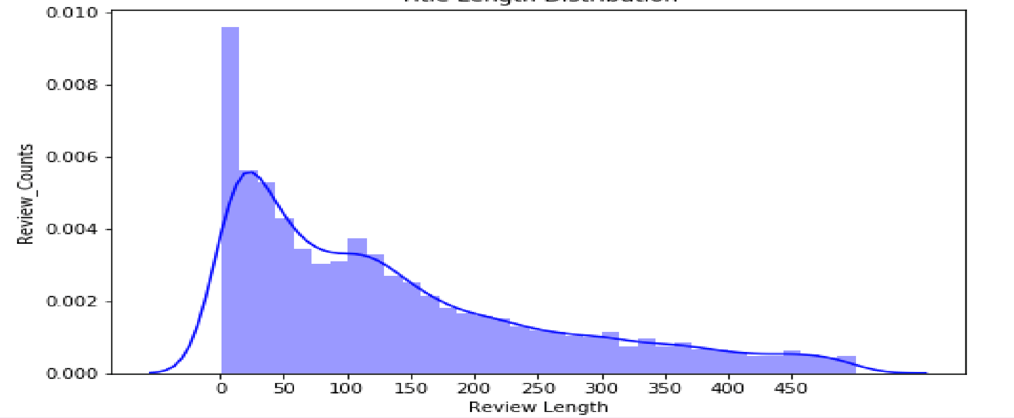
- Sampling by rating:
  - 1: 2%, 2: 2%, 3: 2%, 4: 2%, 5: 1%
  - Records: **13190**
- Final Dataset:

  1:10.8%, 2: 7.1%, 3: 12.4%,

  4: 22.5%, 5: 47.2%







Classification

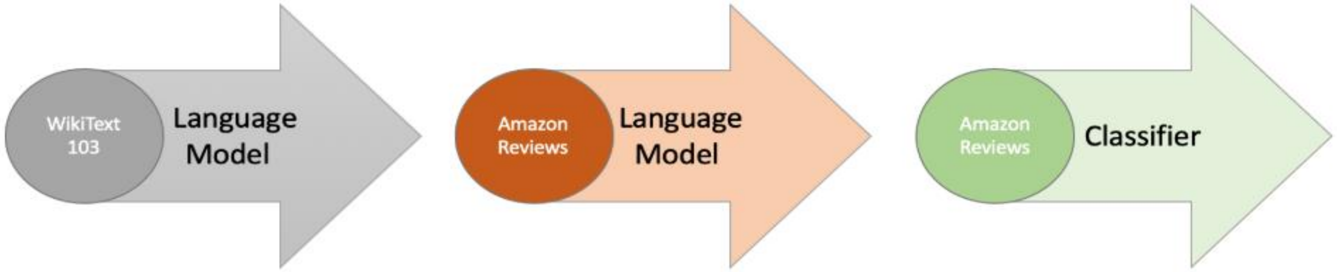
Classification

- Text Classification by fine tune modeling using RoBERTa and ULMFiT models by generating scores for each review.
- Classes:
  - <u>Binary classes</u>
    - **Negative** (ratings 1, 2 and 3) and **Positive** (ratings 4 and 5)
    - **Negative** (ratings 1 and 2) and **Positive** (ratings 3, 4 and 5)
    - **Negative** (ratings 1 and 2) and **Positive** (ratings 4 and 5), rating 3 is excluded
  - <u>Triple classes</u>
    - **Negative** (ratings 1 and 2), **Neutral** (rating 3) and **Positive** (ratings 4 and 5)
  - <u>Multi-classes</u>
    - Each Rating 1, 2, 3, 4, 5 as an individual class

ULMFiT (Universal Language Model Fine-Tuning for Text Classification)

Introduction of ULMFiT

- **U**niversal **L**anguage **M**odel **Fi**ne-**T**uning for Text Classification



- 1. General-domain Language Model pretraining
  - Wikitext-103, 28,595 preprocessed Wikipedia articles and 103 million words
- 2. Target Task Language Model Fine-tuning
  - To adapt to the target dataset which likely has a different distribution than the pretrained dataset
  - Discriminative fine-tuning and slanted triangular learning rates
- 3. Target Task Classifier Fine-tuning
  - Gradual unfreezing: start from the last layer which contains the least general knowledge

Prerequisites:

- fastai
- Works better with Linux
- Experimental with Windows, expected slowness or performance issues
- Pytorch v1 and Python 3.6+
- Refer to docs.fast.ai/install.html: Getting started – Installation & Installation Extras for details

Methodology

Wikitext 103

Pretrained

Amazon Product Review Unlabeled

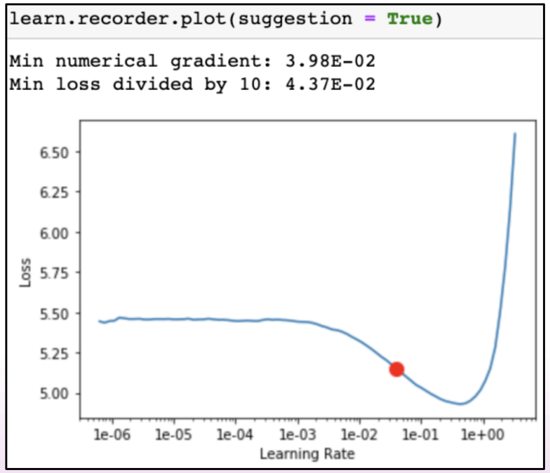
Datablock API
- Create a databunch
- Load the LM model
- Find a lr
- Fine-tune the LM

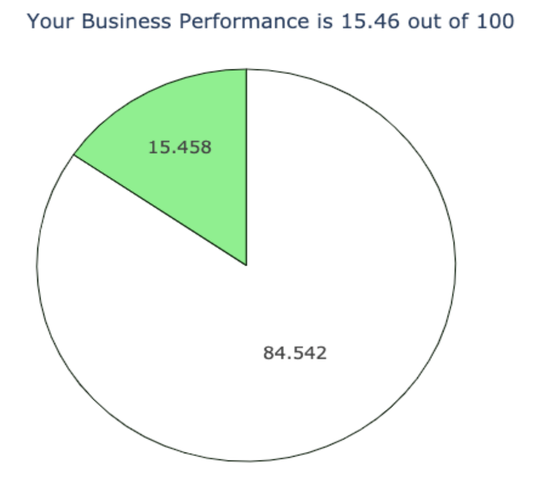
Amazon Product Reviews Labelled

Datablock API
- Create a databunch
- Load the Classifier model
- Freeze before last weights layers
- Find a lr
- Fine-tune the classifier

Prediction:

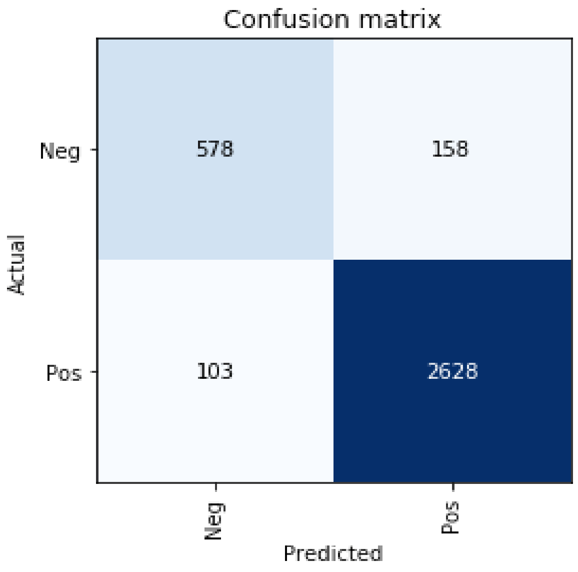
Scale of a product review being negative/positive

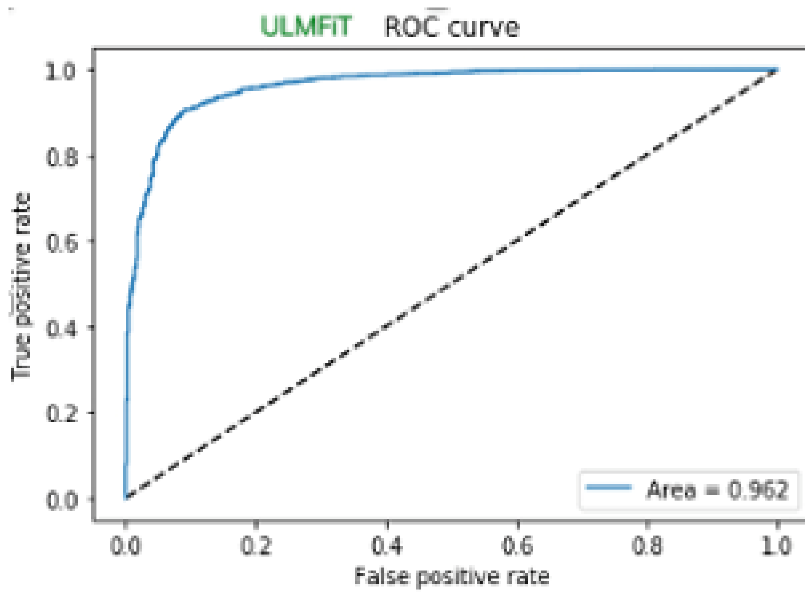




Results

Accuracy	92.47
Sensitivity	96.23
Specificity	78.53
Precision	94.33
F-1	95.27





RoBERTa (Robustly Optimized BERT Pretraining Approach)

Introduction of RoBERTa

RoBERTa is a Robustly Optimized BERT Pretraining Approach created and published by Facebook in July 2019.

This NLP system improves on Bidirectional Encoder Representations from Transformers(BERT) , the self-supervised method released by Google in 2018.

It is designed to help researchers develop ways for their AI systems to process language in a way that is not exclusive to a single task, genre, or dataset.

Why RoBERTa?

- It produces state-of-the-art results on the widely used NLP benchmark, General Language Understanding Evaluation (GLUE).

The modification over BERT include:

1. Training the model longer, with bigger batches
2. Removing the next sentence prediction objective
3. Training on longer sequences
4. Dynamically changing the masking pattern applied to the training data

Prerequisites

- Access to GPU
- Fastai and transformers libraries should be installed
- Access to the Google Colab
- Pytorch, Tensorflow and Cuda toolkit should be installed

Methodology

Setting up tokenizer

Creation and loading Roberta's Vocabulary

Setting up DataBunch

Creation of preprocessors for tokenizer and vocabulary

Loading the Amazon unlabeled data

Create a DataBunch class and Roberta textlist class

Building RoBERTa Model

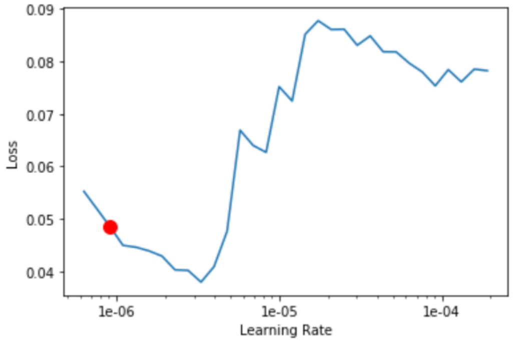
Define model architecture and initialize the model

Train the Model

Initialize Fastai learner and start training

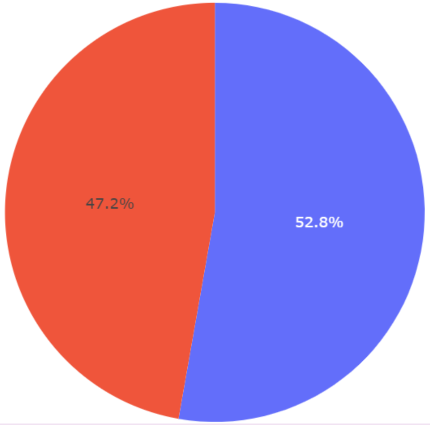
Predictions: Scale of a review being pos/neg

```
Min numerical gradient: 9.12E-07
Min loss divided by 10: 3.31E-07
```



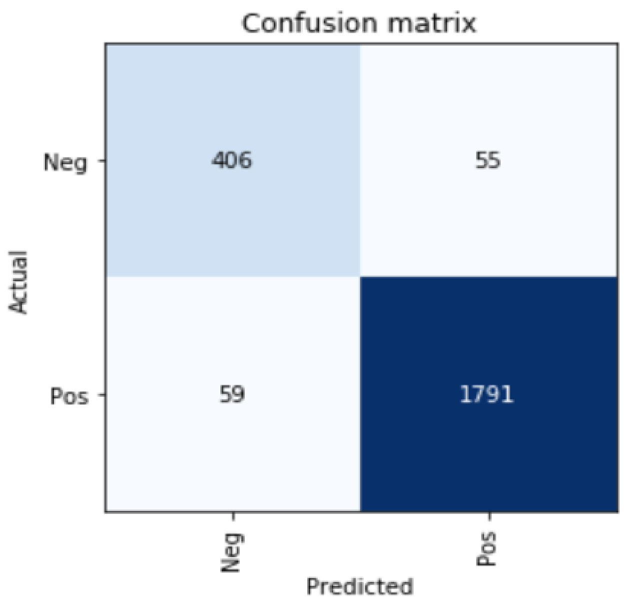

```
The review class is:  Neg , with a scale of  52.77  out of 100.
```

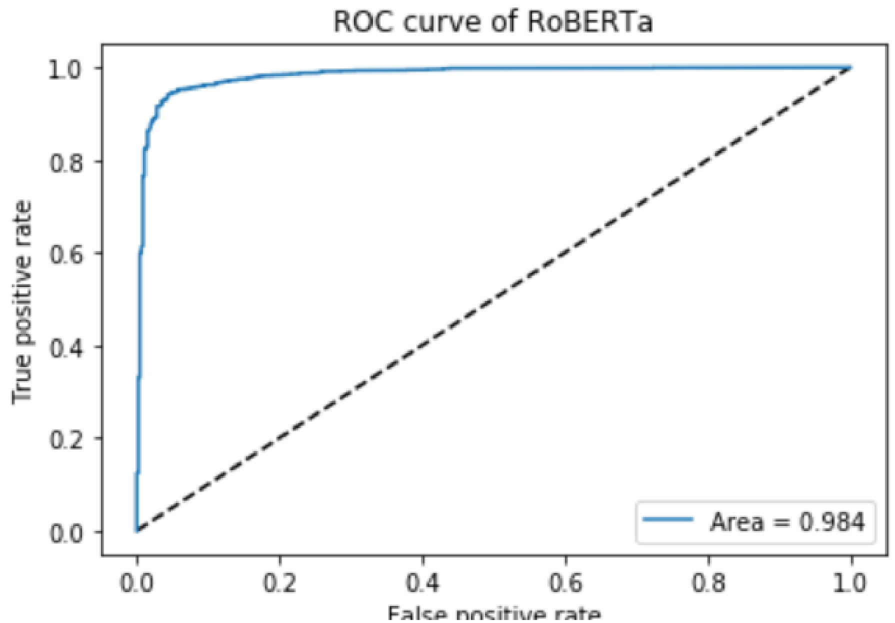
this phone case is cute, but it's not that sturdy...



Results

Accuracy	95.06
Sensitivity	96.81
Specificity	88.06
Precision	97.02
F-1	96.91

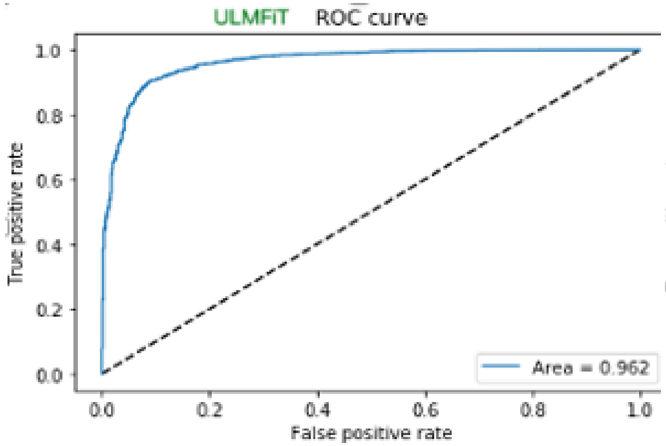


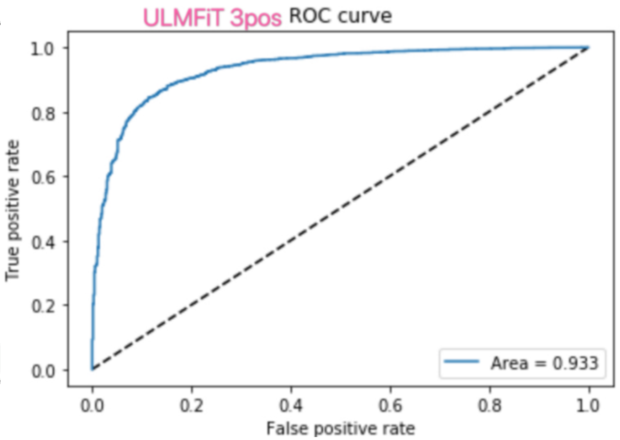


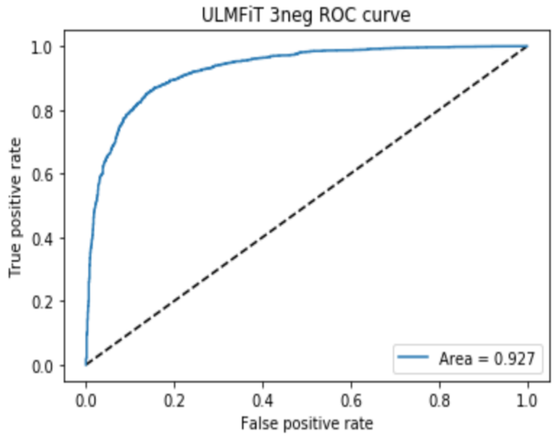
Comparing ULMFiT and RoBERTa Results

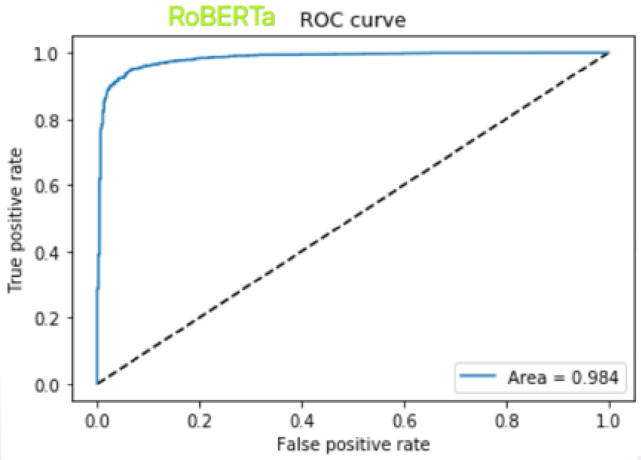
Results of RoBERTa and ULMFiT models:
Accuracy, ROC curve

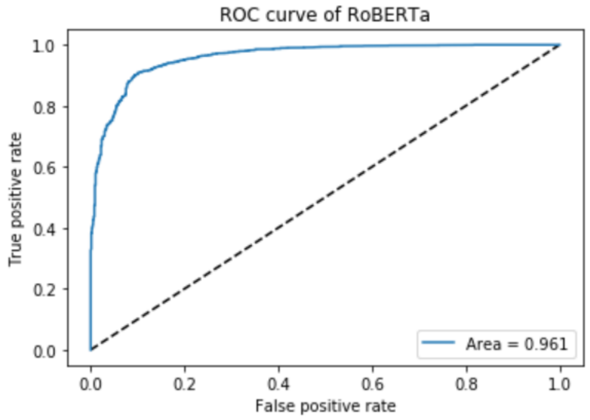
Classes	RoBERTa	ULMFiT
(1,2),(4,5)	95.63	92.47
(1,2,3),(4,5)	91.01	86.86
(1,2), (3,4,5)	92	90.12
(1,2),(3),(4,5)	85.23	81.78
(1),(2),(3),(4), (5)	67.24	61.76

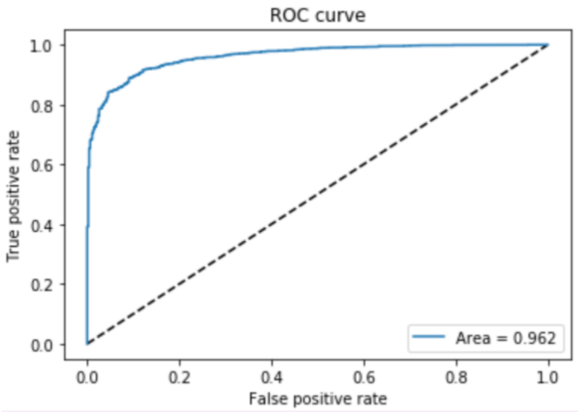








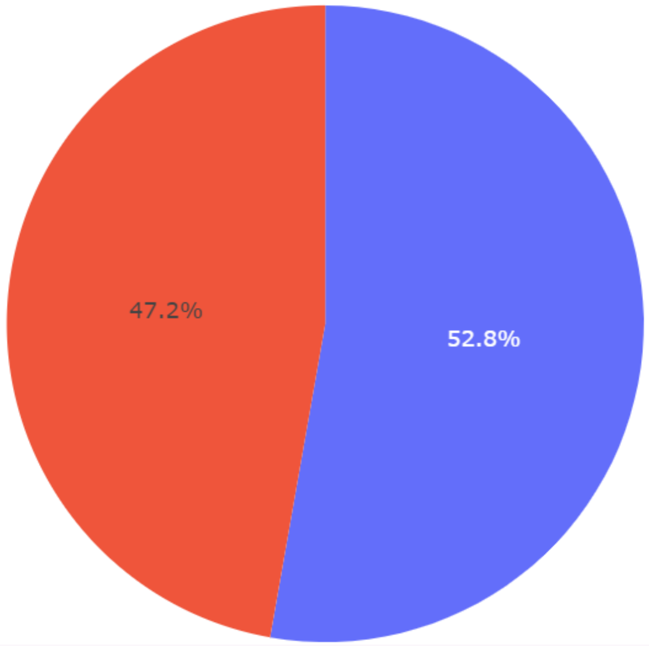




Prediction Demo

```
The review class is:  Neg , with a scale of  52.77  out of 100.
```

this phone case is cute, but it's not that sturdy...



https://www.youtube.com/watch?v=I2JlqNj5FBE

Thank You!