BU Department of Computer Science bu.edu/cs

Digaai: Name-Ethnicity Classification

Yuxi Jiang, Shahin Roozkhosh, Zhiwei Tang, Lucy Zhan

Introduction

Figure 1. Diagaai Logo

Digaai is a digital platform that documents the Brazilian diaspora through media (eg. text, audio, video and images). In addition, the platform maps the Brazilian diaspora.

The purpose of our project is to assist Digaai in locating and classifying locations in the US made of the Brazilian immigrant population. We do so by detecting if a given name is Brazilian.

Id	Firstname	Lastname	IsBrazilian
0	Ithalohfonseca	Chaverinho	1
1	Gustavo	Gon\xc3\xa7alves	1
2	Rafael	Geraldo	1
3	Cristyan	Victor	1
:	:	:	:

Figure 2. Training Dataset

Id	Firstname	Lastname	IsBrazilian
0	Anton	Prasetyo	-1
1	Muhamad	Riyandi	-1
2	Jefersonhugo	Ribeiro	-1
3	Allan	Victor	-1
:	:	:	:

Figure 3. Test Dataset

Our dataset consists of two sets: the train and test set. The train set has 48K values with parameters ID, firstName, lastName, and isBrazilian. Then we have a test set of size 11.9K values with parameters ID, firstName, lastName, and isBrazilian. The isBrazilian parameter currently holds -1 and is the value we must test and predict.

Classification Approach

Our input is a personal name of the structure $X = [x^0, x^1, ..., x^{n-1}]$, where x^i indicates the i-th character of a personal name of length n. Note that x^i can be any n-gram character, such as a unigram or bigram. For our input, all characters are unigram.

Name nationality classification task:

$$\mathbf{P(y^i \mid X) = D(g(X), \theta)} ,$$

where y^i is i-th ethnicity label and $D(\cdot)$ represents any kind of discriminative model with 2 parameters: trainable parameter θ and function $g(\cdot)$ for character level feature extraction [2]. We implement and test three different ML-based discriminative D(.) functions: logistic regression, KNN, and LSTM.

References

1) R. Duda, P. Hart, and D. Stork. *Wiley, New York, 2 edition, (2001)*
2) Cohen, W., Ravikumar, P. and Fienberg, S., 2003, August. A comparison of string metrics for matching names and records. In Kdd workshop on data cleaning and object consolidation (Vol. 3, pp. 73-78).
3) Lee, J., Kim, H., Ko, M., Choi, D., Choi, J. and Kang, J., 2017, August. Name nationality classification with recurrent neural networks. In Proceedings of the 26th IJCAI International Joint Conference on Artificial Intelligence.

Logistic Regression

For each name input, we separate the first and last name. Then, implement the logistic regression for each part and compare the results to draw a conclusion.

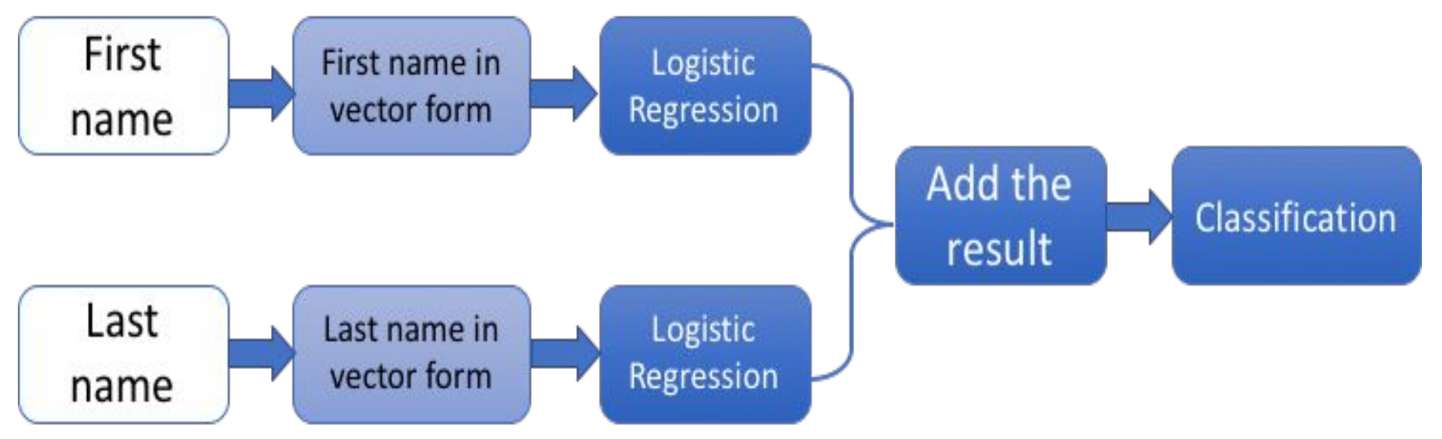

Figure 4. Logistic Regression Implementation

In our project, we transfer the names to character vectors using word2Vec and classify them using logistic regression. After we transfer the first names into vectors, we showcase it with visualization. Using classical projection methods, we reduce the high-dimensional word vectors to two-dimensional plots.

Logistic regression Results:

Figure 5 shows it is easy to distinguish the name-ethnicity. However, it seems we only get good results with a small dataset. As the dataset grows bigger, the distinguishment is not as clear (refer to figure 6). The **accuracy is 52%.**

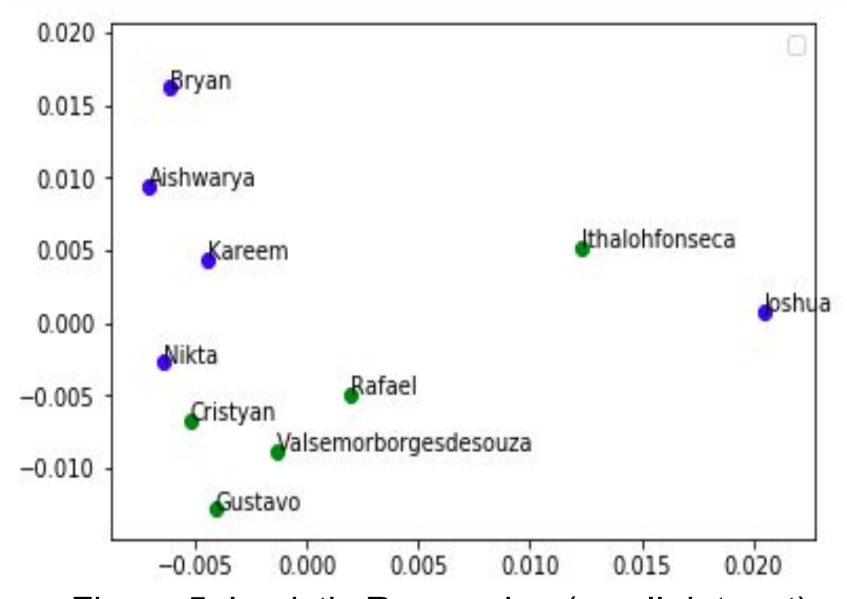

Figure 5. Logistic Regression (small dataset)

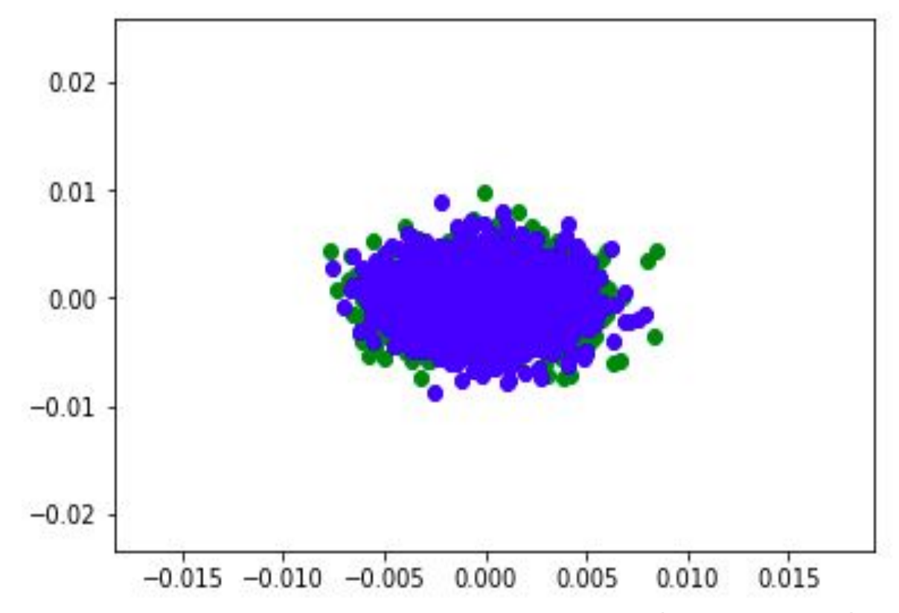

Figure 6. Logistic Regression (big dataset)

KNN

KNN (K nearest neighbor) is a supervised learning method that classifies a test point by measuring the distance between it and the training data points. It is implemented by selecting a group of k points with the shortest distance. The classification of this test point is decided by the label of the majority of the training points in the group.

The advantage of using K-NN is that there is no need to find features of the strings, which makes it easier to implement. The only cost is having a high computation cost because we need to calculate the distance each time we do the classification.

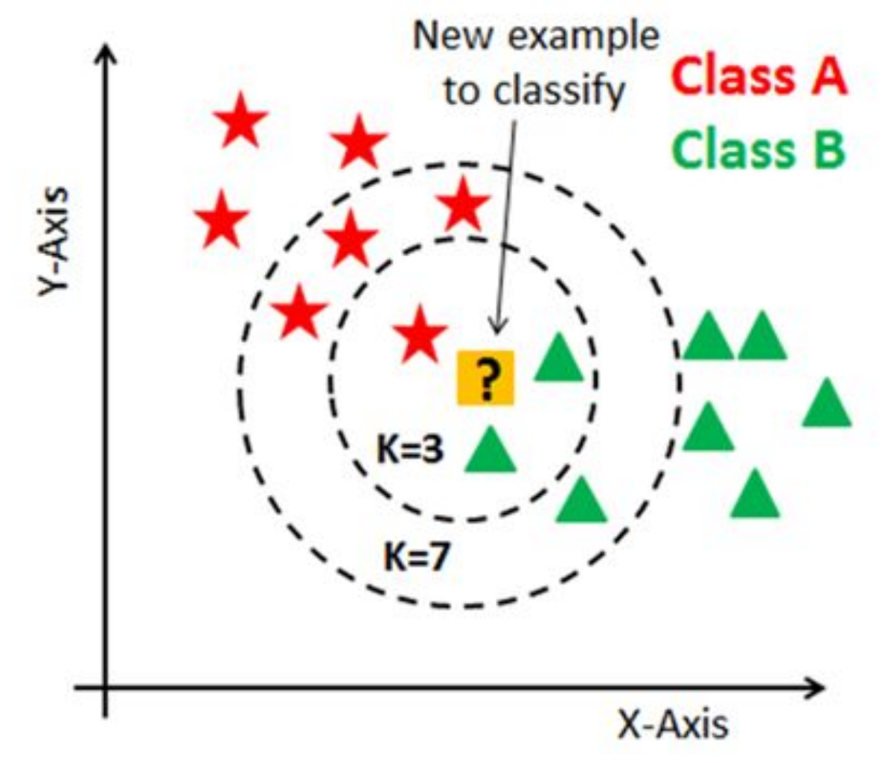

Figure 7. KNN explanation

Distance measuring method:

Jaro–Winkler distance that is a string metric measuring an edit distance between two sequences.[1]

Method for choosing k:

We examine different k values around the root of the size of training set [2].

KNN Results:

The k value we finally chose is 100 and it gives an **accuracy of 82.19%.**

RNN and LSTM

Traditional neural networks assume that all inputs are independent of each other. **RNN** processes sequential data. In our project, we use RNN because our character inputs are dependent upon each other since it forms one sequential name.

Vanilla RNNs have difficulties learning long-term dependencies due to the **vanishing/exploding gradient** problem. LSTM solves this problem.

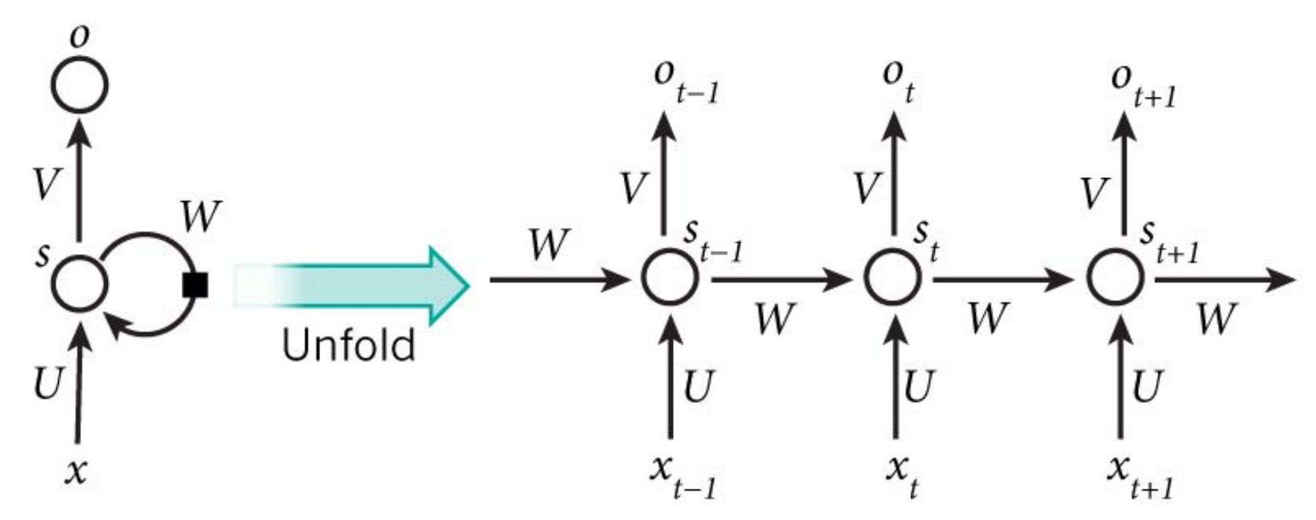

Figure 8. LSTM Implementation

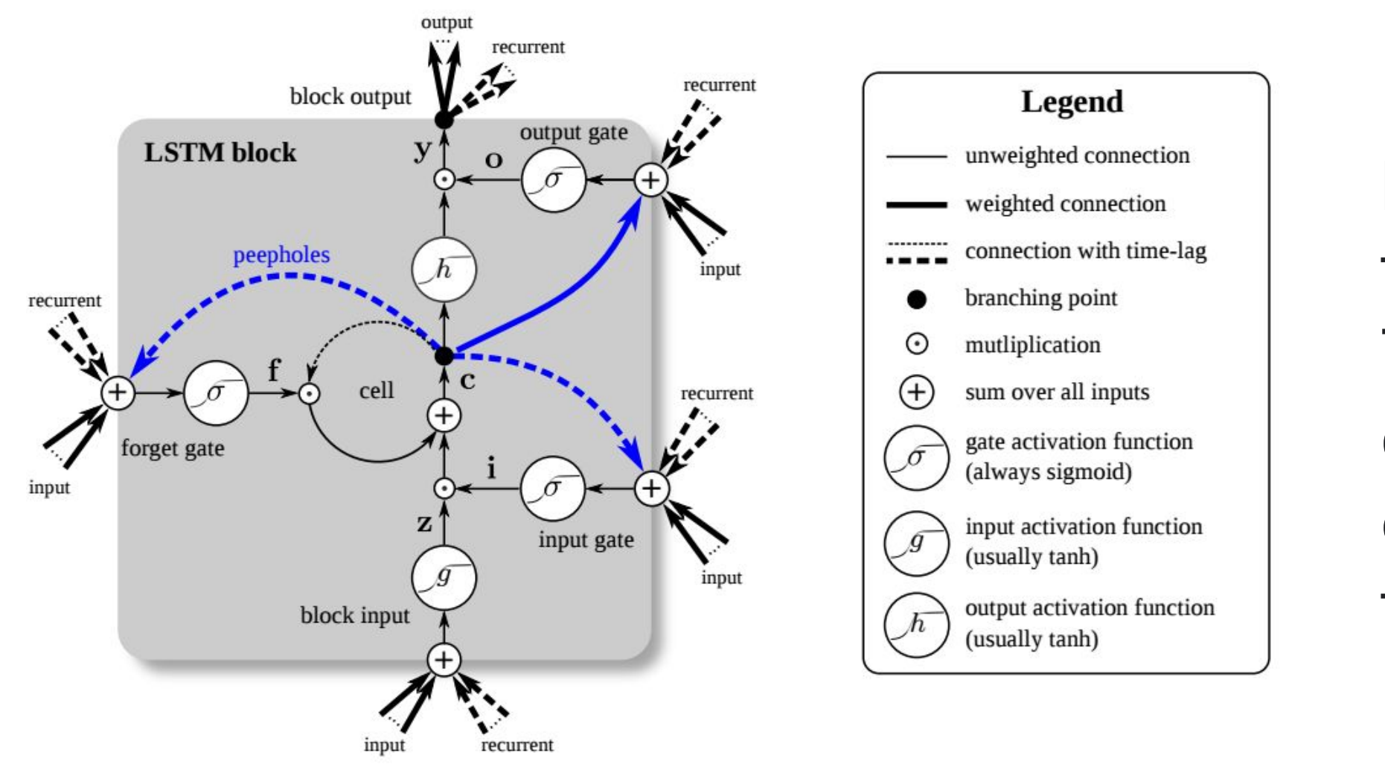

Figure 9. Inside LSTM cell

LSTMs use a different function to compute the hidden state. The memory in LSTMs are called cells and you can think of them as black boxes that take in multiple inputs: the previous state and current input.

Internally, these cells decide what to keep in and what to erase from memory. The cells then combine the previous state, the current memory, and the input. These cells are very good at maintaining long-term dependencies.

For our project, we implement LSTM using keras layers. Our input is the same character vectors.

LSTM-RNN Results:

We get an **accuracy of 90.26%**.

Conclusion

Logistic regression is not a good model in name classification as the characters of the name are hard to detect. However, KNN and LSTM get better results. KNN reaches an accuracy of 82.19%. Whereas LSTM results in a high accuracy of 90.26%. LSTM is the best implementation method for this project.

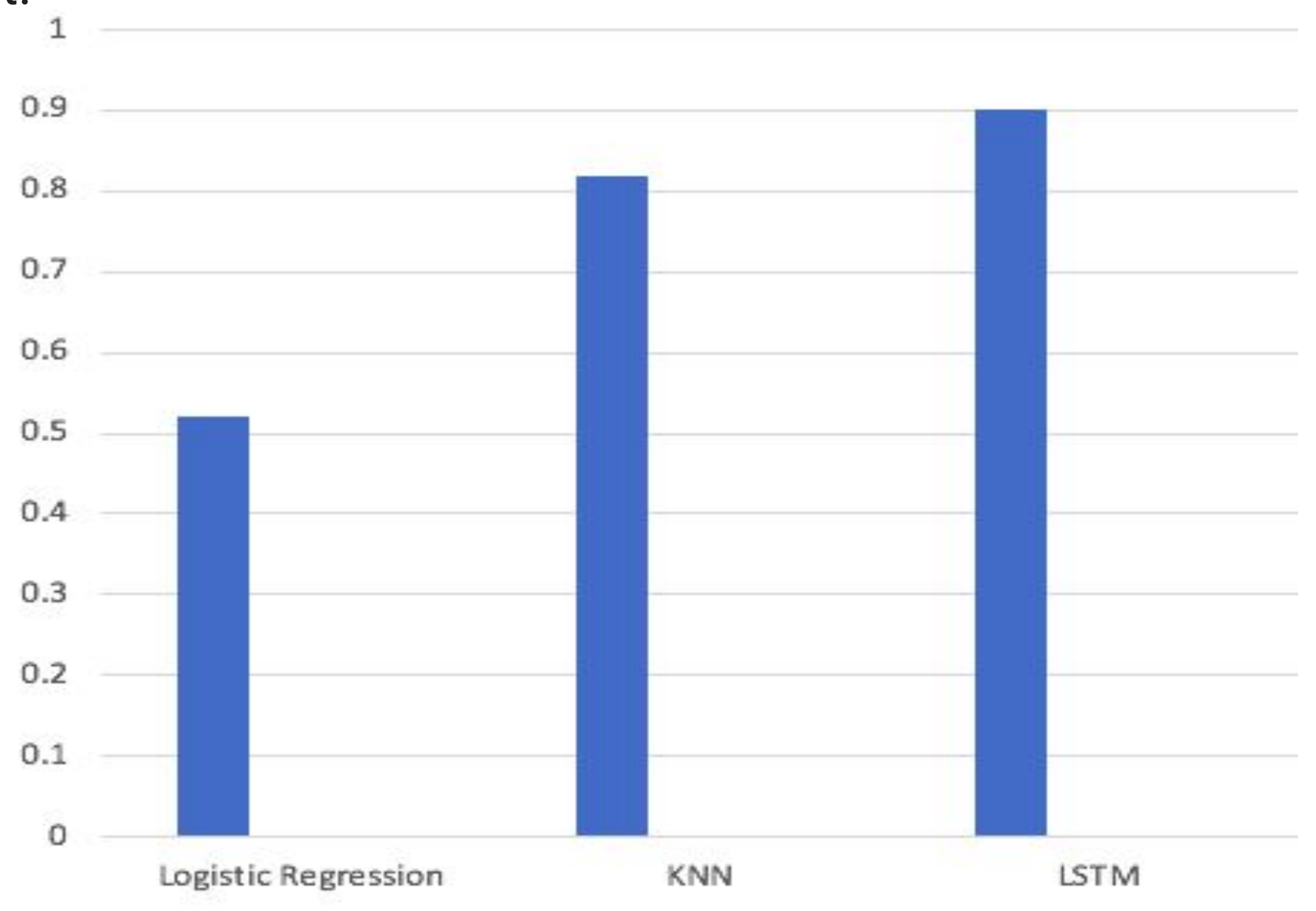

Figure 10. Different Method Accuracy Percentage Levels