CS542 Final Project Presentation

CLASSIFICATION OF BRAZILIAN IMMIGRANTS IN THE U.S.A

Group 1: Federico Siano, Yafei Guo, Runze Liu and Yu Zhao

I. Task

The project is aimed at **identifying immigrants from Brazil in the U.S.A.** using a training set of **full names as distinguishing features**. The task is not trivial for two main reasons. First, (i) immigrants can be second generation and be characterized by an American first name, last name or both. Second, (ii) the Brazilian language derives from the Portuguese colonization and is therefore inherently challenging to distinguish a Brazilian from a Portuguese name.

II. Dataset and Evaluation Metric

(1) **Training set**: 48,014 full names, with 23,965 Brazilians
(2) **Test set**: 11,942 full names (unlabeled)
(3) **Effectiveness Metric**: area under the **ROC curve** (Kaggle)

III. Approach

We implement **multiple classification methodologies** and then down-select the algorithm with the best balance of computational **intensiveness** and **accuracy**. We use: (i) **Logistic Regression/Classification**, (ii) **K-nearest neighbor (KNN)**, (iii) **Cosine Similarity**, (iv) **Character-based Recurrent Neural Network (RNN)**.

IV. Results

Logistic Classification – Federico Siano

Process:

(1) Each name in both the training and test sets are split into **syllables** using the Brazilian **hyphenator** in **Python NLTK**

(2) Each unique syllable is used as a training feature of the Logistic model

(3) The design matrix comprises a number of entries equal to the number of dichotomous variables that are built from unique syllables

Initial Cross-Validation Results (in-sample analysis):

Table 1. Logistic Classification Cross-Validation Results

Average In-Sample Results				
Label	Precision	Recall	F-1 Score	Support
0	0.96	0.84	0.91	5,000
1	0.87	0.85	0.89	5,000

Out-of-sample analysis (test data):

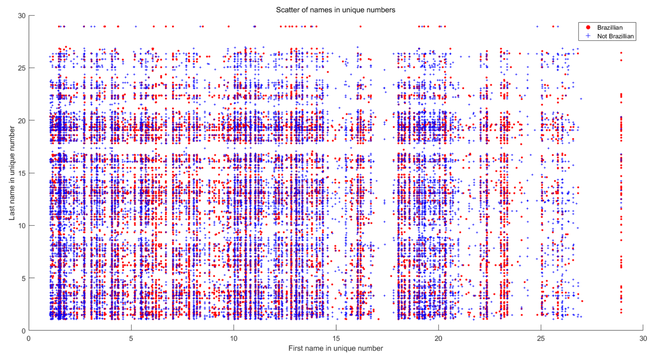
(1) Only first names → AUC: **0.87**, time: **5 min**.
(2) Only last names → AUC: **0.84**, time: **5 min**.
(3) Brazilian-if-or → AUC: **0.86**, time: **5 min**.
(4) Brazilian-if-and → AUC: **0.84**, time: **5 min**.
(5) Concatenation of first and last → AUC: **0.89**, time: **6 min**.

KNN– Yu Zhao

Process:

(1) All special ASCII characters are discarded (only English letters)
(2) Modified names are converted to 27-base system numbers
(3) Example: for the name **Prasetyo** we have $16*26^4 + 18*26^3 + 1*26^2 + 19*26 + 5 + 20*26^{-1} + 25*26^{-2} + 15*26^{-3} = 7.63*10^6$
(4) A 2-dimension feature space is created using first and last names
(5) A new name is classified as Brazilian based on the proximity to a name that was previously classified as such
(6) The model is separately trained for first and last names

Spatial representation:



Out-of-sample analysis (test data)

Both first names and last names → AUC: **0.83**, time: **5 sec**.

Cosine Similarity – Yafei Guo

Process:

(1) Features are comparable to the ones extracted for Logistic Classification
(2) The distance among vectors of features is calculated and minimized for each name in order to classify test data
(3) K=2 for text clustering

In-sample analysis

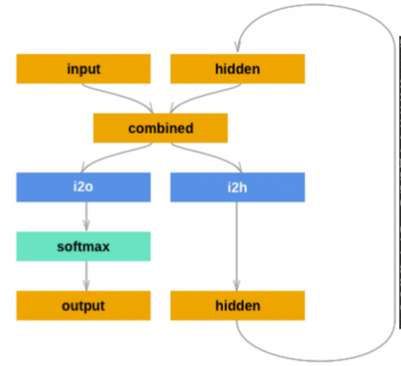
(1) Obtained results are unsatisfactory and comparable with a by-chance choice of labels
(2) Out-of-sample analysis is not performed

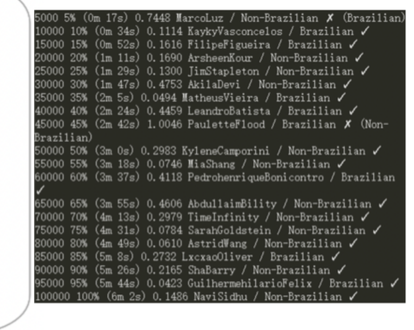
Recurrent Neural Network – Runze Liu

Process:

(1) First names and last names are concatenated and saved in two files based on whether they come from training or test set
(2) Names are converted to **tensors** using **one-hot vectors** which are subsequently joined in a **mxn matrix**
(3) The number of **neurons** is set to **128** and the **learning rate** to **0.005**.
(4) The total number of **iterations** used equals **100,000**

A schematic representation of RNN architecture and output:





Out-of-sample analysis (test data)

Both first names and last names → AUC: **0.80**, time: **10 min**.

Fine-tuning of hyperparameters

The number of hidden layers, the learning rate and the number of iterations are changed based on a simulation approach but the accuracy is not improved

V. Summary

Down-selection

The best classification algorithm in our analysis is the **Logistic Regression** that yields an AUC of **0.89** on test data for a negligible computational time of **6 min**.

VI. References

[1] Berry, M. W. Survey of text mining: clustering, classification, and retrieval. New York: Springer-Verlag, 2004.

[2] Yong, Z., Youwen, L., and Shixiong, X. An improved KNN text classification algorithm based on Clustering. Journal of Computers, 4(3): 230-237, 2009.

[3] Pytorch Tutorial on Char-RNN: https://pytorch.org/tutorials

[4] Github link: https://github.com/YuZ1225/CS542-Project