Gender Detection Project Report

The project is to build a model which can determine female or male based on the first English name.

1. Data

The data have two columns including name and gender. The size of data is 95025*2 . There are no duplication and null. But the name data is not clean(Fig.1).

	name	gender
0	Aaban&&	M
1	Aabha*	F
2	Aabid	M
3	Aabriella	F
4	Aada_	F

Figure 1 Some example of original data set

2. Data Pre-processing

2.1 Data cleaning

I filtered the number and symbols and keep alphabets in the name data. I also checked the gender data, there are almost two times female compare to male. Thus, the group is a bit unbalanced. If the model is bias to female, i will try to batch balance.

	name	gender
0	Aaban	M
1	Aabha	F
2	Aabid	M
3	Aabriella	F
4	Aada	F

Figure 2 Some example of cleaned data set

2.2 Encode

First, all the name will be change into lowercase. Second, the alphabet were transferred

to number. If the data is used for the Ensemble model, i also normalized them as the figure shows. For the LSTM model, i did not normalize the data because the embedding function will do.

	name	gender
0	[1.0, 1.0, 2.0, 1.0, 14.0, 0.0, 0.0, 0.0, 0.0,...	1.0
1	[1.0, 1.0, 2.0, 8.0, 1.0, 0.0, 0.0, 0.0, 0.0, ...	0.0
2	[1.0, 1.0, 2.0, 9.0, 4.0, 0.0, 0.0, 0.0, 0.0, ...	1.0
3	[1.0, 1.0, 2.0, 18.0, 9.0, 5.0, 12.0, 12.0, 1....	0.0
4	[1.0, 1.0, 4.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	0.0

Figure 3 Example of encoding data without normalization

3 Model train

3.1 LSTM model

The long short-term memory model is quite good to learn relationship between name and gender. The model spent about 30 minutes, and the accuracy of training is more than 0.92. The model will become over-fit after 30 epochs, and the accuracy of validation become hard to increase. Thus, i early stop and the accuracy of validation set is also around 0.91.

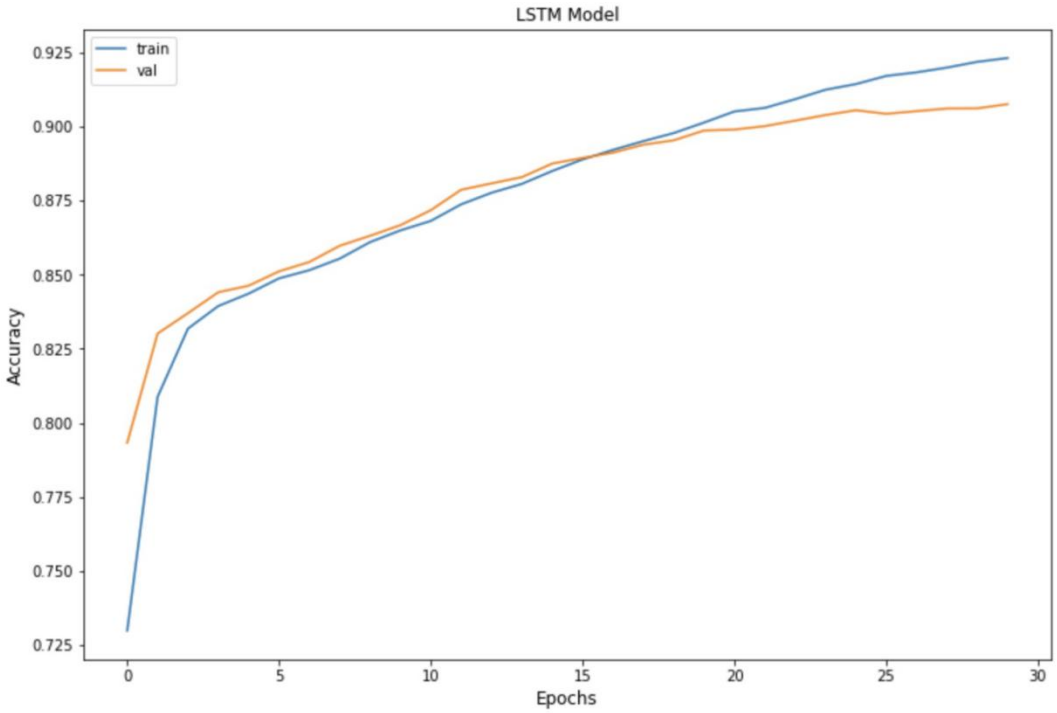

Figure 4 Accuracy of training and validation of LSTM model

3.2 Ensemble model

At first, i random run many simple models such as random forest, decision tree, svm. The result show that the xgboost, random forest and XxtraTrees are the best. Thus, i used gridsearchCV and randomSearchCV to do the hyperparameter tuning. However, the xgboost take several hours to finish the tuning. Thus, we only take two models including random forest and ExtraTree to do the ensemble. The accuracy of training of the ensemble is 0.94 which is higher than that of LSTM model. However, the accurayc of validation set is only 0.79 which is much lower than LSTM. One way to avoid the over-fitting is data augmentation in future.

	precision	recall	f1-score	support
0.0	0.80	0.89	0.84	12061
1.0	0.76	0.62	0.69	6944
accuracy			0.79	19005
macro avg	0.78	0.76	0.76	19005
weighted avg	0.79	0.79	0.79	19005

Figure 5 Accuracy of validation of Ensemble model

4 Model deployment

4.1 Inference

Based on the result, i choose the LSTM model to deploy. The model is save as gender.h5. User can run “python3 inference.py” to use the model . The output is the gender and confidence score.

	Name	F or M?	Probability
0	Joe	M	0.84
1	Biden	M	0.88
2	Kamala	F	0.98
3	Harris	M	0.97

Figure 6 Example of running the inference

4.2 Flask app

Users also can run “python3 app.py” to build a flask app and use postman to test the result. Here is one example, if users input one name Joe, the output is M (Fig.7).

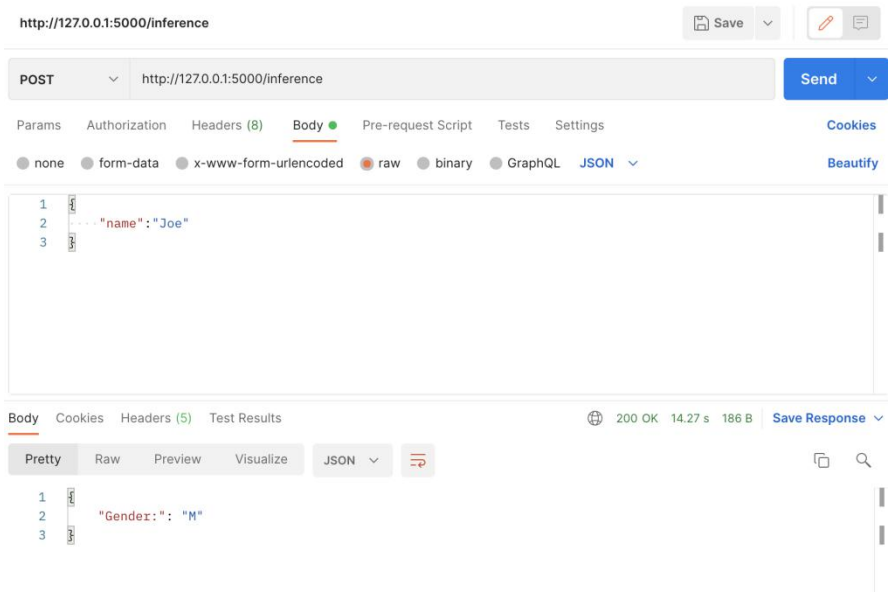

Figure 7 Example of running flask app

4.3 Dockerfile

1. build a image

```
docker build -t gender_detection:1.0 .
```

```
[➜  gender-detection git:(main) ✗ docker build -t gender_detection:1.0 .
[+] Building 176.2s (13/13) FINISHED
 => [internal] load build definition from Dockerfile                          0.0s
 => => transferring dockerfile: 37B                                           0.0s
 => [internal] load .dockerignore                                             0.0s
 => => transferring context: 2B                                               0.0s
 => [internal] load metadata for docker.io/library/python:3.8                 3.3s
 => [internal] load build context                                             0.4s
 => => transferring context: 4.88MB                                           0.4s
 => CACHED [1/8] FROM docker.io/library/python:3.8@sha256:7b72fe8ab313d9b48755f13  0.0s
 => [2/8] COPY ./train.py ./app/train.py                                      0.0s
 => [3/8] COPY ./app.py ./app/app.py                                          0.0s
 => [4/8] COPY ./inference.py ./app/inference.py                              0.0s
 => [5/8] COPY ./gender.h5 ./app/gender.h5                                    0.1s
 => [6/8] COPY ./requirements.txt ./app/requirements.txt                      0.0s
 => [7/8] WORKDIR /app/                                                       0.0s
 => [8/8] RUN pip3 install -r requirements.txt                              152.6s
 => exporting to image                                                       19.6s
 => => exporting layers                                                      19.6s
 => => writing image sha256:6705e7a0004f3ef3fffccb18128aaebf1368e05f690dba943f009  0.0s
 => => naming to docker.io/library/gender_detection:1.0                       0.0s
```

2. run the app

```
docker run -p 5000:5000 gender_detection:1.0
```

3. Test

```
curl -X POST http://127.0.0.1:5000/inference -H
'Content-Type:application/json' -d '{"name":"Joe"}'
```