ResumAI

Brandon Nguyen, Curtis Lee, Joshua Leung
Department of Computer Science, UC Santa Cruz

ABSTRACT
Efficient analysis of resumes have long been an ongoing problem since the inception of the internet. In this paper we propose to help solve the part of resume analysis through multi-class text classification. We will discuss our experiment and the techniques we used in order to help benefit the limited amount of labeled data we had. Furthermore, we apply common neural network architectures and explain our encounters with different settings we applied to our neural network model. Our paper will hopefully be of use to individuals or groups of machine learning practitioners since our idea implemented here are more practical than theoretical.

Dataset:

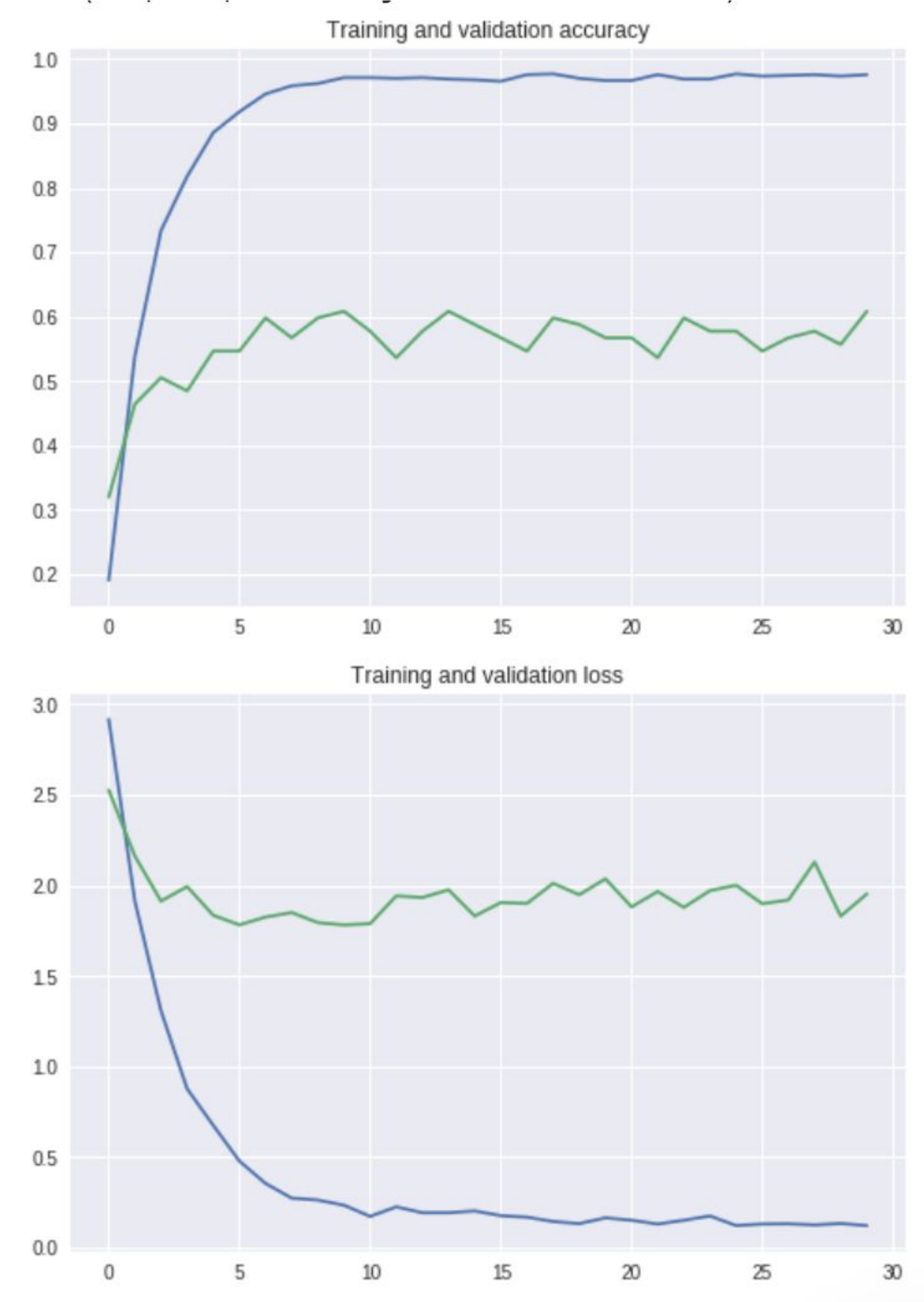
We found the dataset for our project on kaggle.com. It is a .csv file with three columns, an ID number, category, and resume. The category column consist of different job categories such at Engineering, HR, and Aviation just to name a few. The Resume column has the entire text of each resume. The size of the original dataset was 1219 rows and three columns.

We cleaned the data by dropping empty and duplicate rows. Then we reindexed the entire dataset. After cleaning we had a total of 1206 resumes. We further preprocessed our textual data using the Natural Language Toolkit.

Model:

We structured and used a Bag of Words Feed-Forward Neural Network Model to extract features from text to our model. In the neural network model, there is a dense layer with a relu activation layer, a dropout layer for regularization, an additional dense layer for the resume categories, and a softmax to determine which category the resume belongs to.

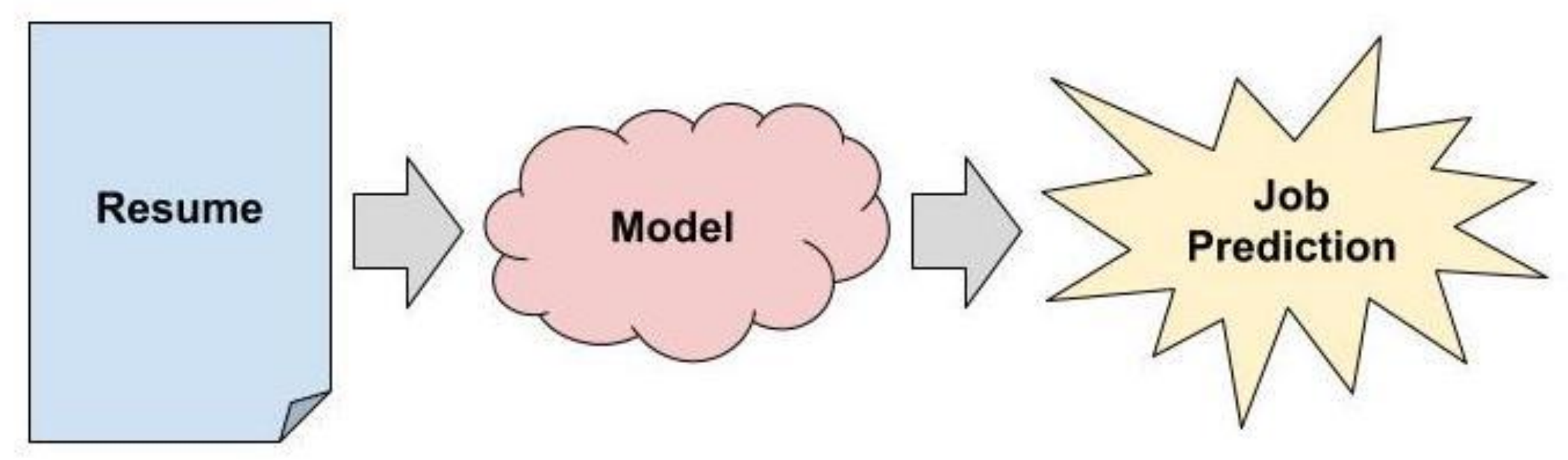
Original Model Accuracy



Initial Attempt:

The blue line represents the training set and the green line represents the validation set. The training set plateaus around 96% accuracy but the validation set plateaus at around 60% accuracy. Having such a big difference in accuracy, was a major red flag for us because it meant we were overfitting.

Flow of the Model

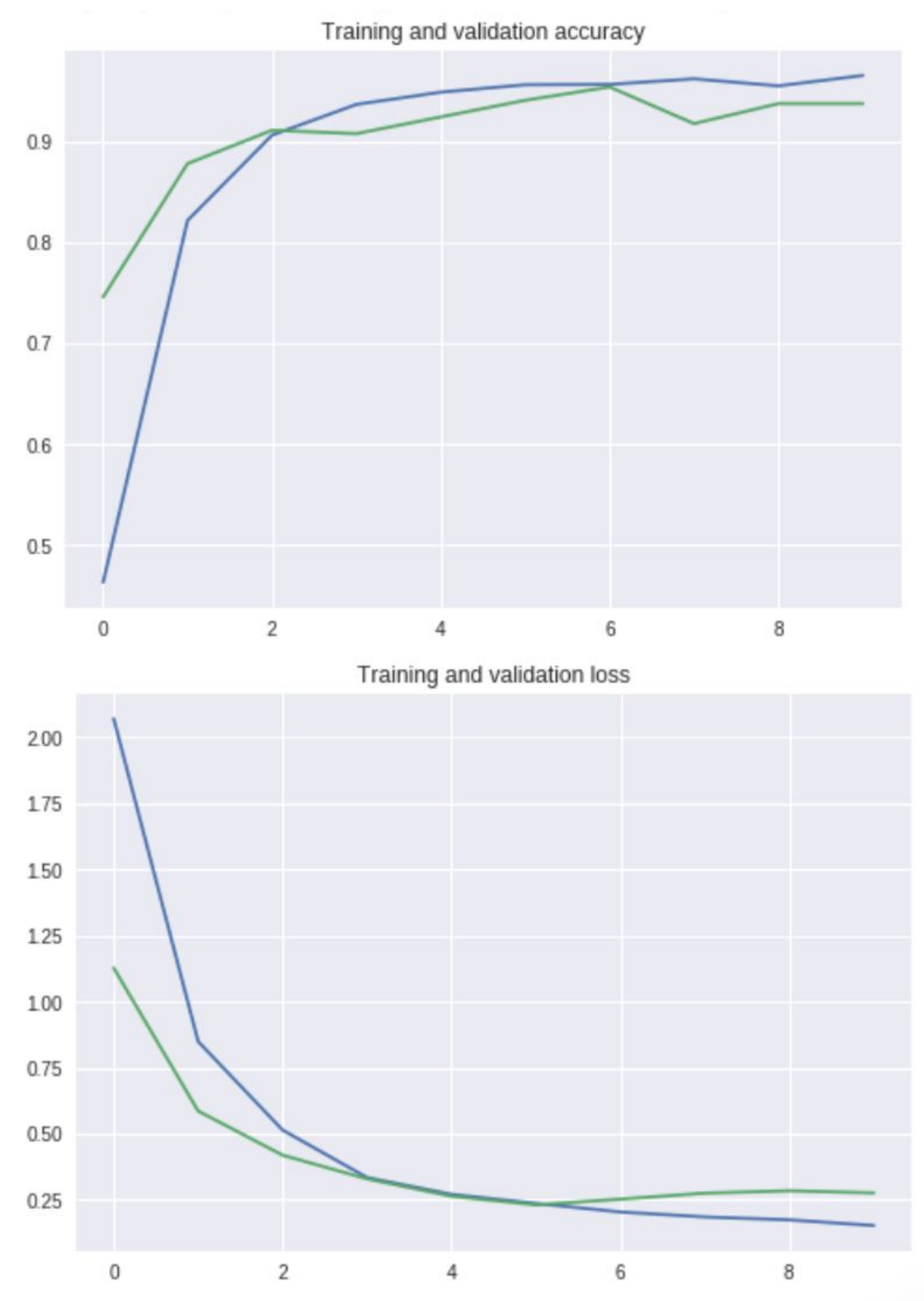


Data Augmentation

We needed more data, so we attempted to augment the resumes we already had. For each resume, we tokenized the sentences and randomly shuffled them to create more resumes. We created up to 10 new resumes from each preexisting resume, only storing non duplicate shuffled resumes.

Original text	Augmented Text
Hello, my name is Sammy. I am taking CMPS 144 at UCSC. I love Machine Learning. Go Banana Slugs!	Go Banana Slugs! I am taking CMPS 144 at UCSC. Hello, my name is Sammy. I love Machine Learning.

First Augmentation



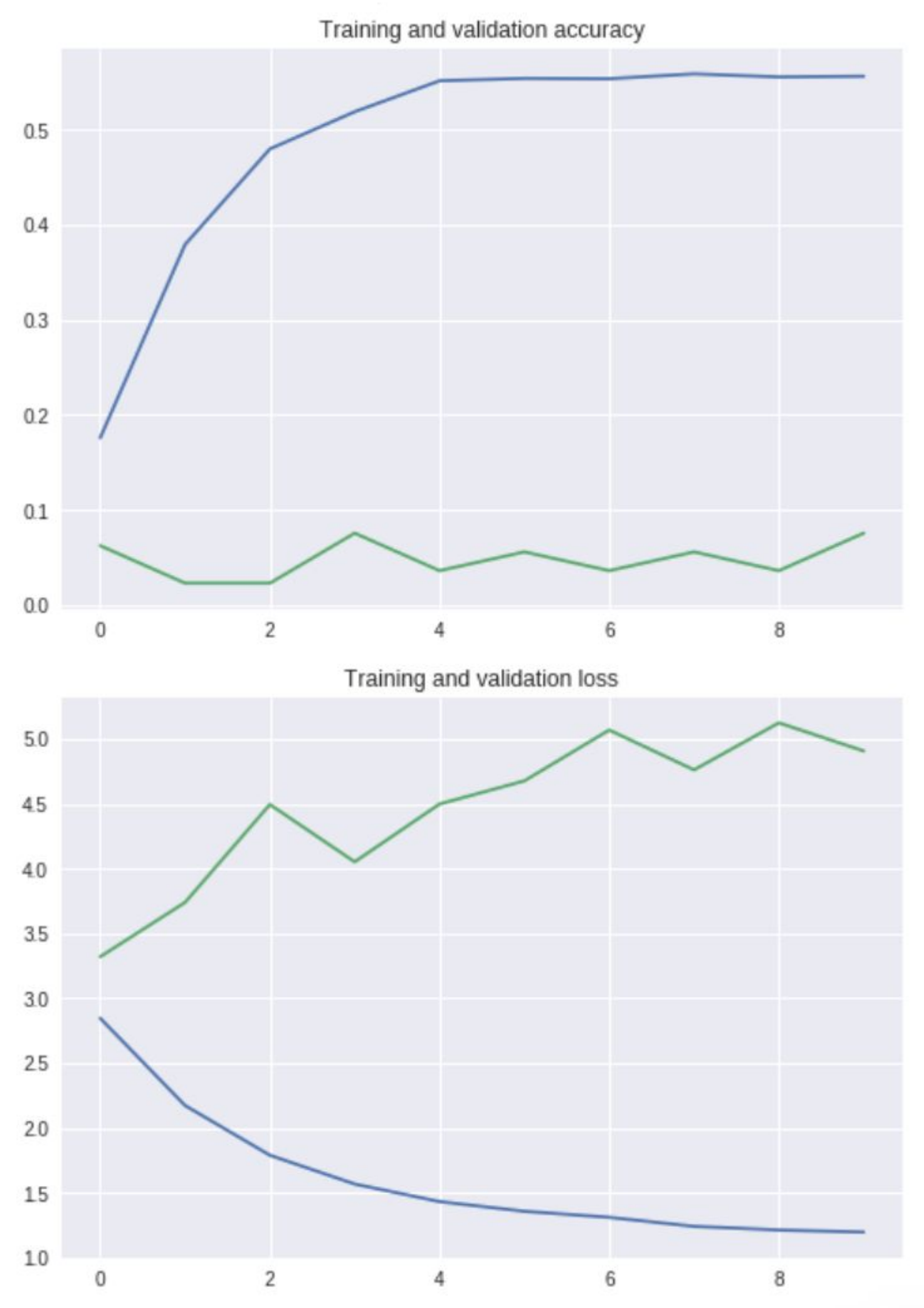
Analysis

This method gave us 93% accuracy without overfitting! However, we realized a major flaw. We were augmenting the dataset as a whole and then splitting it randomly. We were essentially testing our model with our training data because each resume that was augmented was the same vector as the original.

Second Attempt to Augmentation

Next, we tried to split the data into a training set and a validation set first. Then we augmented each set separately so the two sets did not have any resumes in common.

Second Augmentation



Error Analysis

Our model was now heavily over fitting because it was training on the augmented resumes. Each resumes words turned into a vector and each augmented resume had the same vector as it's original.

Another reason why, our model was not getting a higher accuracy is because our data was quite unbalanced. For example, we had 119 Engineering resumes compared to 12 Architecture resumes.

We had to rethink how else we could get more resumes from augmentation. Since our model made the vectors of the resumes based on their words, we decided to change the words themselves. We are currently trying to use databases like WordNet to replace words in the resumes with synonyms. This way, the resumes have the same structure but different words to make new vectors for our model.

Original text	Augmented Text
Hello, my name is Sammy. I am taking CMPS 144 at UCSC. I love Machine Learning. Go Banana Slugs!	Hey, my name is Sammy. I was enrolled in CMPE 100 at University of California, Santa Cruz. I like Machine Learning. Go Banana Slugs!

CONCLUSION

We found that the best results we were able to get was our initial 60% accuracy and our model was heavily overfitting. We needed a larger dataset with an equal number of resumes in each category. Ideally, we want to add new unique resumes to our dataset but given the timespan of the project we tried to simulate this by augmenting the resumes.

ACKNOWLEDGEMENTS
We would like to thank our Professor, Narges Norouzi and our Teaching Assistants Rafael Espericueta and Marcelo Siero for giving us guidance through an entirely new field to us.
We would also like to thank our friends and family for their care and support