2 Process

2.1 NLP

2.1.1 Preprocessing

The stages of our NLP preprocessing are as follows

1. *Data cleaning.* We used regular expression in order to remove punctuation and special characters and substitute words that were spelled wrong in the original data set. We also turned all the characters into lowercase characters and removed extra spaces.
2. *Tokenization.* We broke a stream of textual data into the smallest units of a sentence called tokens.
3. *Filtering.* We removed stop words, words which are filtered out because they are insignificant.
4. *Part-of-speech tagging.* We used part-of-speech tagging, which is the process of marking up a word in a text as corresponding to a particular part of speech, based on both its definiton and its context.
5. *Lemmatisation.* We grouped together the inflected forms of a word into its lemma, the canonical form of a set of word forms.

2.1.2 Document similarity

After performing NLP preprocessing we wanted to check if resumes belonging to the same categories were similar to each other. We chose *Cosine similarity* to measure the similarity between the resumes. Cosine similarity is a measure of similarity between two sequences of numbers (the vectorized resumes in our case). The sequences are viewed as vectors and the cosine similarity is defined as the cosine of the angle between them. Cosine similarity gives a useful measure of how similar two documents are likely to be.

For each resume we computed

- the mean of the cosine similarity between the resume and all the other resumes
- the mean of the cosine similarity between the resume and all the resumes belonging to the same category

Then, for each category, we computed the mean of the two values. We can see from *figure 4* that resumes belonging to the same category have a much higher similarity, so we decided to proceed with the classification.

2.2 Classification

The goal of this project is to develop a classifier that is able to perform automatic CV classification. The classifier will be used by employment offices and postgraduate orientation centers so that the classifier will suggest a list of suitable jobs to the user based on his/her resume.

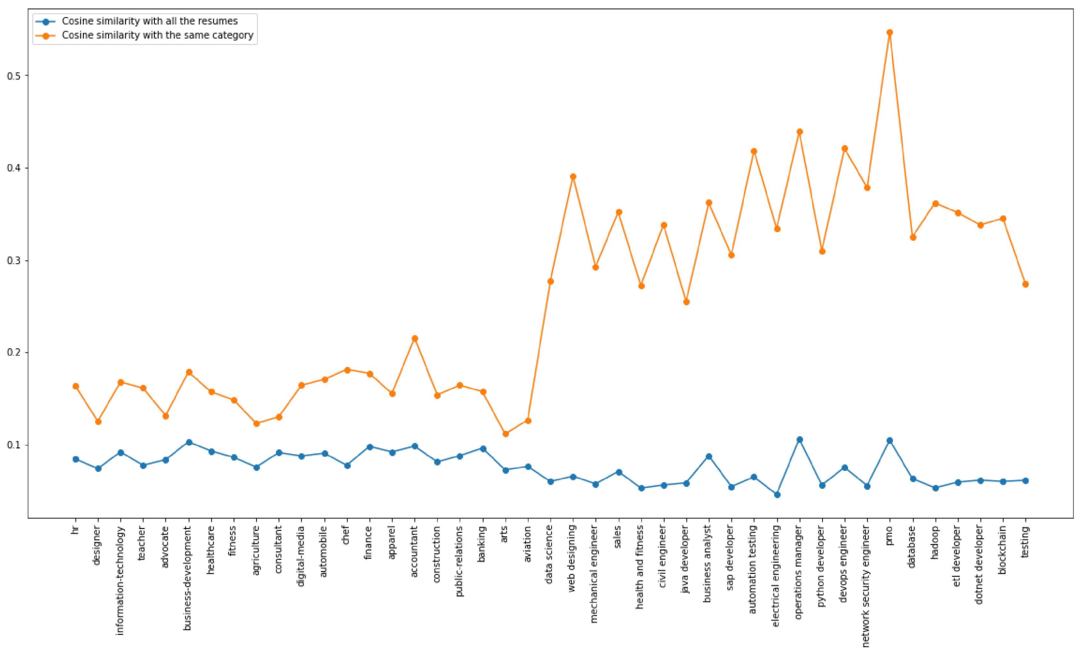

Figure 4: Cosine similarity

2.2.1 Text Vectorization: Term Frequency - Inverse Document Frequency (TFIDF)

TFIDF is based on the Bag of Words (BoW) model, which converts the text into a feature vector by counting the occurrence of words in a document. The **Term Frequency (TF)** is a measure of the frequency of a word in a document. TF is defined as the ratio of a word's occurrence in a document to the total number of words in a document.

$$\frac{\text{occurences of w in document d}}{\text{total number of words in document d}}$$

The **Inverse Document Frequency (IDF)** is the measure of the importance of a word. IDF provides weightage to each word based on its frequency in the corpus D.

$$\ln \frac{\text{total number of documents}}{\text{number of documents containing w}}$$

After applying TFIDF, text in resumes can be represented as a TFIDF sparse matrix of dimensions (number of documents) x (vocabulary words). The value corresponding to each word represents the importance of that word in a particular document.

2.2.2 Classifiers comparison

We evaluated different classifiers using the accuracy as evaluation metric and a K-Fold Cross Validation with K = 10 for a better accuracy result.

Considering this metric the decided to use the *Linear Support Vector* classifier

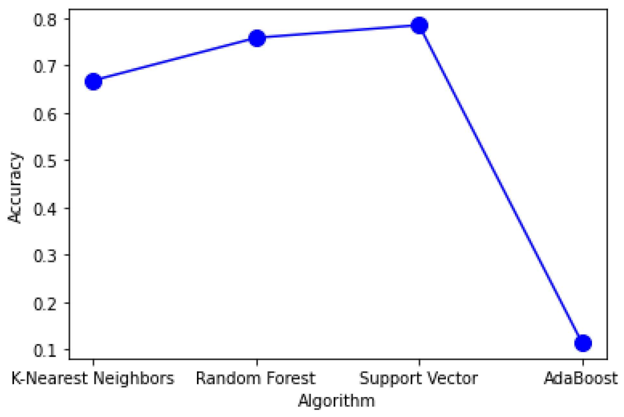

Figure 5: Cosine similarity

2.2.3 Classification remarks

Looking at *figure 6* we can see that the categories with the higher misclassified resumes are the ones that have the lowest similarity according to *figure 4*.

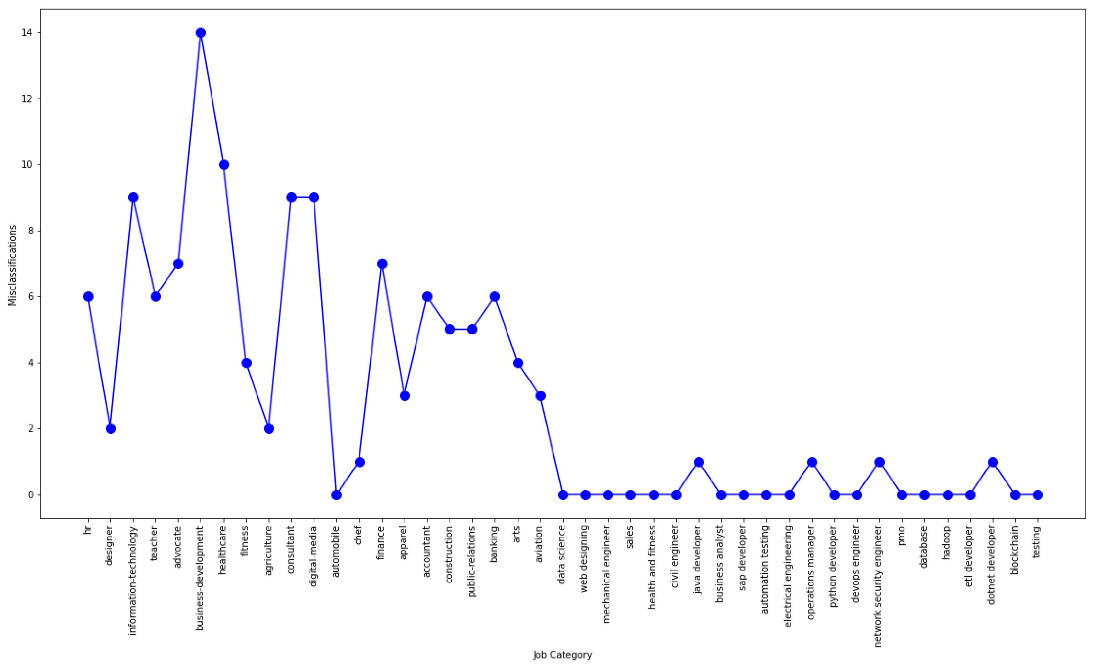

Figure 6: Misclassifications

3 Results

On the cleaned data, a model was built based on Classification. Based on the curriculum and category, the model was designed to classify the curriculum into the right category.

3.1 Classification

The classification was done using Linear Support Vector model: a supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. The Linear Support Vector classifier yielded an accuracy of 0.78 on 10-fold cross-validation. The confusion matrix obtained with our model is shown below, highlighting the discrepancies between predicted and actual labels.