Predicting Gender from Name

Gauri Dandi and Maura Keith
DS 4420
Prof. Wallace
29 April 2022

Contents

1 Abstract

The purpose of this project was to determine if gender can viably be predicted from a name itself. Such models that we built are effective in predicting male or female names, but are least effective against names that are “neutral”, that is, names that can be assigned to either gender. Simple models drawing upon manual feature engineering rely the most on the last letter of a name, whereas our neural network model drew upon the order of letters.

2 Introduction and Motivation

The inspiration for this project came from asking the question, “can we predict the gender of a name from the name itself?” The question is simple enough, yet opens a complex array of arguments and insights. We discussed several points of such a model before proceeding with our execution.

For one, gender is a very hot topic in culture today. The creation of a model that infers gender from a name and is applied to individuals would face huge backlash and chaos. For example, creating such a model to educate ad targeting would be unethical; we’d be assuming a lot about a person and their identity simply from their name. However, there are use cases that can be not only ethical but useful as well. For example, the model could be applied to a dataset of names from LinkedIn with profession and city to examine the gender gap in different fields across different areas. We deemed that such use cases of a gender prediction model would be ethical, and thus chose to proceed forward.

Our second ethical consideration was the source of our data. Initially we were tempted to use the `names-dataset` library in Python, yet learned that this data was collected from a Facebook data leak back in 2019 (The Guardian). Since data points in this dataset were not recorded with consent, we scrapped this dataset and continued our search. We found two other datasets that we deemed ethica, from Harvard University and UC Irvine, mainly drawing upon social security application data. While social security application data may sound invasive, this dataset draws from the Social Security Administration’s public data of popular data names, with their site proclaiming that “Social Security is with you from day one, which makes us the source for the most popular baby names and more!” (Social Security Administration).

An argument could be made that it would be more efficient to store a database of name and gender to query rather than to use a machine learning model. However, such a database would inevitably run into issues when it encounters a name it is not familiar with, and would get “stuck” in such a case. This is where our model comes in.

3 Experimental Setup

3.1 Data Cleaning

As priorly mentioned, we used two datasets to create our model. Our Harvard dataset consists of a list of 300,000 unique names by country with a binary gender assignment. The strength of this dataset is the

country data, because we can train the model by country or analyze the strength of our model across different countries. As shown below in figure 3.1.1, the dataset is mainly comprised of data from the US, Great Britain, and Canada, but many other nations are represented and we have a wide range of coverage. As well, there are more female names in the dataset than male, but both genders are well represented with over 10,000 instances of each (figure 3.1.2)

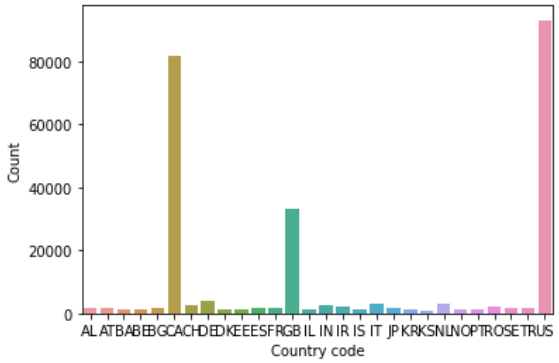


Figure 3.1.1: Country distribution of Harvard dataset

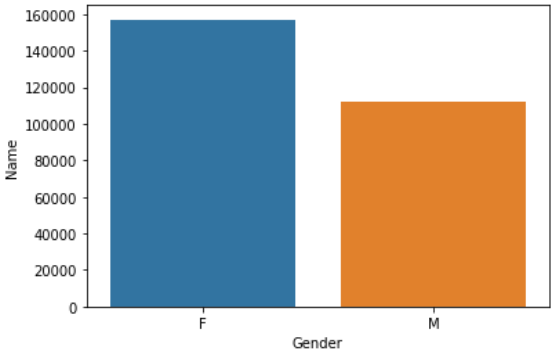


Figure 3.1.2: Gender distribution of Harvard dataset

In cleaning, we lowered all names to lowercase for consistency and filtered our data to only include names composed of English letters. What resulted was a dataframe with three columns: name, country code, and gender (figure 3.1.3).

	Name	code	gender
0	a hannan	CA	M
1	a jay	CA	M
2	a jay	GB	M
3	a k i l	CA	M
4	a lah	CA	F
...	...	...	...
290015	혜림	KR	F
290016	혜진	KR	F
290017	호	KR	M
290018	화자	KR	F
290019	훈	KR	M

269743 rows × 3 columns

Figure 3.1.3: Harvard dataset

Our second dataset comes from UC Irvine and contains 130,000 unique names. This dataset is where most of our analyses came from because it includes instance counts. For example, the name “Terry” can be male or female, so it appears twice in this dataset: once as female with the count of women named Terry and once as male with the count of men named Terry. From these counts we were able to calculate the ratio of male and female for all names. Shortcomings of this dataset are its smaller size and that it does not include international names, which is why we chose to explore using both sets in our models. An initial exploratory data analysis we discovered that there are more female names represented in this dataset (figure 3.1.4), yet more men are represented by the dataset overall because male names tend to

have higher counts (figure 3.1.5). Nonetheless, there are well over 4,000 unique male and female names each and over 1.5 million each of men and women represented, so we felt confident in moving forward using this data.

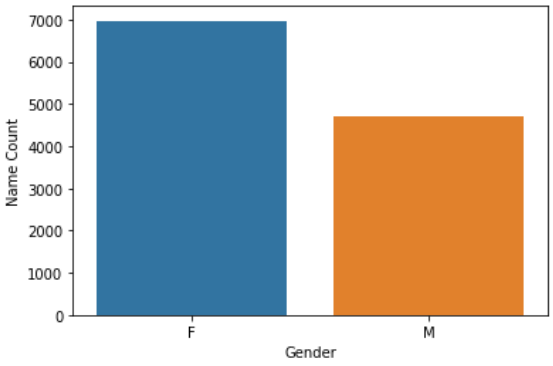


Figure 3.1.4: Gender distribution of unique names for UC Irvine dataset

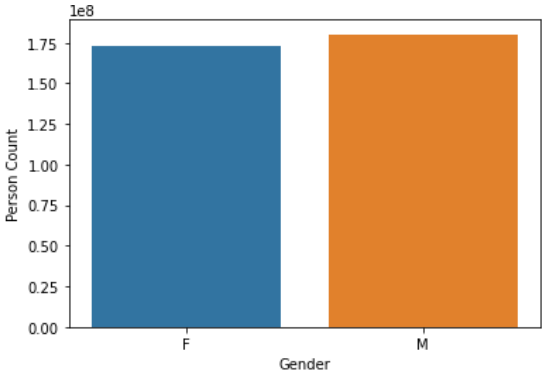


Figure 3.1.5: Gender distribution of people for UC Irvine dataset

	Name	Gender	Count
0	james	M	5304407
1	john	M	5260831
2	robert	M	4970386
3	michael	M	4579950
4	william	M	4226608
...	...	...	...
11661	binyomin	M	1001
11662	braydin	M	1001
11663	cosimo	M	1001
11664	juanita	M	1001
11665	ridley	M	1001

11666 rows × 3 columns

Figure 3.1.6: UC Irvine dataset

To clean this dataset, we filtered data for names with at least 1,000 instances so as to train on more common names and avoid small sample size biases. This filter decreased our sample size to just over 11,000 rows. We took three different approaches in cleaning this dataset, making three different output datasets:

1) Binary assignment

   For this approach, names that appear for both genders are assigned only to the gender they most often are assigned to. Since our Harvard dataset is binary, it is compatible to be fed into a model created with data in this format.

	Name	Gender	Count
0	aaden	M	4877
1	aadhya	F	1733
2	aadya	F	1133
3	aakash	M	1003
4	aaleyah	F	1202
...	...	...	...
10707	zuri	F	7895
10708	zuriel	M	1448
10709	zyaire	M	2640
10710	zyler	M	1046
10711	zyon	M	2639

10712 rows × 3 columns

Figure 3.1.7: UC Irvine dataset with binary gender assignment

2) Continuous scale

This approach created a target variable which is the ratio of female instances to all instances of a name. For example, the name Terry has a female ratio of 0.19, meaning that 19% of people named Terry are women. The purpose of this format is to create a linear regression model to see if the gender ratio of a name can be predicted.

	Name	Count	Female Ratio
0	james	5328370	0.004497
1	john	5282978	0.004192
2	robert	4990971	0.004124
3	michael	4602810	0.004967
4	william	4242917	0.003844
...	...	...	...
11656	lin	1002	1.000000
11657	siani	1002	1.000000
11658	yocelin	1002	1.000000
11659	africa	1001	1.000000
11660	evalee	1001	1.000000

10712 rows × 3 columns

Figure 3.1.8: UC Irvine dataset with continuous scale gender assignment

3) Neutral assignment

The neutral dataset assigns a neutral label to all names that are below a 95% majority one gender, that is, female ratio is less than 0.05 or above 0.95. Names past that threshold are assigned to male or female following the majority of its count. This approach is a middle

ground between binary assignment and a continuous scale, and thus is where most of our analysis took place.

	Name	Gender	Count
0	kalena	F	2370
1	ravyn	F	2204
2	jaymee	F	2163
3	rawan	F	1131
4	jaylynn	F	8412
...	...	...	...
10707	landyn	N	10711
10708	lane	N	43027
10709	reese	N	50334
10710	christen	N	14584
10711	payton	N	77931

10712 rows × 3 columns

Figure 3.1.9: UC Irvine dataset with neutral gender assignment

3.2 Feature Engineering

For our simpler models, we engineered features manually to draw upon. Since these features simply come from the names themselves, functions to engineer features worked for both the Harvard and UC Irvine datasets. These features included:

- 27 variables each representing a one-hot encoded value of the presence of a letter of the alphabet in a name or a space
- An integer to represent the first letter of a name
- An integer to represent the last letter of a name
- The length of a name
- The number of vowels in a name
- The ratio of vowels in a name

Name	Gender	Count	Length	First letter	Last letter	Number of vowels	Number of consonants	Ratio vowels	Ratio consonants	...	Contains 'q'	Contains 'r'	Contains 's'	Contains 't'	Contains 'u'	C
aaden	M	4877	5	0	13	3	2	0.600000	0.400000	...	0	0	0	0	0	
aadhya	F	1733	6	0	0	4	2	0.666667	0.333333	...	0	0	0	0	0	
aadya	F	1133	5	0	0	4	1	0.800000	0.200000	...	0	0	0	0	0	
aakash	M	1003	6	0	7	3	3	0.500000	0.500000	...	0	0	1	0	0	
aaleyah	F	1202	7	0	7	5	2	0.714286	0.285714	...	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
zuri	F	7895	4	25	8	2	2	0.500000	0.500000	...	0	1	0	0	1	
zuriel	M	1448	6	25	11	3	3	0.500000	0.500000	...	0	1	0	0	1	
zyaire	M	2640	6	25	4	4	2	0.666667	0.333333	...	0	1	0	0	0	
zyler	M	1046	5	25	17	2	3	0.400000	0.600000	...	0	1	0	0	0	
zyon	M	2639	4	25	13	2	2	0.500000	0.500000	...	0	0	0	0	0	

Figure 3.2.1: UC Irvine dataset with engineered features

3.3 Model Construction

For the binary-formatted data from UC Irvine, we constructed a straightforward logistic regression model from the `scikit-learn` library. Since this data format is the same as the Harvard dataset, we were able to feed Harvard data into this model to analyze performance by country.

A linear regression model was constructed upon the continuous scale data format, again from the `scikit-learn` library, using first all names in the dataset. The model was then reconstructed to be trained upon only names that appear for both gender, and again for names that were only below a 95% majority of one gender, to analyze how performance changes with these thresholds.

As we saw the most applications to using data formatted to be labeled as male, female, or neutral, most of our models were created from this dataset. We created logistic regression, SVC, and random forest models from `scikit-learn`, and a Bidirectional LSTM neural network from the `Keras` library. As results looked the most promising from the random forest model, this model was hyperparameter tuned using GridSearchCV and also used to conduct a feature importance analysis.

4 Results and Discussion

4.1 Logistic Regression for Binary Names

Our first and simplest model created was a logistic regression model trained on the UC Irvine dataset with binary name assignments. This model resulted in a 77% testing accuracy, with 72% precision and 69% recall (figure 4.1.1). We were pleased with these results because the tradeoff between precision and recall was well-balanced. There were roughly the same number of true females misclassified as male as there were true males misclassified as female (figure 4.1.2); however, as mentioned above there are more female names in this dataset than male names (figure 3.1.4), indicating that the model tends to perform better in correctly identifying female names.

Set	Accuracy	F1 Score	Precision	Recall	ROC AUC	Sample Size
Train	0.7827	0.7186	0.7421	0.6966	0.7682	7498
Test	0.7707	0.7015	0.7181	0.6857	0.7557	3214

Figure 4.1.1: Train and test metric results of binary logistic regression model

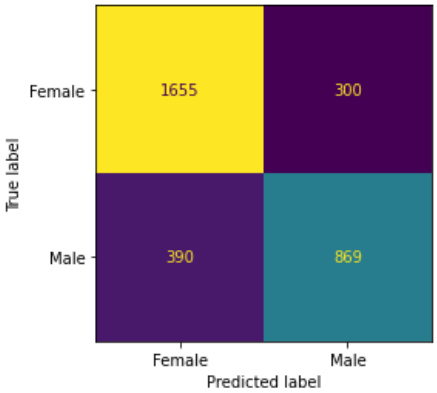


Figure 4.1.2: Test confusion matrix for binary logistic regression model

The binary logistic regression model is the only model we created that is compatible with the Harvard dataset format which binarily assigns gender to name, so Harvard data was fed into the binary logistic regression model for a country accuracy analysis. It's worth noting that since there are not counts of name occurrences in this dataset, many obscure names are in this dataset such as "Zzyzx" and names with spaces such as "A K I L", which likely caused the most of our errors in testing. As shown below in the confusion matrix (figure 4.1.3), the model is vastly more effective in correctly identifying female names than in correctly identifying male names.

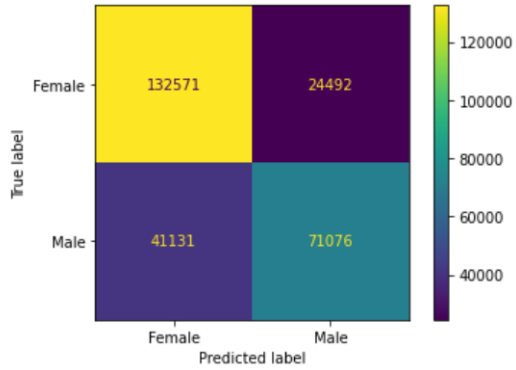


Figure 4.1.3: Confusion matrix for binary logistic regression model run on Harvard dataset

Set	Accuracy	F1 Score	Precision	Recall	ROC AUC	Sample Size
Harvard dataset	0.7563	0.6842	0.7437	0.6334	0.7387	269270

Figure 4.1.4: Train and test metric results of binary logistic regression model

As shown in our country analysis below in figure 4.1.5, of the 45 countries with at least 100 names in the dataset the United States shockingly ranked 24th. Top ranking countries include Latvia, Lithuania, and Portugal. Upon inspection of names in these countries, most end in a vowel. Our model is likely more effective against these sets because the 'rules' of gendering a name more directly follow a pattern of female names ending with an 'a'. This pattern is also consistent with the model's stronger performance overall for female names than male names in this dataset. Countries the model is least effective for are Turkey, India, and Japan, all of which have names that are generally structured very differently than the Latin-based names most commonly found in the US-based dataset from UC Irvine. Most notably,

accuracy was 49.7% for Japan, meaning that if the model predicted the opposite of its actual estimate, it would do a better job in gender prediction than it currently does.

Rank	Country	Accuracy (%)	Sample Size
1	Latvia	97.8	689
2	Lithuania	90.1	849
3	Portugal	89.0	1034
…	…	…	…
22	Belgium	77.9	1477
23	United States	77.8	93,028
24	Belarus	76.3	511
…	…	…	…
43	Turkey	62.5	1763
44	India	56.0	2690
45	Japan	49.7	1586

Figure 4.1.5: Binary logistic regression model performance by country

4.2 Linear Regression for Continuous Scale Names

A linear regression model was implemented upon the continuous scale names data format. The initial model has an R^2 value of 0.35, but upon graphing these results it was revealed that this is because the majority of names are a 0 or a 1 on the continuous scale (figure 4.2.1). Therefore, a reasonable R^2 results from the bimodal nature of the data. Therefore the model was reconstructed excluding names that are a 0 or 1 on the scale, that is, entirely male or entirely female names, and R^2 dropped to 0.09 (figure 4.2.2). The model was reconstructed once more with names that have less than a 95% majority of one name, assigned to 'neutral' in our other data format, and R^2 became -0.06 (figure 4.2.3). Since this model performed very poorly, especially upon names that are not on the edges of our scale, it was not explored further.

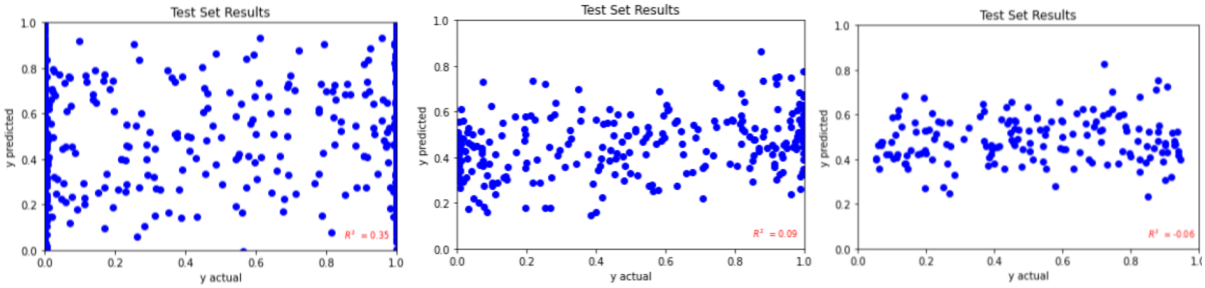


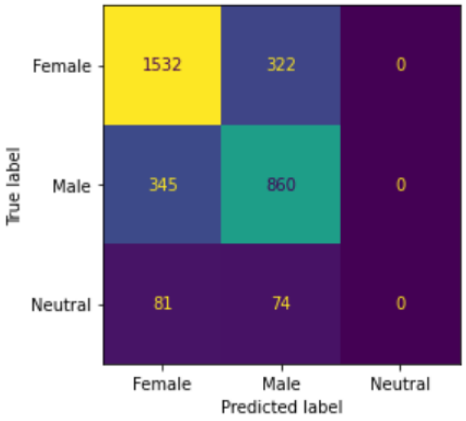
Figure 4.2.1: Linear regression model constructed from all names
Figure 4.2.2: Linear regression model constructed from names with scale values above 0 and below 1
Figure 4.2.3: Linear regression model constructed from names with scale values above 0.05 and below 0.95

4.3 Logistic Regression and SVC for Neutral Formatted Names

Both our logistic regression and SVC models performed similarly, with 75% and 74% testing accuracy for each, respectively (figure 4.3.1). These results seem promising, until digging into results via confusion matrices (figures 4.3.2 and 4.3.3). Both models predict male and female names fairly well, but never predict the neutral label. The binary logistic regression model had 77% accuracy (figure 4.1.1), and these models stand at 74% and 75% with just under 5% of names being neutral; therefore performance is nearly the same as the binary logistic regression model for male and female names, with overall accuracy lower due to the consistent miscategorization of neutral names.

Model	Set	Accuracy
Logistic Regression	Train	0.7519
Logistic Regression	Test	0.7508
SVC	Train	0.7522
SVC	Test	0.7442

Figure 4.3.1: Logistic regression and SVC results



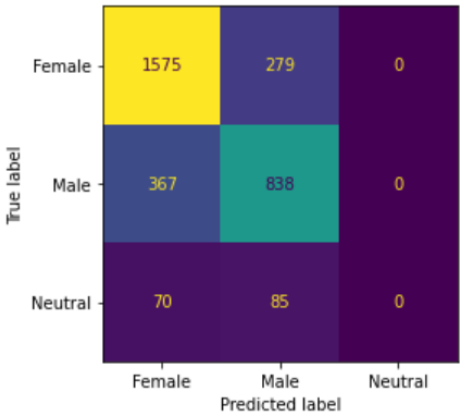


Figure 4.3.1: Logistic regression trained with neutral names testing confusion matrix
Figure 4.3.2: SVC trained with neutral names testing confusion matrix

4.4 Random Forest for Neutral Formatted Names

The random forest model trained without any hyperparameter tuning quickly revealed itself to deliver more accurate results, standing at 83% accuracy. Furthermore, it did sometimes predict names to be neutral, which addresses the largest issues with logistic regression and SVC upon this dataset. Overfitting was quickly revealed to be an issue, as the training set encountered 99.5% accuracy.

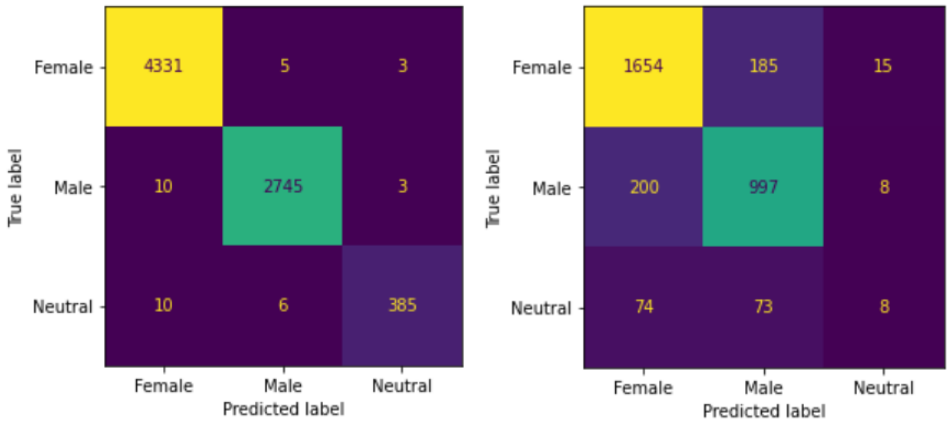


Figure 4.4.1: Random forest model trained with neutral names training confusion matrix
Figure 4.4.2: Random forest model trained with neutral names testing confusion matrix

We proceeded forward with hyperparameter tuning using GridSearchCV, but only found marginal improvement from 82.7% to 83.4% accuracy from the four-hour runtime. The tuned parameters are as follows:

Parameter	Value
Bootstrap	False
Max depth	40
Max features	Sqrt
Min samples per leaf	1
Min samples split	5
Number of estimators	1000

Figure 4.4.3: Tuned hyperparameters for random forest model

Overfitting still remains an issue, even post-tuning, as the training set accuracy is 98.6%. This is a marginal improvement against the over-99% accuracy pre-tuning, but there is still a large gap between training and testing accuracy. Furthermore, hyperparameter tuning did not increase performance on true neutral names, which was the largest point of concern.

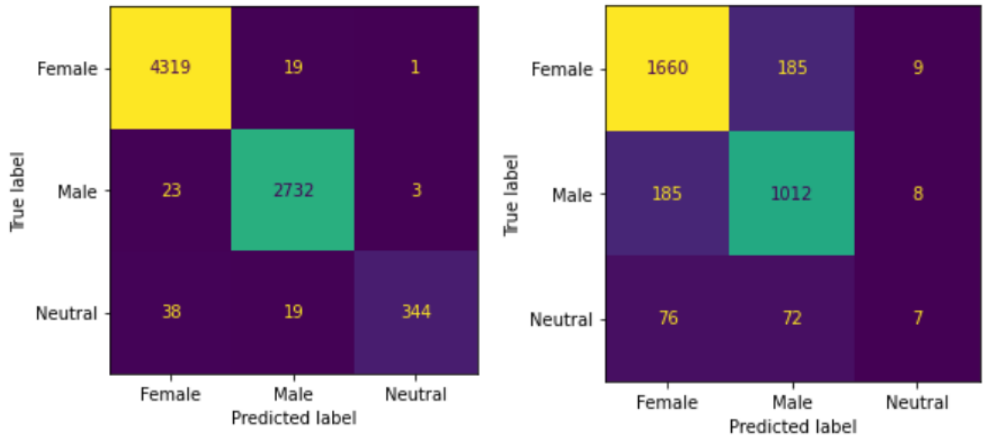


Figure 4.4.4: Random forest post-tuning trained with neutral names training confusion matrix
Figure 4.4.5: Random forest post-tuning trained with neutral names testing confusion matrix

To examine our random forest results more intuitively, we pulled the top three most common names from each category in our confusion matrix. Male names miscategorized as female often end in a ‘y’, neutral names categorized as female - Kelly, Jamie, and Dana - can be argued that they intuitively sound female, and the same for neutral names categorized as male.

		Predicted		
		Male	**Female**	**Neutral**
Actual	**Male**	James, John, Michael	Timothy, Peter, Billy	Dylan, Evan, Gage
	Female	Ruth, Rose, Doris	Elizabeth, Linda, Ashley	Ona, Jailyn, Kenyetta
	Neutral	Terry, Logan, Robin	Kelly, Jamie, Dana	Kamryn, Reece, Jaelyn

Figure 4.4.6: Top names from confusion matrix of random forest regressor

4.5 LSTM for Neutral Formatted Names

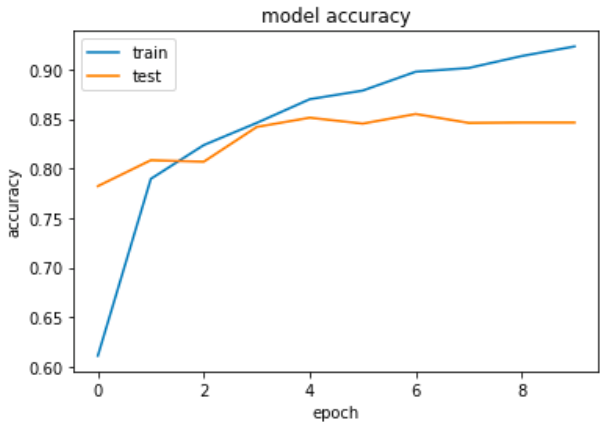
When approaching the LSTM model, we had to keep in mind that this type of model takes “time” into account, meaning our input data needed to represent an enriched version of each name. In order to satisfy this constraint, we used a 1-Dimensional Convolutional Neural Network (1-D CNN). The idea behind using a 1-D CNN for character embedding is that we were able to create a sequence representation of each name, allowing for the LSTM model to learn more intricate patterns in each name in the training set and apply it to the validation dataset for new names it is not familiar with.

We ended up with a Neural Network that integrates the 1-D CNN character embedding with the LSTM layers in one model. The architecture consists of 2 1-D CNN layers with ReLU activations, 2 Bidirectional LSTM layers, and one Dropout layer. After running for 10 epochs with a batch size of 200, the model had a 92.3% train accuracy and a 84.7% validation accuracy. While there is definitely room for

improvement in this model, it is able to predict neutral genders with slightly better accuracy than the Random Forest model, showing that the character embedding does enrich the prediction method.

```
Layer (type)                 Output Shape              Param #
=================================================================
embedding_3 (Embedding)      (None, 100, 28)           812

conv1d_6 (Conv1D)            (None, 94, 256)           50432

activation_10 (Activation)   (None, 94, 256)           0

conv1d_7 (Conv1D)            (None, 92, 256)           196864

activation_11 (Activation)   (None, 92, 256)           0

bidirectional_6 (Bidirectio  (None, 92, 512)           1050624
nal)

activation_12 (Activation)   (None, 92, 512)           0

bidirectional_7 (Bidirectio  (None, 512)               1574912
nal)

dense_6 (Dense)              (None, 128)               65664

dropout_4 (Dropout)          (None, 128)               0

dense_7 (Dense)              (None, 3)                 387

=================================================================
Total params: 2,939,695
Trainable params: 2,939,695
Non-trainable params: 0
```

Figure 4.5.1: LSTM Neural Network architecture with parameters



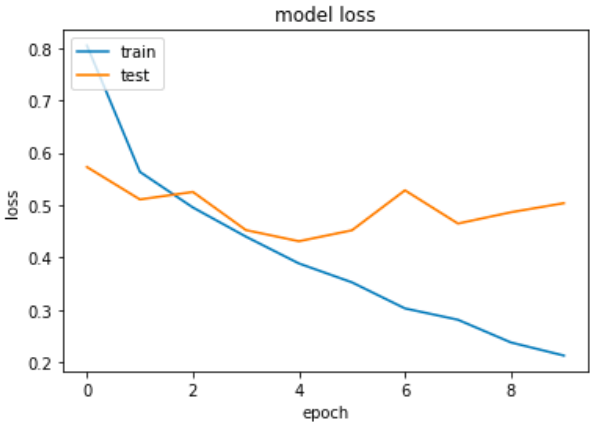


Figure 4.5.2: LSTM model accuracy over 10 epochs
Figure 4.5.3: LSTM model loss over 10 epochs

5 Conclusions and Future Work

In conclusion, we found that supervised learning models such as Random Forest and Logistic Regression drew most heavily on the last letter of a name when trying to predict the gender of a name. Moreover, the neural network we trained uses the sequence of characters in the training set to try to rebuild what a name might look like for a specific gender: Male, Female, or Neutral. This type of learning is viable with ML, however there are exceptions when coming to this conclusion. For example, our models are most accurate for Latin-based names, as a majority of our data does consist of these types of names. Additionally, our models are more accurate in predicting female names, due to the fact that they typically end in a vowel and generally contain more vowels than male or neutral names.

In the future, these models might benefit from having a more tuned neutral name threshold. For this project, we used a threshold of 95% to determine if a name that showed up as both Male and Female in our dataset should be classified as Neutral. However, in order to more accurately predict Neutral names, a higher or lower threshold might be useful. Additionally, we would look more into hyperparameter tuning for all of our models. For Random Forest specifically, this process took over 4 hours to complete and resulted in high overfitting with a poor prediction accuracy for Neutral names. Finally, we would really like to apply our models to a real-life problem, such as observing the gender gaps in different fields such as Computer Science and Medicine using LinkedIn data.

6 Works Cited

Associated Press in New York. “Facebook Data Leak: Details from 533 Million Users Found on Website for Hackers.” The Guardian, Guardian News and Media, 5 Apr. 2021, https://www.theguardian.com/technology/2021/apr/03/500-million-facebook-users-website-hackers.

“Baby Names.” Social Security Administration, Social Security Administration, https://www.ssa.gov/OACT/babynames/.

“Gender by Name Data Set.” UCI Machine Learning Repository, UC Irvine, 15 Mar. 2020, https://archive.ics.uci.edu/ml/datasets/Gender+by+Name.

Julio, Raffio, and Lax-Martinez Gema. “World Gender Name Dictionary.” Harvard Dataverse, Harvard University, 2018, https://dataverse.harvard.edu/file.xhtml?persistentId=doi%3A10.7910%2FDVN%2FYPRQH8%2FSO6SXA&version=1.1.