Utilizing the Transformer models for Analysing Deceptive Reviews and Aspects of the reviews

MSc Research Project
MSc Data Analytics

Santhosh Vinayagamurthy
Student ID: 20186177

School of Computing
National College of Ireland

Supervisor: Prashanth Nayak

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Santhosh Vinayagamurthy……………………………….

Student ID: X20186177………………………………………………

Programme: MSc Data Analytics……. **Year:** 2021………………………..

Module: Research project…………………………………………………………

Lecturer: Prashanth Nayak………………………………………………………………………
Submission Due Date: 15-12-2022…………………………..……

Project Title: … Utilizing the Transformer models for Analysing Deceptive Reviews and Aspects of the reviews………

…852……………………………… **Page Count:**
Word Count: ………9…………………………..…….………

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Santhosh V………………………………………………………………………

Date: 15-12-2022………………………………………………………………………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Santhosh Vinayagamurthy
20186177:

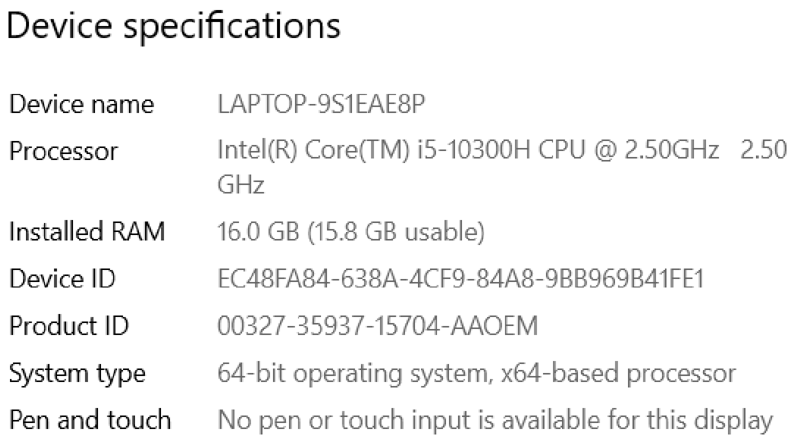
1 Introduction

The hardware and the software which are used in this research project have been described in this manual. The code which has been written for this project is explained here step by step.

2 Hardware and Software Used:

To implement fake review detection the following hardware and software have been used. The python version used is 3.7. The experiment was carried out in the Google colab with a Gmail account. The dataset used here is a publicly available dataset

Hardware Details.



Device specifications

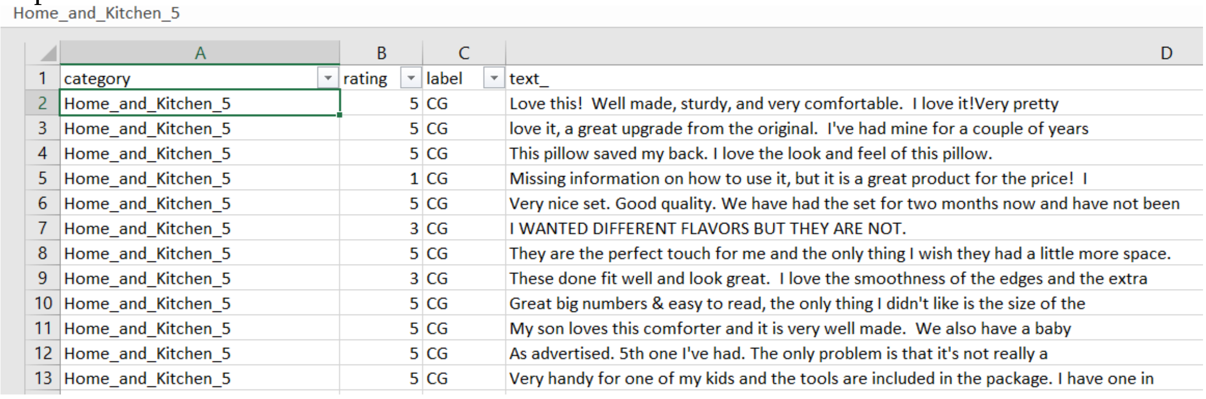
Device name	LAPTOP-9S1EAE8P
Processor	Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz 2.50 GHz
Installed RAM	16.0 GB (15.8 GB usable)
Device ID	EC48FA84-638A-4CF9-84A8-9BB969B41FE1
Product ID	00327-35937-15704-AAOEM
System type	64-bit operating system, x64-based processor
Pen and touch	No pen or touch input is available for this display

Software Details

IDE	PROGRAMMING LANGUAGE	FRAMEWORK/LIBRARY	GPU TYPE	Number of GPU
Google Colab(trail version)	Python	SimpleTransformer, HuggingFace Transformer, Sklearn,Pandas,Numpy,PyTorch, PYABSA	Used the trial version GPU Persistence-M	1

3 Dataset

The Dataset used in this project is in the format of CSV. The original reviews are derived from the Amazon customer and the fake review created based on these reviews using the GPT. There are totally four columns Category, Label, Rating, and Text. The Label and the text are important columns that are used for this detection.

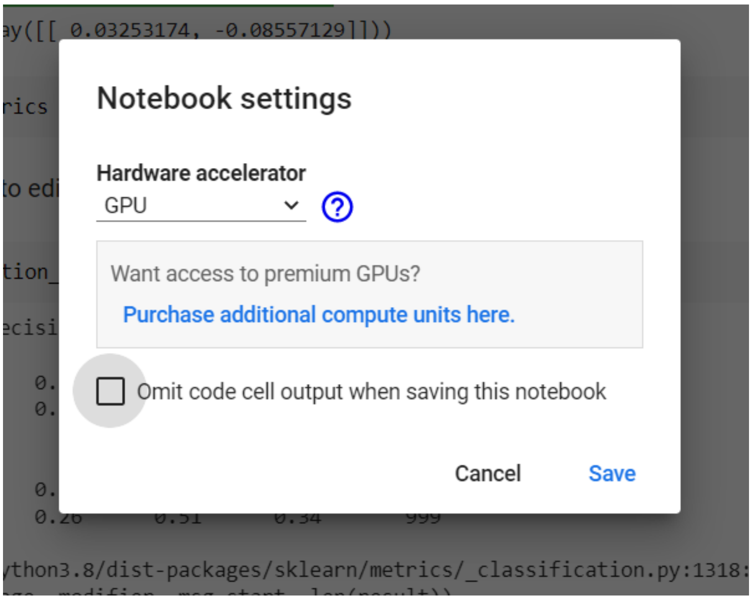


Home_and_Kitchen_5

	A	B	C	D
1	category	rating	label	text_
2	Home_and_Kitchen_5	5	CG	Love this! Well made, sturdy, and very comfortable. I love it!Very pretty
3	Home_and_Kitchen_5	5	CG	love it, a great upgrade from the original. I've had mine for a couple of years
4	Home_and_Kitchen_5	5	CG	This pillow saved my back. I love the look and feel of this pillow.
5	Home_and_Kitchen_5	1	CG	Missing information on how to use it, but it is a great product for the price! I
6	Home_and_Kitchen_5	5	CG	Very nice set. Good quality. We have had the set for two months now and have not been
7	Home_and_Kitchen_5	3	CG	I WANTED DIFFERENT FLAVORS BUT THEY ARE NOT.
8	Home_and_Kitchen_5	5	CG	They are the perfect touch for me and the only thing I wish they had a little more space.
9	Home_and_Kitchen_5	3	CG	These done fit well and look great. I love the smoothness of the edges and the extra
10	Home_and_Kitchen_5	5	CG	Great big numbers & easy to read, the only thing I didn't like is the size of the
11	Home_and_Kitchen_5	5	CG	My son loves this comforter and it is very well made. We also have a baby
12	Home_and_Kitchen_5	5	CG	As advertised. 5th one I've had. The only problem is that it's not really a
13	Home_and_Kitchen_5	5	CG	Very handy for one of my kids and the tools are included in the package. I have one in

4 Implementation

In the Colab, change the runtime setting to make use of the GPU provided by the google colab.



```
ay([[ 0.03253174, -0.08557129]])
```

Notebook settings

Hardware accelerator
GPU

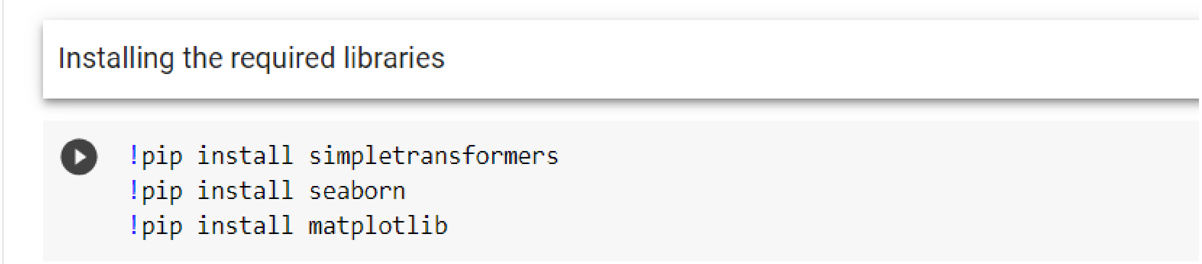
Want access to premium GPUs?
Purchase additional compute units here.

□ Omit code cell output when saving this notebook

```
ython3.8/dist-packages/sklearn/metrics/_classification.py:1318:
```

4.1 Installing the required libraries

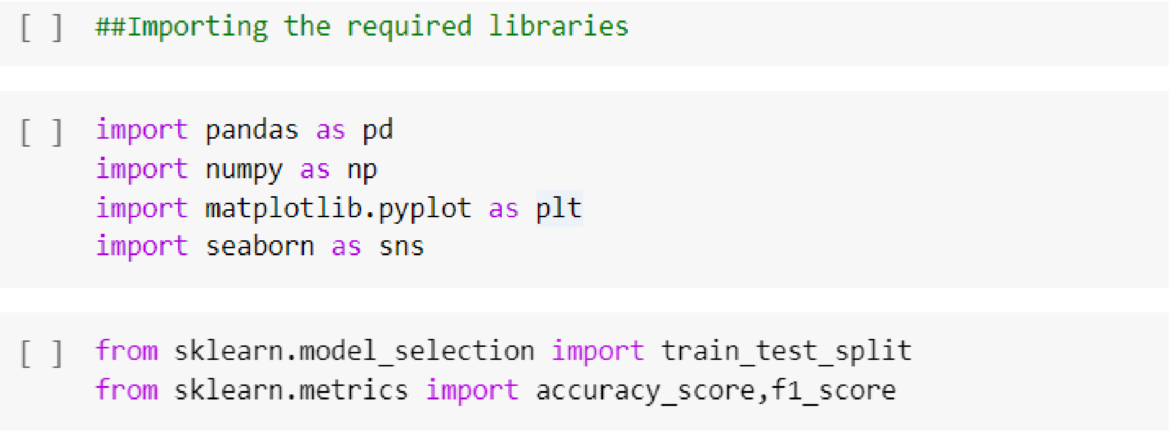
The Simpletransformer, Seaborn, and Matplot libraries were installed for the implementation of the transformer models and visualization.



Installing the required libraries

```
!pip install simpletransformers
!pip install seaborn
!pip install matplotlib
```

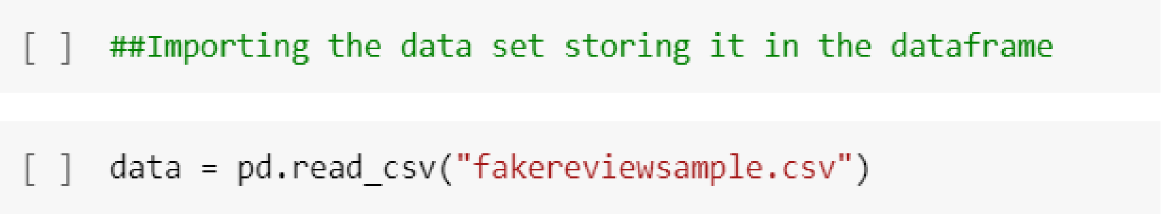
Once these libraries are installed they are imported.



```
[ ]  ##Importing the required libraries

[ ]  import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     import seaborn as sns

[ ]  from sklearn.model_selection import train_test_split
     from sklearn.metrics import accuracy_score,f1_score
```

4.2 Importing the data set and storing it in the data frame.

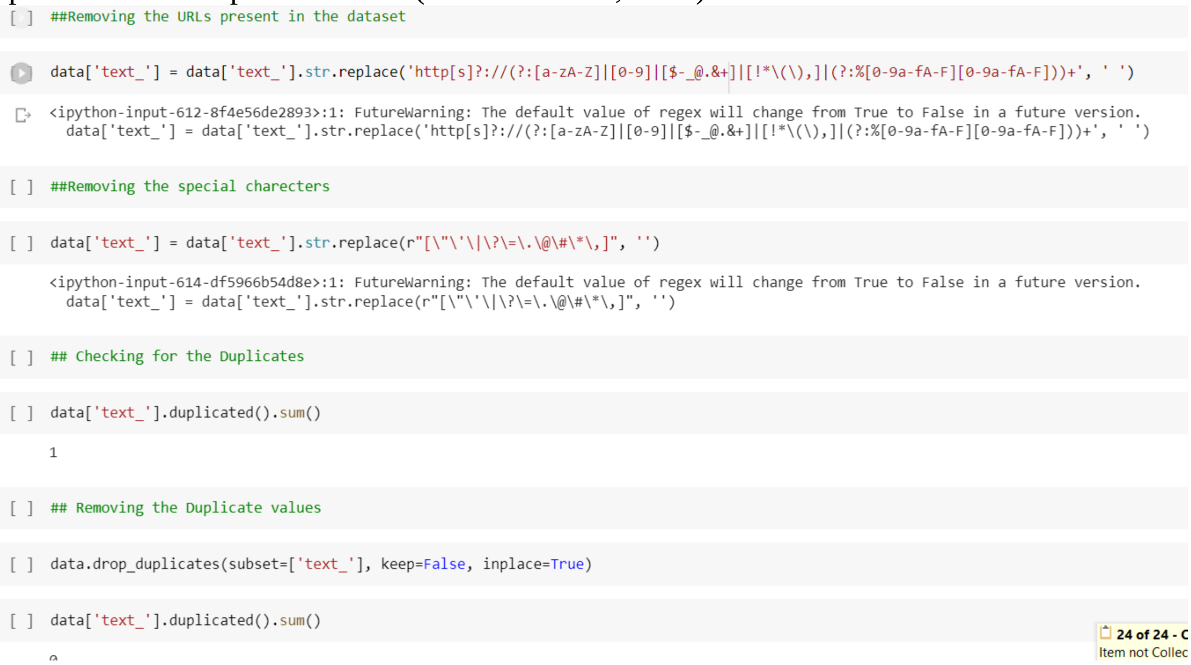
The Dataset has been imported and stored in the Dataframe.



```
[ ]  ##Importing the data set storing it in the dataframe

[ ]  data = pd.read_csv("fakereviewsample.csv")
```

4.3 Data Pre-processing phase

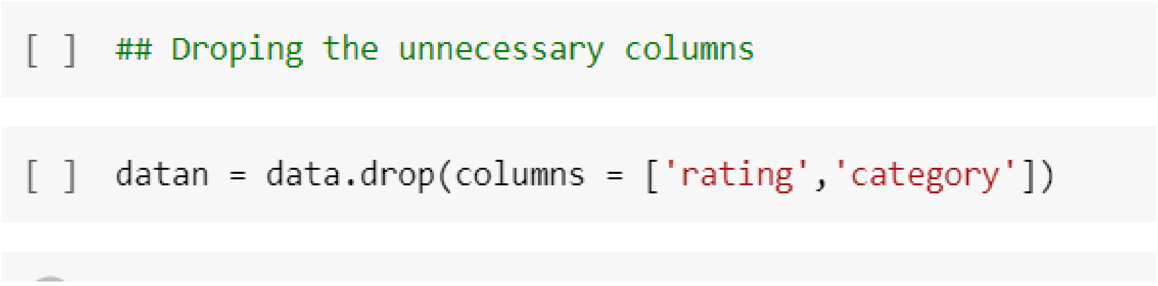
In this, null checks, the URL, Special character, and duplicates are checked. In this dataset, there is only one duplicate value that has been removed. The URL and Special character are replaced with the Space character(Salminen et al., 2022).



```
[ ]  ##Removing the URLs present in the dataset

     data['text_'] = data['text_'].str.replace('http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+]|[!*\(\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+', ' ')

     <ipython-input-612-8f4e56de2893>:1: FutureWarning: The default value of regex will change from True to False in a future version.
       data['text_'] = data['text_'].str.replace('http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+]|[!*\(\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+', ' ')

[ ]  ##Removing the special charecters

[ ]  data['text_'] = data['text_'].str.replace(r"[\"\'\|\?\=\.\@\#\*\,]", '')

     <ipython-input-614-df5966b54d8e>:1: FutureWarning: The default value of regex will change from True to False in a future version.
       data['text_'] = data['text_'].str.replace(r"[\"\'\|\?\=\.\@\#\*\,]", '')

[ ]  ## Checking for the Duplicates

[ ]  data['text_'].duplicated().sum()

     1

[ ]  ## Removing the Duplicate values

[ ]  data.drop_duplicates(subset=['text_'], keep=False, inplace=True)

[ ]  data['text_'].duplicated().sum()

     0
```

4.4 Dropping the Unnecessary columns.

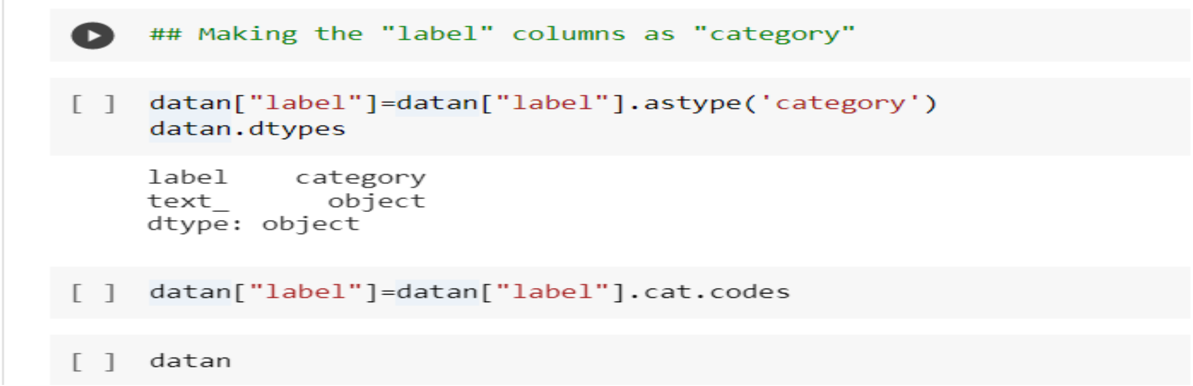
The columns which are not useful for this prediction have been dropped.



```
[ ]  ## Droping the unnecessary columns

[ ]  datan = data.drop(columns = ['rating','category'])
```

4.5 Encoding the category variable

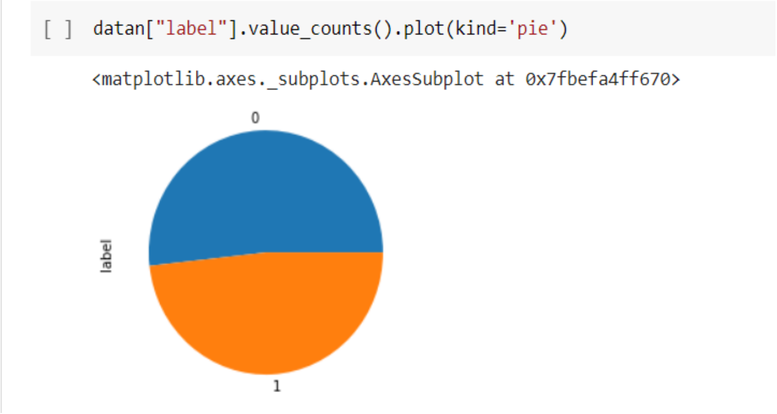
The category variable (Label columns) has been encoded and stored in the data frame.



```
     ## Making the "label" columns as "category"

[ ]  datan["label"]=datan["label"].astype('category')
     datan.dtypes

     label      category
     text_        object
     dtype: object

[ ]  datan["label"]=datan["label"].cat.codes

[ ]  datan
```

4.6 Distribution of label column.

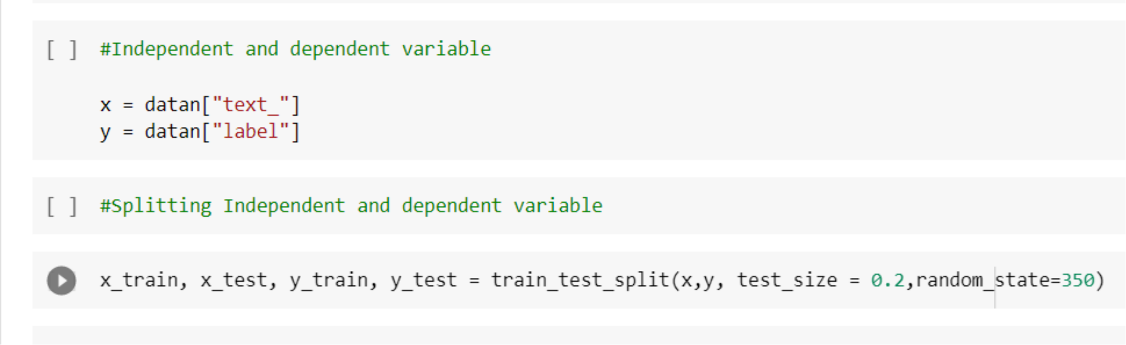
The label column contains an equal amount of fake reviews and original reviews.



```
[ ]  datan["label"].value_counts().plot(kind='pie')

     <matplotlib.axes._subplots.AxesSubplot at 0x7fbefa4ff670>
```

4.7 Splitting the Dataset

The text and the label columns are stored in separate dependent and independent variables x and y respectively. After that, the data will be split into test and training sets.



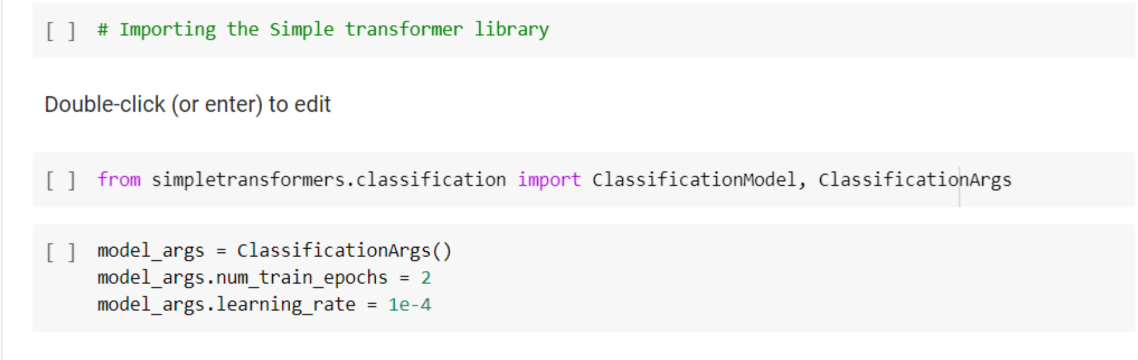
```
[ ]  #Independent and dependent variable

     x = datan["text_"]
     y = datan["label"]
```

```
[ ]  #Splitting Independent and dependent variable
```

```
     x_train, x_test, y_train, y_test = train_test_split(x,y, test_size = 0.2,random_state=350)
```

4.8 Importing the Simple transformer library

The transformer models are imported and the Epoch, training values are set.



```
[ ]  # Importing the Simple transformer library
```

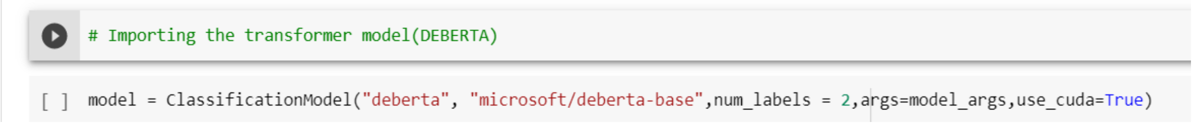
Double-click (or enter) to edit

```
[ ]  from simpletransformers.classification import ClassificationModel, ClassificationArgs
```

```
[ ]  model_args = ClassificationArgs()
     model_args.num_train_epochs = 2
     model_args.learning_rate = 1e-4
```

4.9 Importing the transformer model(DEBERTA)

The DeBERTa model has been imported from the transformer model.

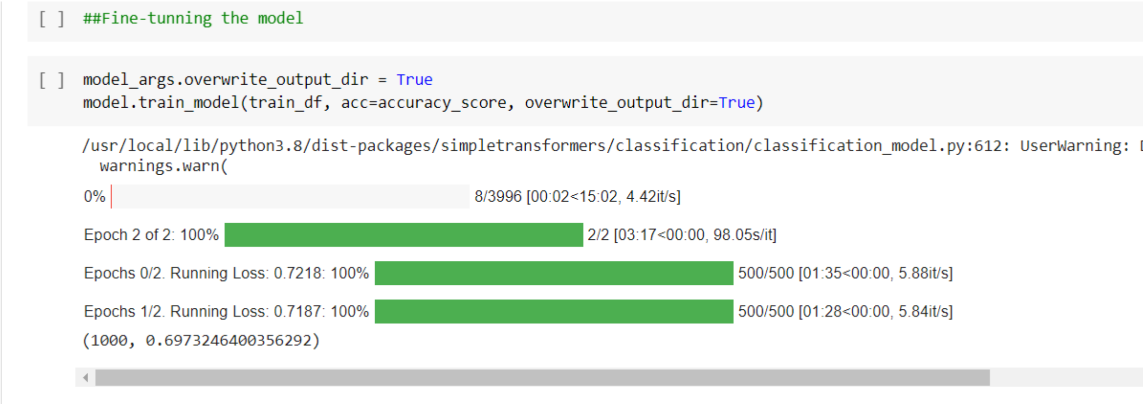


```
     # Importing the transformer model(DEBERTA)
```

```
[ ]  model = ClassificationModel("deberta", "microsoft/deberta-base",num_labels = 2,args=model_args,use_cuda=True)
```

4.10 Fine-tuning the model

The model is finetuned with the training data.



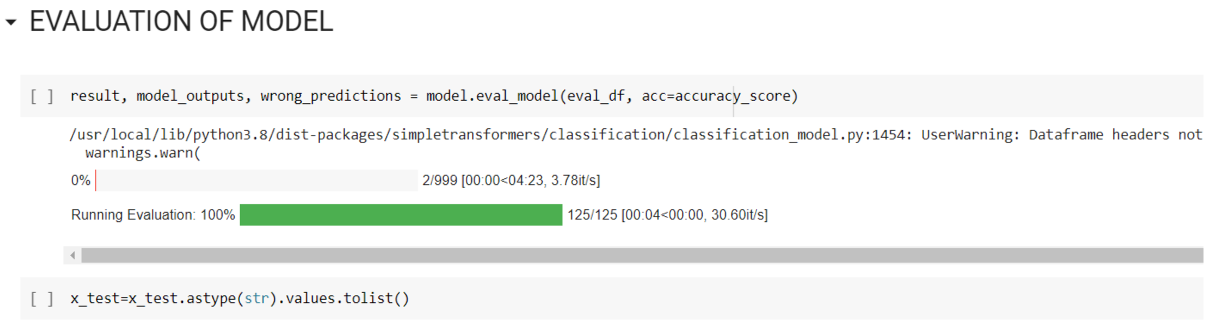
```
[ ]  ##Fine-tunning the model
```

```
[ ]  model_args.overwrite_output_dir = True
     model.train_model(train_df, acc=accuracy_score, overwrite_output_dir=True)
```

```
/usr/local/lib/python3.8/dist-packages/simpletransformers/classification/classification_model.py:612: UserWarning: 
  warnings.warn(
0%                                            8/3996 [00:02<15:02, 4.42it/s]
Epoch 2 of 2: 100%                            2/2 [03:17<00:00, 98.05s/it]
Epochs 0/2. Running Loss: 0.7218: 100%        500/500 [01:35<00:00, 5.88it/s]
Epochs 1/2. Running Loss: 0.7187: 100%        500/500 [01:28<00:00, 5.84it/s]
(1000, 0.6973246400356292)
```

4.11 Prediction and evaluation

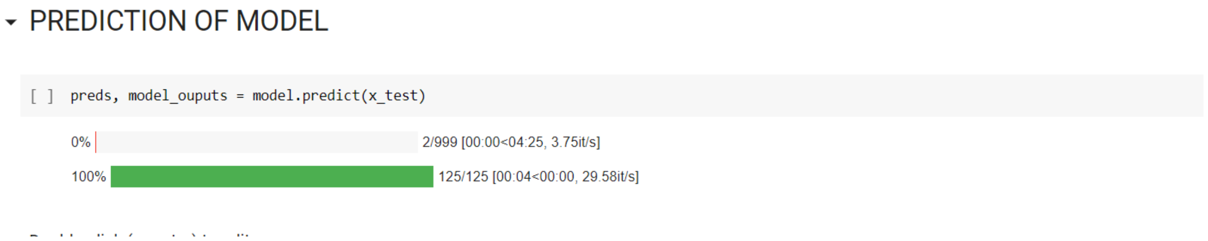
The model evaluation and prediction have been done.



EVALUATION OF MODEL

```
[ ] result, model_outputs, wrong_predictions = model.eval_model(eval_df, acc=accuracy_score)

    /usr/local/lib/python3.8/dist-packages/simpletransformers/classification/classification_model.py:1454: UserWarning: Dataframe headers not
      warnings.warn(
    0%|          | 2/999 [00:00<04:23, 3.78it/s]
    Running Evaluation: 100%|██████████| 125/125 [00:04<00:00, 30.60it/s]
```

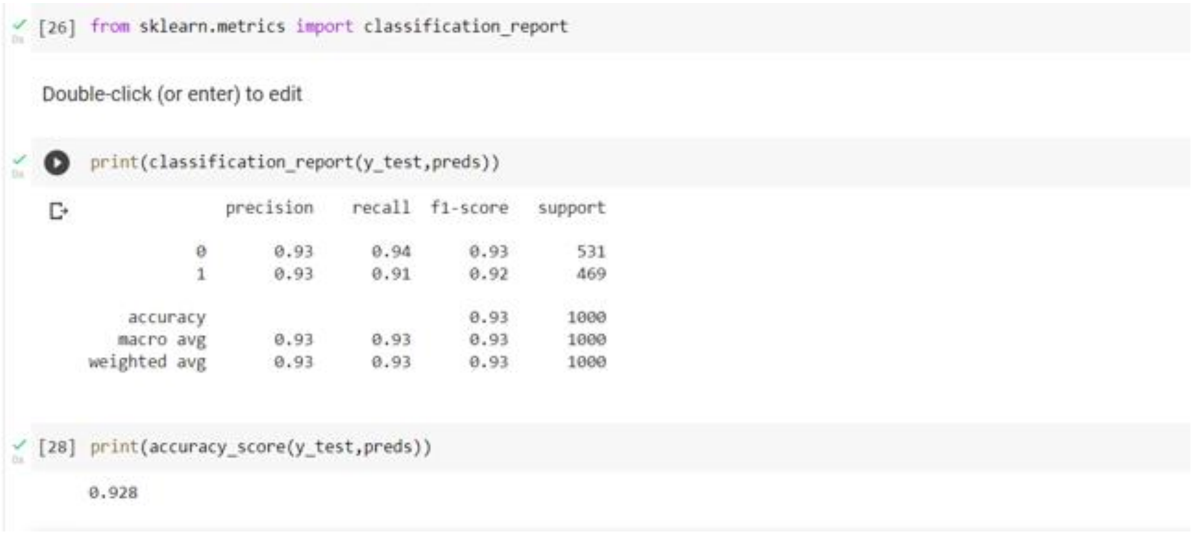
```
[ ] x_test=x_test.astype(str).values.tolist()
```



PREDICTION OF MODEL

```
[ ] preds, model_ouputs = model.predict(x_test)

    0%|          | 2/999 [00:00<04:25, 3.75it/s]
    100%|██████████| 125/125 [00:04<00:00, 29.58it/s]
```

4.12 Results



```
[26] from sklearn.metrics import classification_report
```

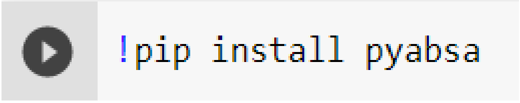
Double-click (or enter) to edit

```
print(classification_report(y_test,preds))

              precision    recall  f1-score   support

           0       0.93      0.94      0.93       531
           1       0.93      0.91      0.92       469

    accuracy                           0.93      1000
   macro avg       0.93      0.93      0.93      1000
weighted avg       0.93      0.93      0.93      1000
```

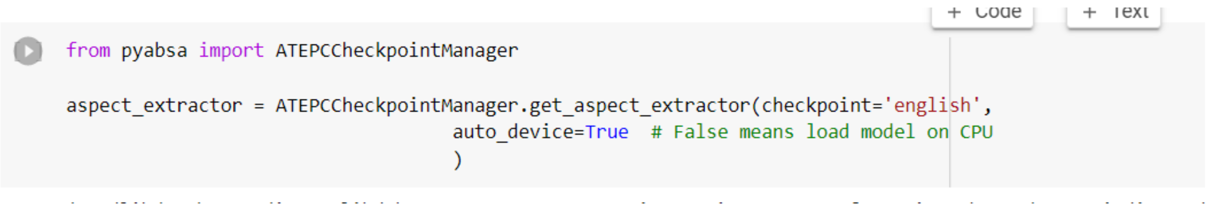
```
[28] print(accuracy_score(y_test,preds))

     0.928
```

5 Analysing the Aspect of the reviews

5.1 Importing the library.



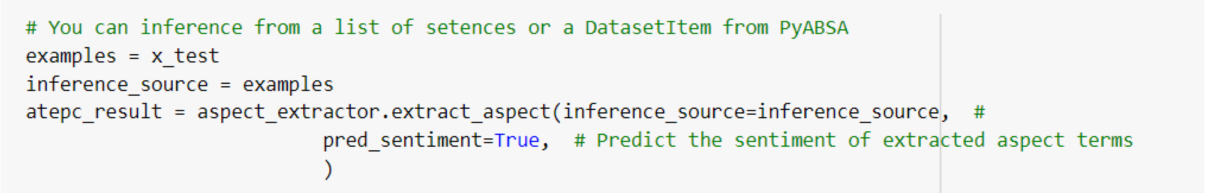
```
!pip install pyabsa
```



```
from pyabsa import ATEPCCheckpointManager

aspect_extractor = ATEPCCheckpointManager.get_aspect_extractor(checkpoint='english',
                                                                auto_device=True  # False means load model on CPU
                                                                )
```

5.2 Finding the aspects and sentiment.

Passing the data into the aspect extractor and soring the results in the ate pec result variable.



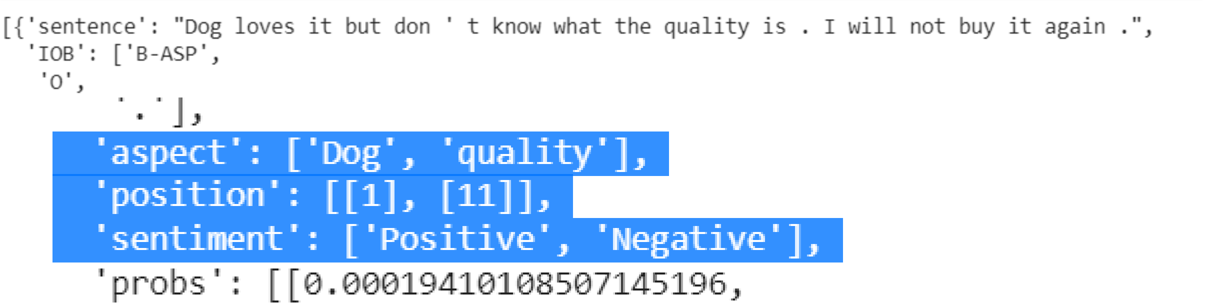
```
# You can inference from a list of setences or a DatasetItem from PyABSA
examples = x_test
inference_source = examples
atepc_result = aspect_extractor.extract_aspect(inference_source=inference_source,  #
                                               pred_sentiment=True,  # Predict the sentiment of extracted aspect terms
                                               )
```

5.3 Results

Results show the sentiment present in the sentence and its aspect and its sentiments.

For example

In this sentence "Dog loves it but don’t know what the quality is. I will not buy it again ." The aspect is Dog and the quality. The first part of the sentence is positive and the second part of the sentence is negative which is predicted by the model correctly.



```
[{'sentence': "Dog loves it but don ' t know what the quality is . I will not buy it again .",
  'IOB': ['B-ASP',
   'O',
      '.'],
   'aspect': ['Dog', 'quality'],
   'position': [[1], [11]],
   'sentiment': ['Positive', 'Negative'],
   'probs': [[0.00019410108507145196,
```

References

Salminen, J., Kandpal, C., Kamel, A.M., Jung, S. gyo, Jansen, B.J., 2022. Creating and detecting fake reviews of online products. Journal of Retailing and Consumer Services 64. https://doi.org/10.1016/j.jretconser.2021.102771