# Wells Fargo Campus Analytical Challenge

Antonio Alonso*
*Department of Electrical and Computer Engineering
Virginia Tech, Blacksburg, Virginia 24060
Email: antonioa19@vt.edu

## I. Abstract

TABLE I
Dataset Breakdown

| | |
|---|---|
| Retail Trade | 13500 |
| Entertainment | 11255 |
| Trade, Professional and Personal Services | 5275 |
| Health and Community Services | 4157 |
| Services to Transport | 2317 |
| Travel | 1489 |
| Property and Business Services | 1095 |
| Education | 445 |
| Communication Services | 282 |
| Finance | 185 |

### A. Section A: Load Data

The training and testing datasets were loaded into GoogleColab from their Excel Sheets as well as loading in the pre-trained word vectors.

### B. Section B: Preprocessing

This project focused on classifying 10 different categories of purchases that a typical person would do. The dataset contained 12 different features containing specifics about each purchase. The dataset initially had a large class imbalance. Classes that were too large were downsampled and classes with few sampled were oversampled with SOMTENC which is a technique to synthetically create new data for imbalanced datasets. Feature engineering was done to clean the features and convert the categorical to a numerical representation. The numerical features were normalized with min-max normalization because the distribution was even and in a range of [0, 1] for the model to interpret. The categorical features were either tokenized and then mapped to GloVe pre-trained word vectors for quick and efficient computation and recourse constraints of length 100. The other categorical features if the unique values were small were one-hot encoded as new features. Lastly, the CDF-SEQ ID feature was encoded using a Hashing function because it doesn't provide much information and would be better used as a hash to interpret the difference between two IDs and doesn't contain any words strictly numbers.

### C. Assumptions

All null values in categroical features were replaces with a "ukn" token and all null values in numerical features were replaced with 0. This was done so that there was a uniform baseline for the model to interpret. I assumed that the transaction description would provide the most insight into if the model to predict the transaction type.

*1) Section B2: Dataset Normalization:* This dataset does not have an equal label distribution. Therefore batch normalization was used to equally distribute the classes. The classes over 3000 were downsampled and the classes below were oversampled with SMOTENC which is a technique to synthetically create new data for imbalanced datasets. After normalization, each of the labels has 3000 samples.

### D. Section E1: Model Architecture

This model takes in six inputs: five of the categorical feature mapping and one data frame with twelve features. First, the categorical features are passed through embedding layers and then passed through a 1D Convolutional layer. After filtering, the output is passed through a GlobalMaxPooling layer and then concatenated with each feature. Then the transaction description feature is then passed through multiple dense layers and concatenated with the output of the rest of the categorical features because the description contains the most information. After one dense layer, it is then concatenated with the input data frame which contains the numerical and one-hot encoded features. After concatenation, it is all passed through multiple dense layers for classification. Multiple dropout layers were incorporated to reduce over-fitting as well as using batch normalization. The last layer is a dense layer has an output of 10 neurons for the ten classes needed for classification. This was done because all of the inputs were of different sizes and could not be concatenated together initially. The numerical and one-hot encoded features served as one input data frame with fifteen different features in total. The embedded categorical features used convolution so that the model could be able to develop its own features for inference. The transaction ID was hashed because it provided the least amount of information.

### E. Section E2: Training and Evaluation

Model training was trained for 85 epochs with a validation split of 0.3 with a final training accuracy of .92. The model began to over fit to the dataset because of the few amount of samples for each class. This model was able to generate labels for the testing set

### F. Limitations

This dataset was very imbalanced had very few samples for multiple classes such as Finance. Since synthetic data had to be generated it is not as strong as the classes with

a full set of examples. With more real data performance would greatly increase. After analysis, the classes that did not need generated data were classified much better than classes that did. Additionally, classifying ten different classes is very difficult on such a small dataset. Another Limitation was that this entire project was run on Google Colab which has limits on RAM and its computation time. With more resource allocation this project would have been able to generate more samples and improve its performance drastically.

### *G. Real World Application*

This model could be used by banks such as Wells Fargo to help classify transaction genres and help customers know what their spending their money on. As well as identify potential fraudulent purchases if a customer doesn't usually purchase something from a particular genre.

## REFERENCES

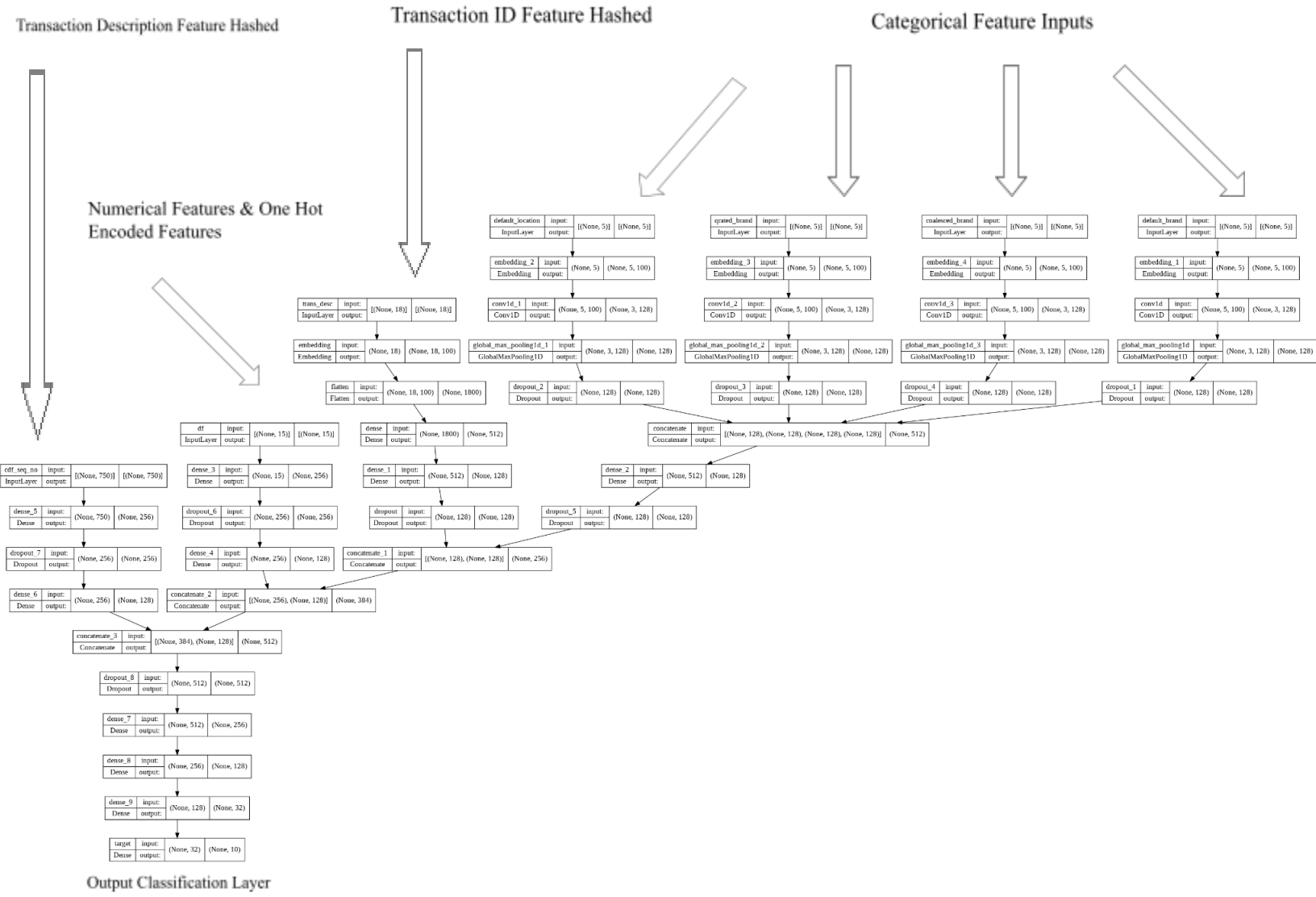

[1] J. Pennington, "GloVe: Global Vectors for Word Representation," Glove: Global vectors for word representation. [Online]. Available: https://nlp.stanford.edu/projects/glove/. [Accessed: 09-Jul-2022].

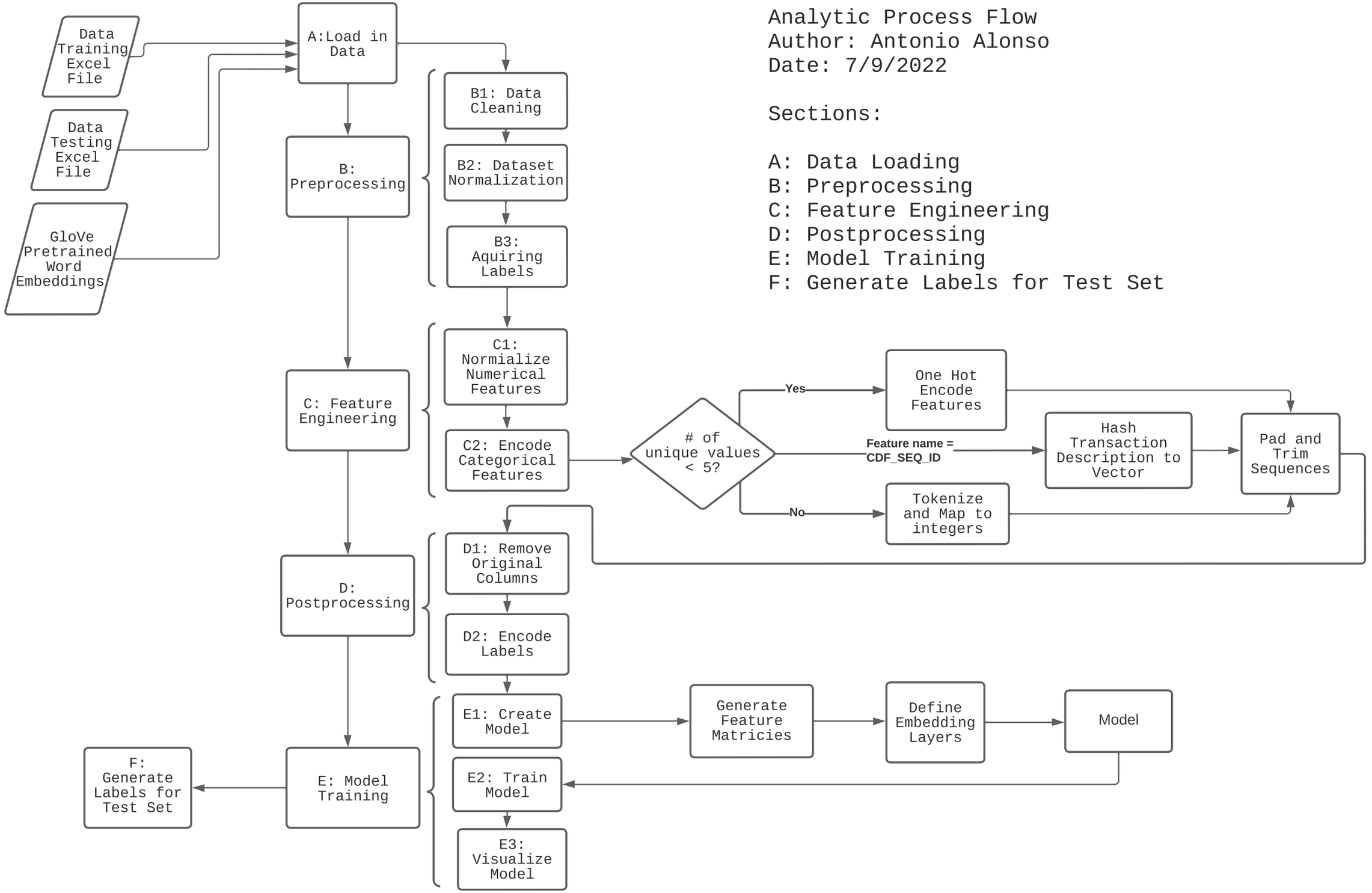
[2] C. Guo and F. Berkhahn, "Entity embeddings of categorical variables," arXiv.org, 22-Apr-2016. [Online]. Available: http://arxiv.org/abs/1604.06737. [Accessed: 09-Jul-2022].

[3] "The functional API Tensorflow Core," TensorFlow. [Online]. Available: https://www.tensorflow.org/guide/keras/functional. [Accessed: 09-Jul-2022].

[4] J. Brownlee, "How to use word embedding layers for deep learning with keras," Machine Learning Mastery, 01-Feb-2021. [Online]. Available: https://machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras/. [Accessed: 09-Jul-2022].

# Model Architecture




Analytic Process Flow
Author: Antonio Alonso
Date: 7/9/2022
Sections:
A: Data Loading
B: Preprocessing
C: Feature Engineering
D: Postprocessing
E: Model Training
F: Generate Labels for Test Set
Data Training Excel File
Data Testing Excel File
GloVe Pretrained Word Embeddings
A:Load in Data
B: Preprocessing
B1: Data Cleaning
B2: Dataset Normalization
B3: Aquiring Labels
C: Feature Engineering
C1: Normialize Numerical Features
C2: Encode Categorical Features
# of unique values < 5?
Yes
One Hot Encode Features
Feature name = CDF_SEQ_ID
Hash Transaction Description to Vector
No
Tokenize and Map to integers
Pad and Trim Sequences
D: Postprocessing
D1: Remove Original Columns
D2: Encode Labels
E: Model Training
E1: Create Model
Generate Feature Matricies
Define Embedding Layers
Model
E2: Train Model
E3: Visualize Model
F: Generate Labels for Test Set