

Configuration Manual

MSc Research Project
Data Analytics

Deepak Kumar
Student ID: x20195028

School of Computing
National College of Ireland

Supervisor: Mr. Prashanth Nayak

National College of Ireland
MSc Project Submission Sheet
School of Computing



Student Name: Deepak Kumar
Student ID: X20195028
Programme: M.Sc Data Analytics **Year:** 2022
Module: M.Sc Research Project
Supervisor: Mr. Prashanth Nayak
Submission Due Date: 01/Feb/2023
Project Title: Configuration Manual
Word Count: 1373 **Page Count:** 10

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: Deepak Kumar

Date: 29/Jan/2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	<input type="checkbox"/>
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.	<input type="checkbox"/>

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

Deepak Kumar
X20195028

Introduction

This configuration manual gives a clear understanding of the research project and all the tools, and software required to generate the results for “Gender Prediction Based on various Nationality Names using Deep Learning techniques”. All the dependencies and requirement which cannot be mentioned in the Project report is recommended to be present in the configuration manual. The objective of the research is mentioned in the report as well “How well can Sequence Classification be applied to the various nationalities and names of individuals to predict gender using deep learning Techniques?”. And for this work, four deep learning techniques were implemented to find out which one is the best and why? BERT, DistilBERT, XLNet, and RoBERTa have been used and the dataset for this research is acquired from UCI Machine Learning Repository which is open-source and licensed for academic work. This manual will be helpful to understand not only to configure the tools and software but also how to reproduce the results for this work.

1. Software and Hardware Specifications

For this research project below mentioned Software and Hardware specification have been used for pre-processing, EDA (Exploratory Data Analysis), training and evaluation.

1.1 Software Specifications

Programming Language	Python
Development Tool	Google Colab Pro, Anaconda Navigator, Spyder
Other Tools	Numbers and Microsoft Excel

Table 1.1: Software Requirement

1.2 Hardware Specifications

System	Specification
Mac OS edition	Ventura 13.0.1
Processor	Apple M2 chip
RAM	16 GB
CPU/GPU	8-core CPU, 8-core GPU
Neural Engine	16-core Neural Engine

Table 1.2: Hardware Requirement

2. Software Installation Guide

2.1. Installation of the Google Colab Pro

- 2.1.1. Open the URL using browser preferred with chrome
“<https://colab.research.google.com/>”
- 2.1.2. Purchase the Plan as per the required Compute Units
- 2.1.3. Open the editor and keep your code in the cells or upload the code file using path File < - Open Notebook < - Browse < - Upload.

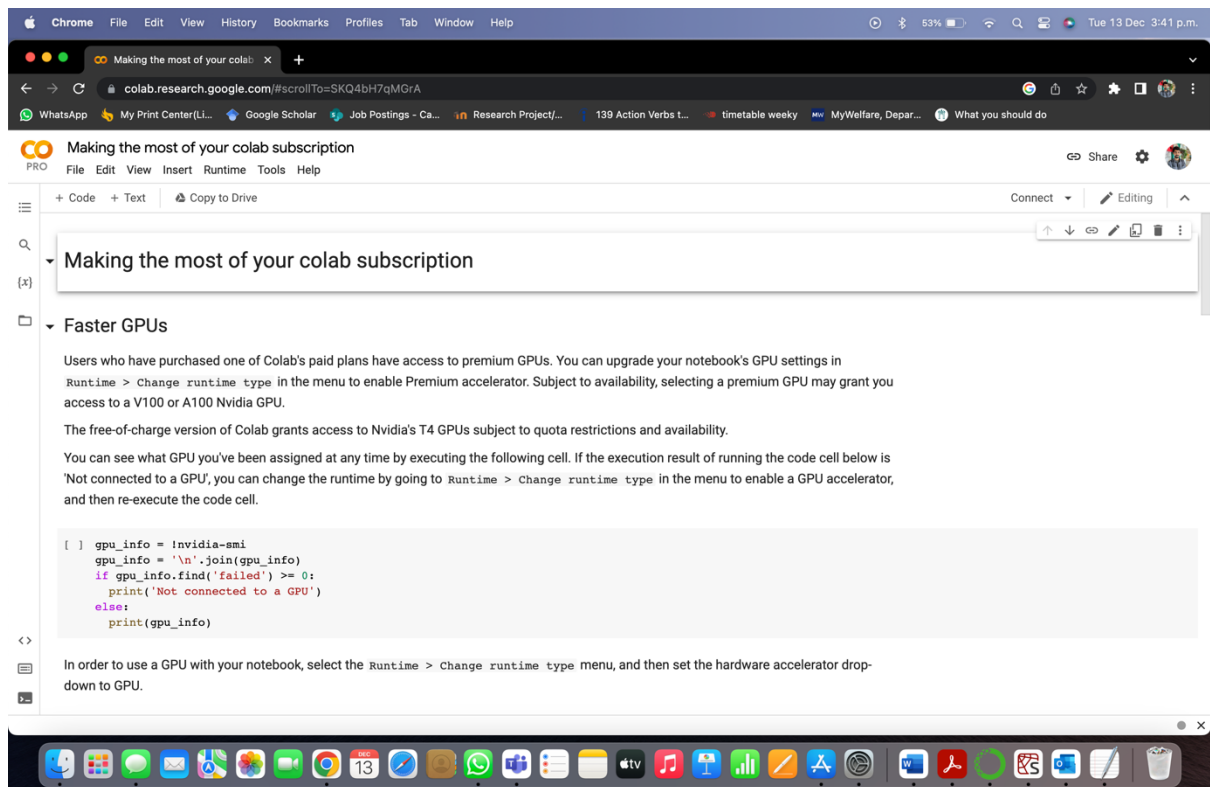


Fig 2.1: Google Colab Pro Editor

2.2. Installation of Anaconda Navigator

- 2.2.1. Open URL <https://docs.anaconda.com/anaconda/install/mac-os/> using browser and download the mac OS installer.
- 2.2.2. Open the file and click continue to install the anaconda in machine.
- 2.2.3. Read the introduction, read me and license section of installer guide.
- 2.2.4. Then select the destination for installation and select “for me only” and proceed with continue button as shown in below figure.

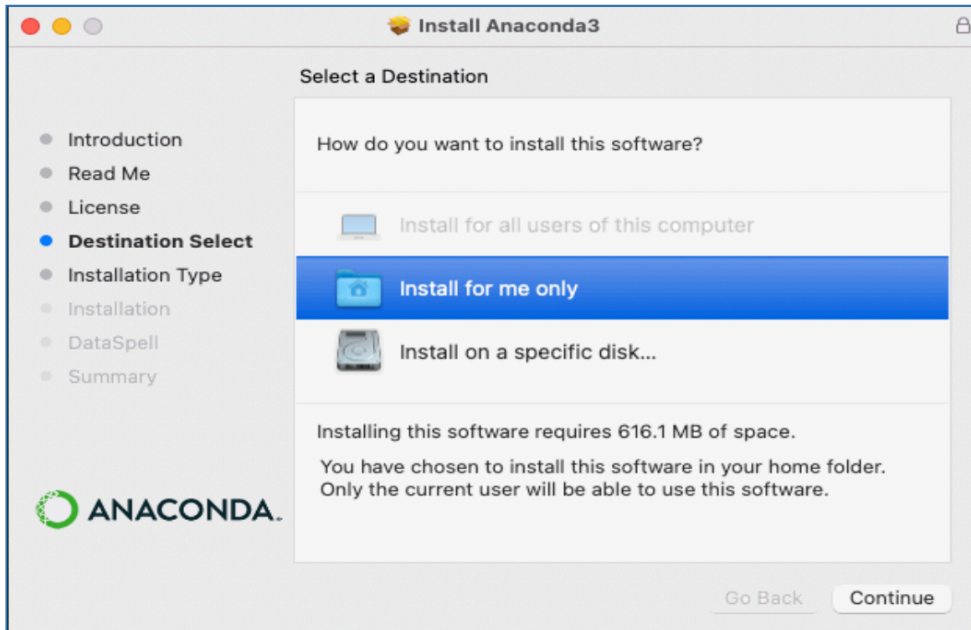


Fig 2.2: Select Destination

2.2.5. Next step, to select the type of installation, here the directory for the root file can be selected.

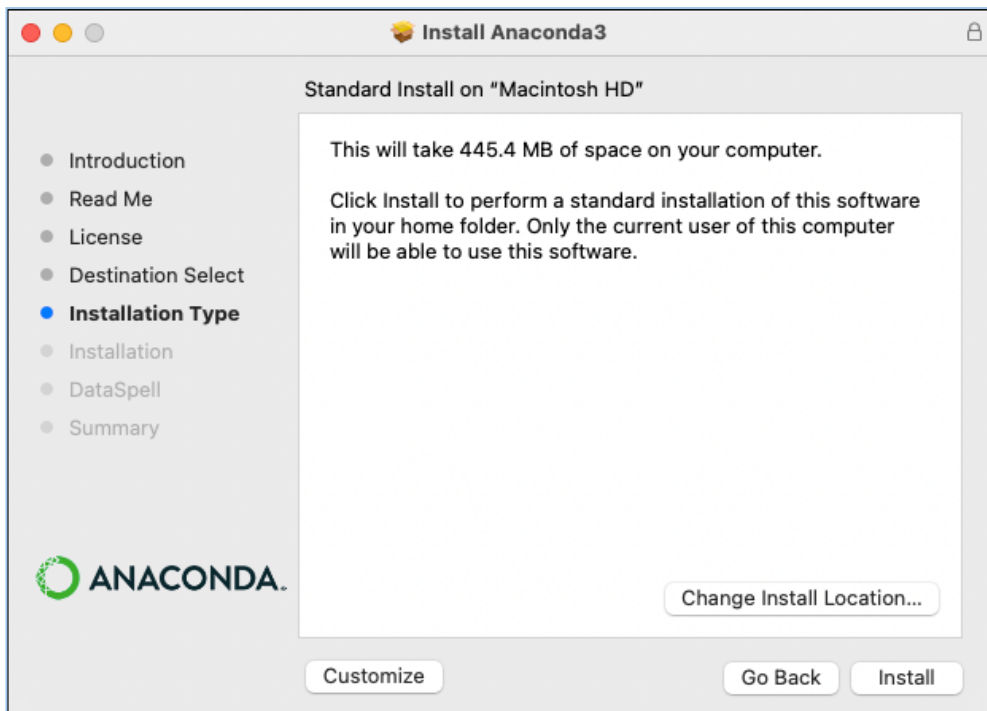


Fig 2.2: Installation Type

2.2.6. Click on install and continue with Dataspell and summary and finish the installation process.

2.2.7. Later, Anaconda Navigator may ask for the latest updates and can be updated using it.

2.2.8. Open anaconda Navigator and Spyder is ready to use in root environment.

3. Pre-processing and EDA

Basic Operations like converting the text into lower-case, removing the special characters, punctuation, etc. The gender column has been encoded in integer type were 1 is female and 0 is male. The datasets column header is shown in the below figure with the top five row data.

```
In [61]: df.head()
Out[61]:
```

	Name	Gender
0	jossie	0
1	denica	1
2	bricola	1
3	audranna	1
4	quatavius	0

Fig3.1: Top 5 rows of encoded data

Name length has been visualised based on the alphabet in a name. The count of the names for the different character lengths has been displayed in the below image. The most common was for alphabets for more than 17 thousand names.

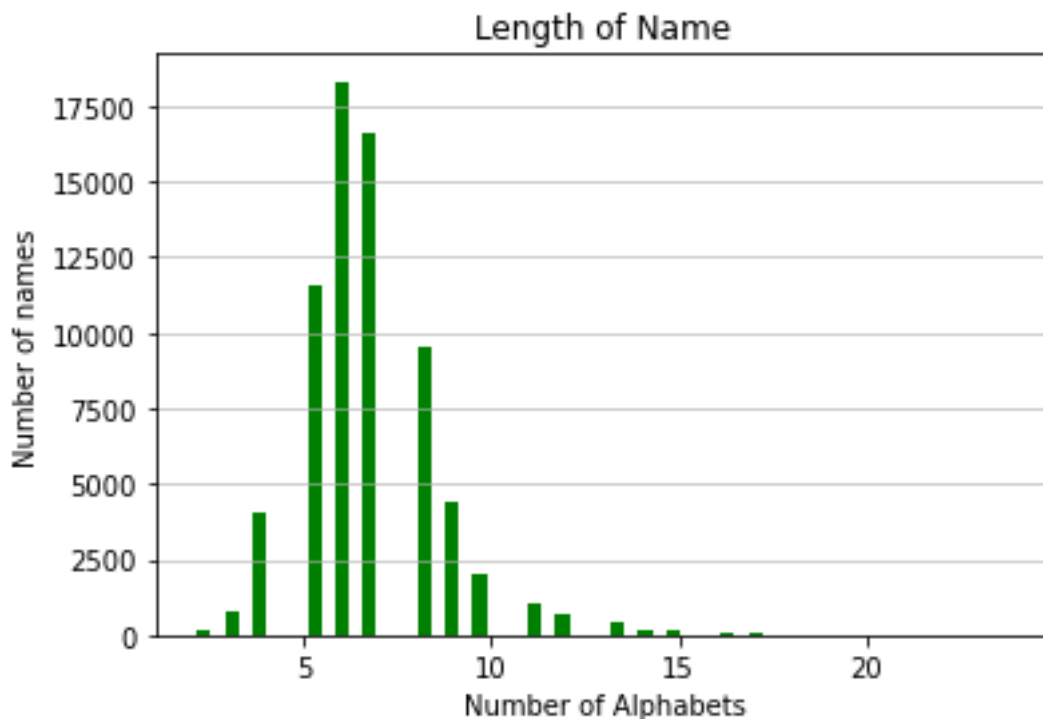


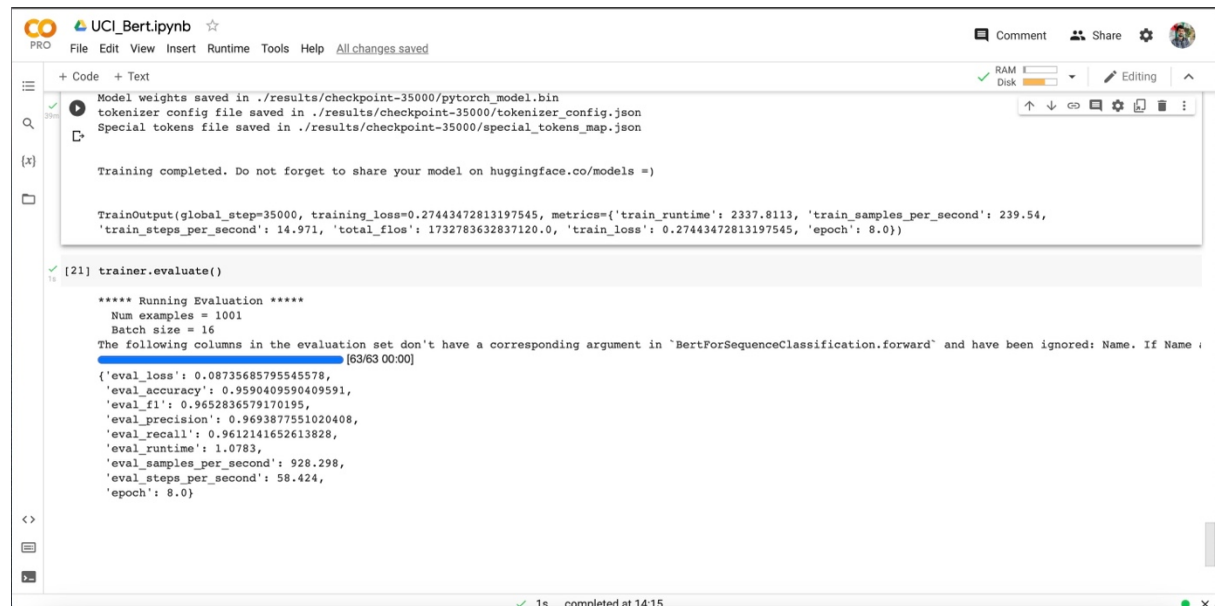
Fig 3.2: Length of the names visualised using histogram

4. Experiments

Model implementation has been covered in this section. This section gives a clear understanding of the implementation of all four models i.e. BERT, DistilBERT, XLNet, RoBERTa. Dependencies of the libraries and package are separately given in table with their specific versions used. In the artefacts folder, each model has its python code file with its respective name which has been uploaded separately. For the required model respective python file can be downloaded and executed to reproduce the results. Also, file names can be seen in the attached screenshots.

4.1. BERT

In this research work, this model has been considered as the best performing model with 95.90 per cent of accuracy but the time consumed by this model is satisfactory but not as faster as compared to DistilBERT.



```
Model weights saved in ./results/checkpoint-35000/pytorch_model.bin
tokenizer config file saved in ./results/checkpoint-35000/tokenizer_config.json
Special tokens file saved in ./results/checkpoint-35000/special_tokens_map.json

Training completed. Do not forget to share your model on huggingface.co/models =)

TrainOutput(global_step=35000, training_loss=0.27443472813197545, metrics={'train_runtime': 2337.8113, 'train_samples_per_second': 239.54,
'train_steps_per_second': 14.971, 'total_flos': 1732783632837120.0, 'train_loss': 0.27443472813197545, 'epoch': 8.0})

[21] trainer.evaluate()

***** Running Evaluation *****
Num examples = 1001
Batch size = 16
The following columns in the evaluation set don't have a corresponding argument in `BertForSequenceClassification.forward` and have been ignored: Name. If Name :
[63/63 00:00]
{'eval_loss': 0.08735685795545578,
 'eval_accuracy': 0.9590409590409591,
 'eval_f1': 0.9652836579170195,
 'eval_precision': 0.9693877551020408,
 'eval_recall': 0.9612141652613828,
 'eval_runtime': 1.0783,
 'eval_samples_per_second': 928.298,
 'eval_steps_per_second': 58.424,
 'epoch': 8.0}
```

Fig 4.1 Results for BERT with 8 epochs

4.2. DistilBERT

This model has performed well and is also considered one of the best models in this research. However, BERT is found more accurate than this with very less difference in accuracy. DistilBERT is considered faster and more efficient which is proved by this model in this study as well. DistilBERT showed 95.40 per cent of accuracy in the least time frame of 19 minutes.

```

11500      0.334100

trainer.evaluate()
***** Running Evaluation *****
Num examples = 1001
Batch size = 16
The following columns in the evaluation set don't have a corresponding argument in `DistilBertForSequenceClassification.forward` and have been ignored: Name. If
[63/63 00.00]
{'eval_loss': 0.09496887028217316,
 'eval_accuracy': 0.954045954045954,
 'eval_f1': 0.9609507640067911,
 'eval_precision': 0.9675213675213675,
 'eval_recall': 0.954468802698145,
 'eval_runtime': 0.5882,
 'eval_samples_per_second': 1701.687,
 'eval_steps_per_second': 107.099,
 'epoch': 8.0}
  
```

Fig 4.2: Results for DistilBERT with 8 epochs

4.3.XLNet

XLNet has shown satisfactory results with 88.21 per cent of accuracy. however, this is the least accuracy achieved in comparison to all models used in this study. Additionally, previous work mentioned in the related work section has some models which are not as accurate as XLNet.

```

Special tokens file saved in ./results/checkpoint-34500/special_tokens_map.json
Saving model checkpoint to ./results/checkpoint-35000
Configuration saved in ./results/checkpoint-35000/config.json
Model weights saved in ./results/checkpoint-35000/pytorch_model.bin
tokenizer config file saved in ./results/checkpoint-35000/tokenizer_config.json
Special tokens file saved in ./results/checkpoint-35000/special_tokens_map.json

Training completed. Do not forget to share your model on huggingface.co/models =)

TrainOutput(global_step=35000, training_loss=0.3869327144077846, metrics={'train_runtime': 3900.1329, 'train_samples_per_second': 143.585,
'train_steps_per_second': 8.974, 'total_flos': 1996254046416960.0, 'train_loss': 0.3869327144077846, 'epoch': 8.0})

trainer.evaluate()
***** Running Evaluation *****
Num examples = 1001
Batch size = 16
The following columns in the evaluation set don't have a corresponding argument in `XLNetForSequenceClassification.forward` and have been ignored: Name. If Name
[63/63 00.01]
{'eval_loss': 0.2631475031375885,
 'eval_accuracy': 0.8821178821178821,
 'eval_f1': 0.9005059021922428,
 'eval_precision': 0.9005059021922428,
 'eval_recall': 0.9005059021922428,
 'eval_runtime': 1.3925,
 'eval_samples_per_second': 718.85,
 'eval_steps_per_second': 45.242,
 'epoch': 8.0}
  
```

Fig 4.2: Results for XLNet with 8 epochs

4.4. RoBERTa

This model is not as much as accurate as compared to DistilBERT and BERT however it has shown an accuracy of 91.80% which is quite acceptable and if compared to previous models which were implemented using traditional machine and deep learning techniques it has shown better results.

```

Saving model checkpoint to ./results/checkpoint-35000
Configuration saved in ./results/checkpoint-35000/config.json
Model weights saved in ./results/checkpoint-35000/pytorch_model.bin
tokenizer config file saved in ./results/checkpoint-35000/tokenizer_config.json
Special tokens file saved in ./results/checkpoint-35000/special_tokens_map.json

Training completed. Do not forget to share your model on huggingface.co/models =)

TrainOutput(global_step=35000, training_loss=0.35578491864885603, metrics={'train_runtime': 2397.0314, 'train_samples_per_second': 233.622,
'train_steps_per_second': 14.601, 'total_flos': 1919765148772800.0, 'train_loss': 0.35578491864885603, 'epoch': 8.0})

trainer.evaluate()

***** Running Evaluation *****
  Num examples = 1001
  Batch size = 16
  The following columns in the evaluation set don't have a corresponding argument in `RobertaForSequenceClassification.forward` and have been ignored: Name. If Na
  [63/63 00:00]
{'eval_loss': 0.1914263367652893,
 'eval_accuracy': 0.9180819180819181,
 'eval_f1': 0.9307432432432433,
 'eval_precision': 0.9323181049069373,
 'eval_recall': 0.9291736930860034,
 'eval_runtime': 0.9753,
 'eval_samples_per_second': 1026.386,
 'eval_steps_per_second': 64.598,
 'epoch': 8.0}
  
```

Fig 4.2: Results for RoBERTa with 8 epochs

4.5. Training Parameters

Training arguments of the models like learning rate, the batch size for training, batch size of evaluation, numbers of epochs, and weight decay are kept the same to fairly compare the results of these models. The learning rate is $2e-5$ (0.00002), and the batch size for training and testing is 16. All the models are executed with 8 epochs and the weight decay is 0.01. All the results were interpreted using the compute metric function using packages of the sklearn library.

5. Comparison of Developed Models

Results obtained for all models are tabulated and shown in below table.

Model	DistilBERT	BERT	XL Net	RoBERTa
Evaluation Loss (Test Set)	0.09	0.08	0.26	0.19
Accuracy	95.4	95.9	88.21	91.8
F1 Score	96.09	96.52	90.05	93.07
Precision	96.75	96.93	90.05	93.23

Recall	95.44	96.12	90.05	92.91
Epochs	8	8	8	8
Execution Time(mins)	19:04	38:54	64:57	39:54

Table 5.1: Experiment Results

6. Conclusion

Overall, it can be concluded easily that DistilBERT and BERT has produced the better results than others. In view of time; BERT has generated the almost same and accurate results in 19 minutes which is quite faster as compared to BERT because BERT consumed almost 39 minutes for the same work and with very little variation in accuracy. Since, objective of the study has been achieved with DistilBERT and BERT models for text classification domain.