

2025

# Text Classification

## A business opportunity

Quang Dong (Wade) Nguyen - Data Analyst



# Introduction

Using advanced ML techniques to classify texts and predict sentiments



Text classification is a ML technique that categorizes open-ended text into **predefined labels, organizing various types of content like articles and medical research**. With increasing data in modern businesses, ML enables efficient text processing, automating tasks such as grouping, filtering, and sentiment analysis. These methods **enhance ETL processes and predictive models, reducing the need for manual intervention in data-heavy environments**.



## Scenario

In this example, our private client partner, Nexy, a social media startup whose business finances are derived from developing apps and algorithms that facilitate users' use of social media platforms. However, due to their lack of machinery infrastructure & financial supports, we (DataNet) aim to reach out and provide assistance to Nexy. Therefore, our goal in this research is to help construct optimal data models using dataset resources that are provided by Nexy.



# About us

DataNet is a premier data consultancy firm catering to data-driven industries such as banking and oil companies, offering large-scale data assessment, modelling, processing and insight extraction services to enhance user flexibility and data comprehension.

**51,000**  
Total customers

**\$50,000 USD**  
Monthly Revenue



*mission*



# Table of content

DataNet offers large-scale data modeling and insight extraction services to enhance user flexibility and data comprehension.

## Overview

- Problems identification
- Nexy | Data Profile

## Apache Hadoop & ETL

- ETL & Hadoop
- Process report

## Predictive modelling

- Model construction
- Model configuration & settings
- Model evaluation
- Test model conclusion

## Insight recommendation

- Conclusion & Recommendation
- Next steps



# Overview

# Nexy | Problem statement

## Problem:

Though Nexy has made efforts to establish itself as a leading company in the media sector, it still **lacks technological integration and management capabilities**. The absence of technological infrastructure at Nexy, a social media company, leads to **its disadvantages in building data models to forecast future data**.

## Discussed solution:

After the discussion with the data team in Nexy, we have decided to offer suitable data streamlining services in which we process large volumes of data using our central Hadoop Bigdata Server, construct the basis of data prediction models, setting a limited threshold to monitor the successiveness of the models.





2025

## Nexy | reviews data profile

- Given 60 sets of reviews data collected from their social media platform, Nexifly, in which the data store information about a certain movie's reviews in .txt format.
- Though the original package of dataset contains approximately 50,000 movie reviews. To adhere by their data policy reason, a small, modified version (i.e. identity masked with ID) is provided only for experimentation until proven effective.

*Goals*

DataNet is tasked with creating a completely automated program to perform **sentiment classification** from feature extraction to classification.

# Nexy | data profile

## Nexy data | summary

The original data package includes a total of 50,000 reviews of .txt formatted files & each text file contains approximately 50 to 200 unprocessed words.

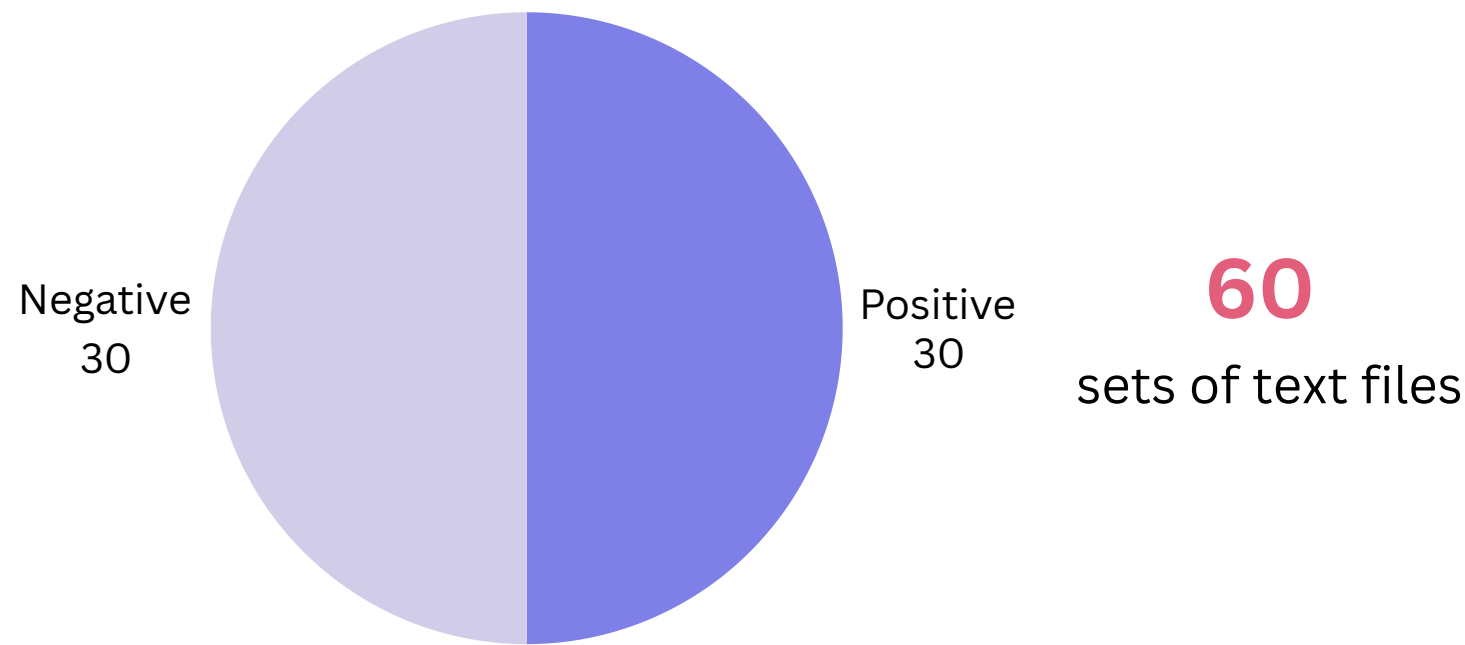


**50,000**  
reviews package

**50 - 200**  
words each

## No. of text files for training & testing

There are 60 review texts in total given by Nexy for training & testing, in which 30 of them are labelled positive and the other 30 are labelled as negative reviews.

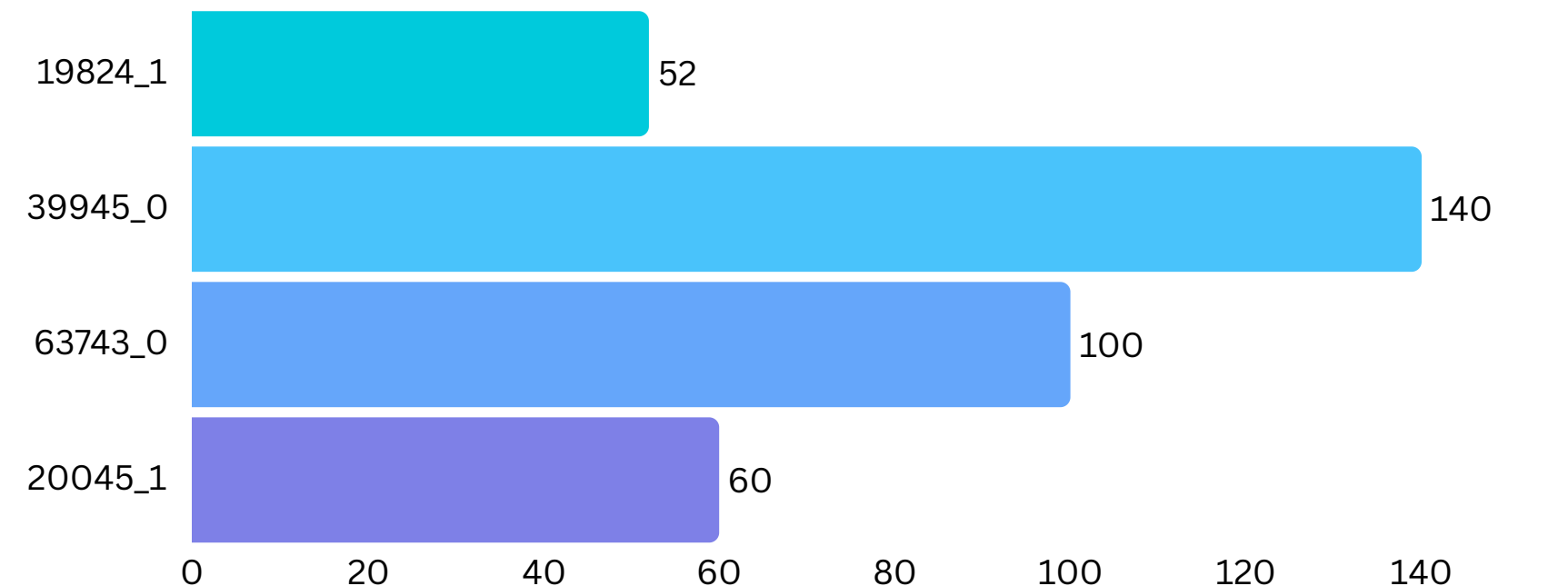


## Rating score across text files.

Each file has been attached with a rating score at the end of its filename. For instance, a filename called 19824\_1.txt means it is a text file with an id number of 19824 has a rating score of 1.

## Word count distribution across text files

There is a total of 60 text files in which each text file contains about 50 to 200 words each, the following graph demonstrates samples of text that represent the whole distribution





# Apache Hadoop & ETL

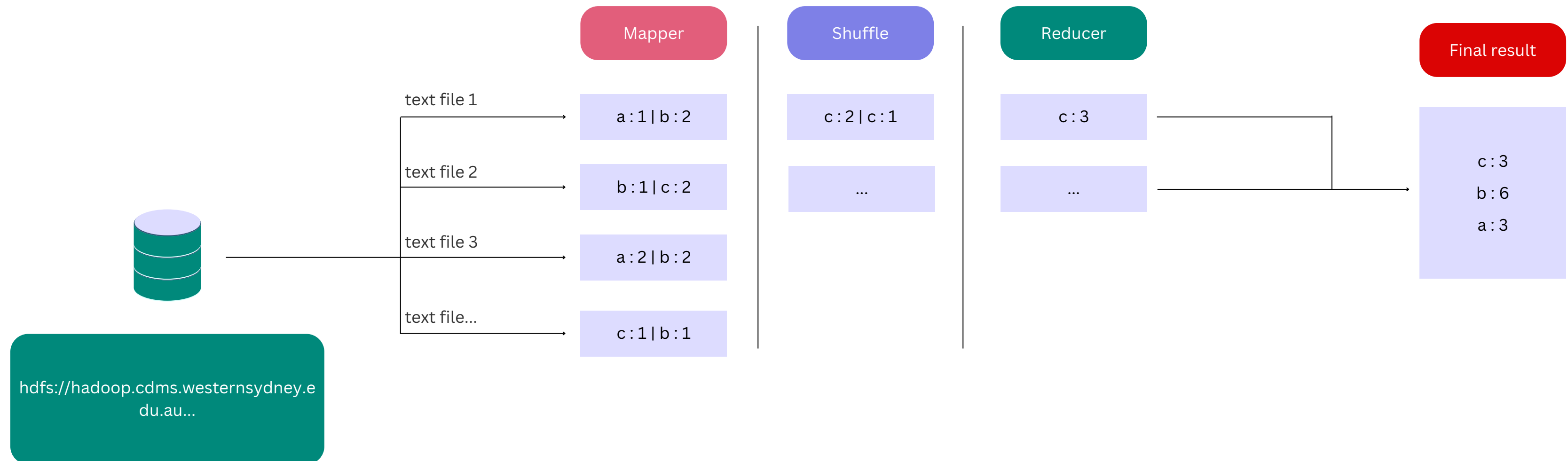
# Nexy | ETL & Hadoop

## Hadoop HDFS definition

According to Hadoop, Hadoop HDFS is a distributed system designed for storing very large files with streaming data access patterns and distributing them across the cluster for processing. It provides high throughput access to application data and is suitable for applications that have large datasets as in this research.

## Hadoop MapReduce definition

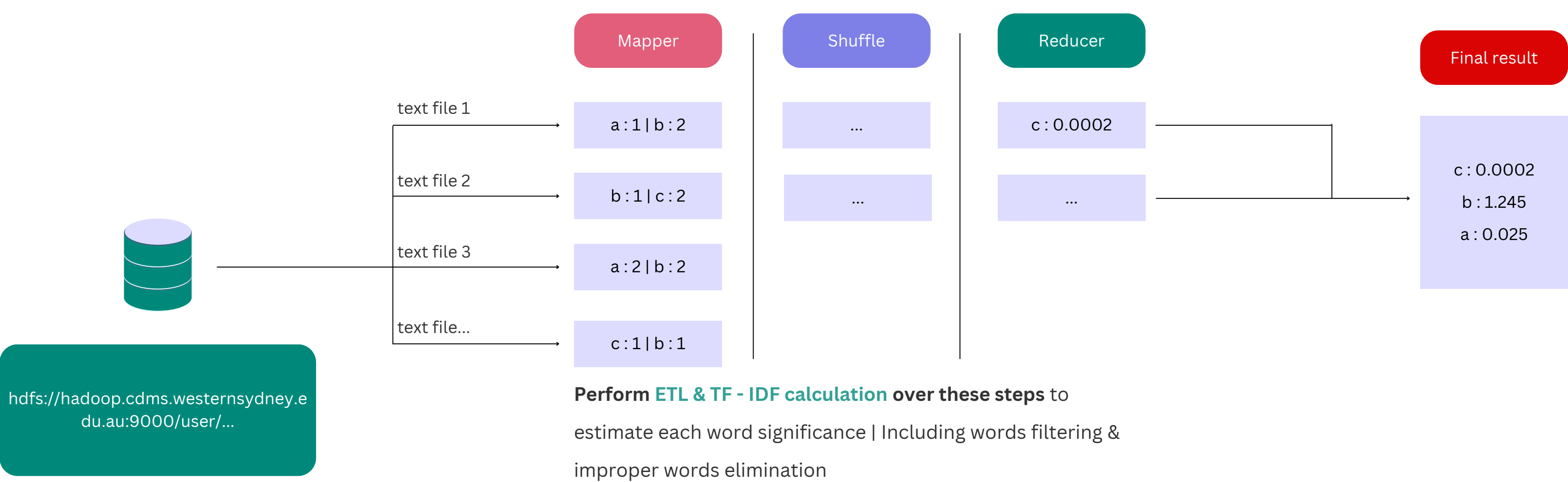
On the other hand, Hadoop MapReduce is a framework to process and retrieve large datasets parallelly in a distributed fashion. It processes the data which is present within HDFS and provides output through similar processing patterns (in this case that is to calculate word-document retrieval values across the review files).



# Nexy | ETL & Hadoop (cont)

## TF - IDF calculation:

TF - IDF, or Term Frequency - Inverse Document Frequency, is a widely used statistical method to measures how important a term is within a document relative to a collection of documents. In simple words, TF-IDF calculate the significance of each word in the overall set of documents



## Conclusion

Efficiency: Running process with Hadoop MapReduce takes around 15 seconds for a total of 60 unprocessed files to be processed and producing TF - IDF values for each word.

# Nexy | Products report

After obtaining the final results, we are able to achieve the following files that describes:

## Original words

[”abaddon”, ”abad”, ”ab”, ”ab’s”, ”acczde”, ”abbs”,  
 ”aebas”, ”abc’d”, ”acted”, ”act”, ”accs”, ”akkkkk”, ...]

## After pre-processing

["abandon", "abc's", "abc", "ability", "about", "above",  
"absolutely", "abused", "accents", ...]

## Sample of final word lists

[ 'abandon', 'abc's', 'abc', 'ability', 'about', 'above', 'absolutely', 'abused', 'accent', 'accepting', 'accused', 'achieve', 'acid', 'across', 'act', 'acted', 'acting', 'action', 'active', 'actor', 'actors', 'actresses', 'acts', 'actual', 'actually', 'adam', 'adaptation', 'add', 'added', 'address', 'admirable', 'admired', 'admires', 'admit', 'adopting', 'advantage', 'advent', 'adventures', 'adversaries', 'advice', 'aeris', 'affability', 'affair', 'affected', 'affects', 'afford', 'afraid', 'africa', 'african', 'africans', 'after', 'again', 'age', 'agenda's', 'agendas', 'aggressively', 'aging', 'ago', 'agree', 'agreed', 'agreement', 'ahead', 'aileen', 'air', 'aired', 'airplane', 'akane', 'alan', 'alas', 'albert', 'alcoholism', 'alfred', 'alice's', 'alice', ...]

## Corresponding TF-IDF values of each text files

[illegible]

### Dictionary of corresponding rating score of each text files

```
{'10016': 8, '10432': 10, '10689': 3, '10718': 1, '11051': 1, '11536': 3, '11680': 3, '11957': 8, '1195': 1, '12100': 4, '12182': 1, '12339': 10, '12418': 10, '12445': 10, '1729': 1, '1901': 1, '1914': 7, '1950': 4, '1981': 2, '2366': 2, '2409': 8, '2479': 1, '2683': 4, '2716': 10, '2785': 8, '2990': 2, '3088': 8, '3400': 1, '379': 2, '4453': 4, '4654': 2, '4675': 9, '4726': 10, '4801': 7, '4884': 10, '5177': 8, '5690': 2, '6014': 8, '6243': 9, '6325': 9, '6413': 3, '6570': 1, '7169': 3, '7210': 1, '7219': 7, '7836': 9, '7878': 7, '7929': 4, '7988': 9, '8015': 10, '8328': 1, '845': 7, '8524': 4, '8699': 1, '9187': 10, '9203': 9, '9232': 1, '9370': 8, '9461': 7, '9561': 8}
```

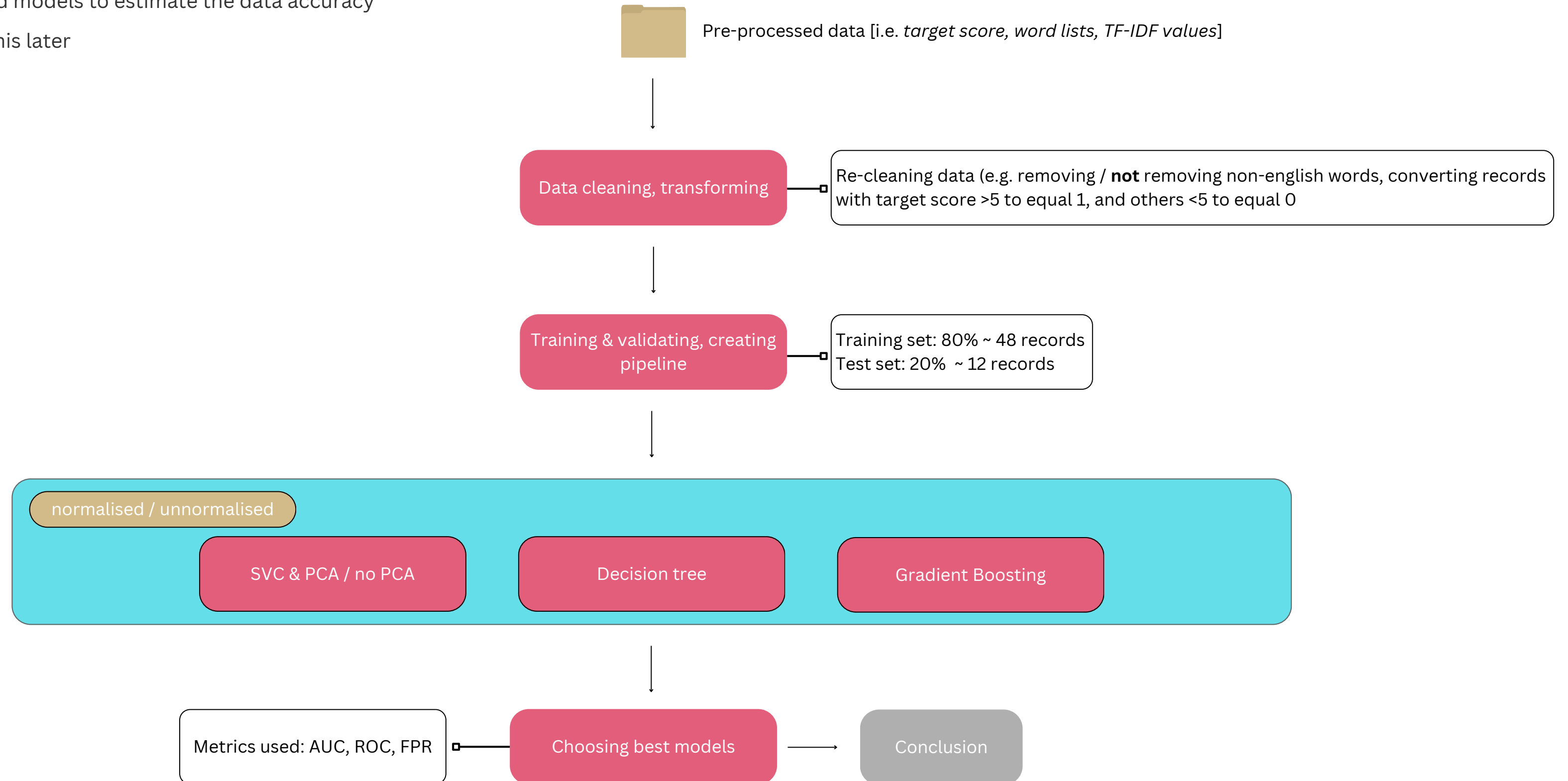


# Predictive Modelling

*Using Scikit-learn library & python*



# Nexy | Model Construction

We use the standard models to estimate the data accuracy  
and so on., will fix this later



# Nexy | Model Configuration & Settings

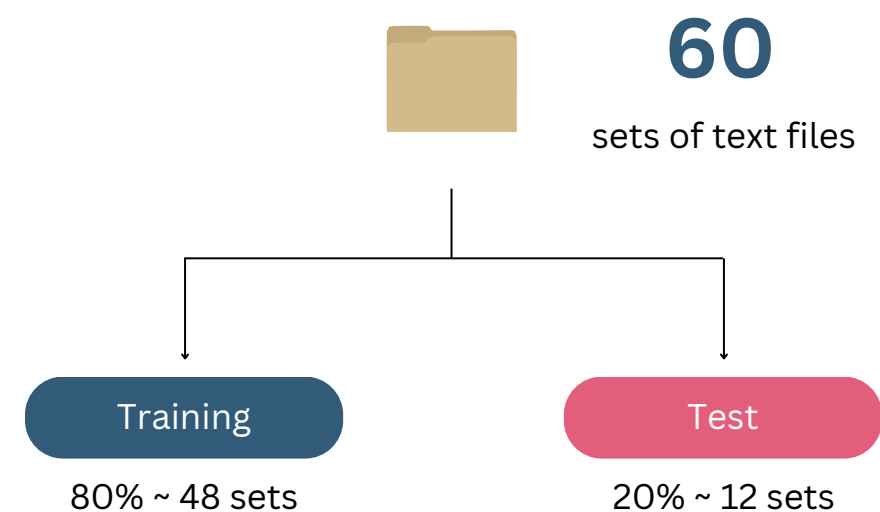
## Insight

As we increase sets of data,   
the number of variables is also increasing 

## Model train-test split

Why do we divide dataset into 80% - 20%:

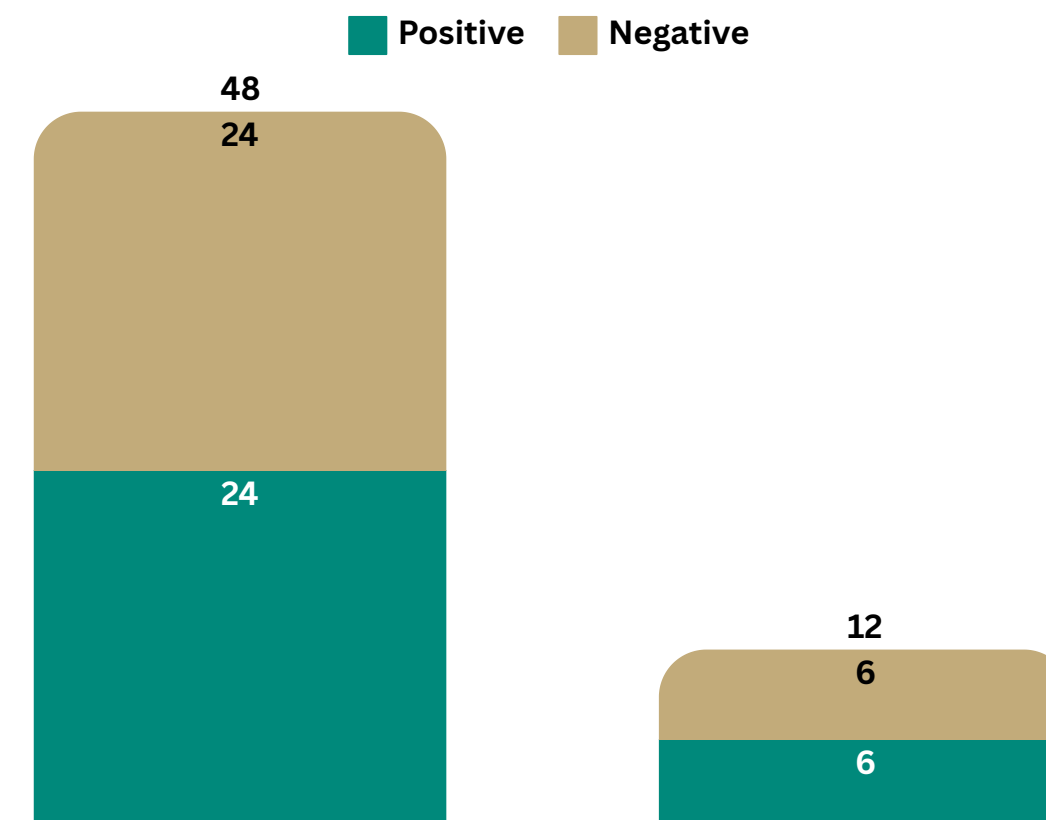
1. prevent overfitting, (we have a small dataset hence overfitting is negligible)
2. evaluate the model more effectively
3. Optimise hyper-parameter



## Stratified sampling & equal splitting

Stratified sampling:

- We split training data and test data with equal number of positives and negatives
- The training set contains 48 samples in which 24 of them are positives and others are negative. Likewise for the test set

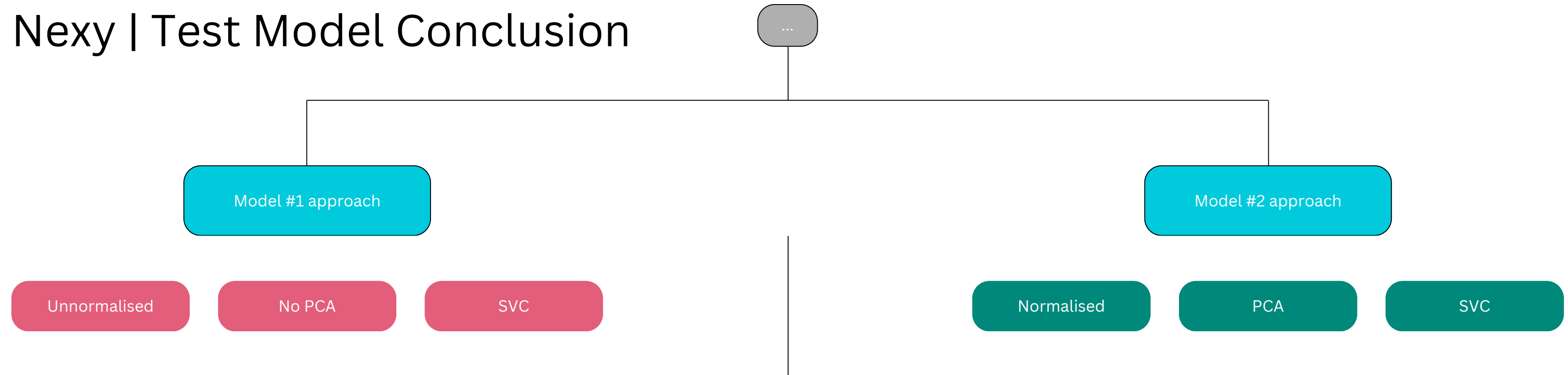


# Nexy | Model Evaluation

	Standardised data	Feature Selection	Model	Gini Score (2 * AUC score - 1)	AUC Score
English word = False	Unnormalised	PCA	SVC	50.00	75.00
		no PCA		66.00	83.33
	Normalised	PCA		66.00	83.33
		no PCA		33.32	66.66
	Unnormalised		Decision tree	-0.3334	33.33
	Unnormalised		Gradient Boosted tree	16.66	58.33
English word = True	Unnormalised	PCA	SVC	50.00	75.00
		no PCA		16.66	58.33
	Normalised	PCA		33.32	66.66
		no PCA		33.32	66.66
	Unnormalised		Decision Tree	33.32	66.66
	Unnormalised		Gradient Boosted Tree	33.32	66.66



# Nexy | Test Model Conclusion

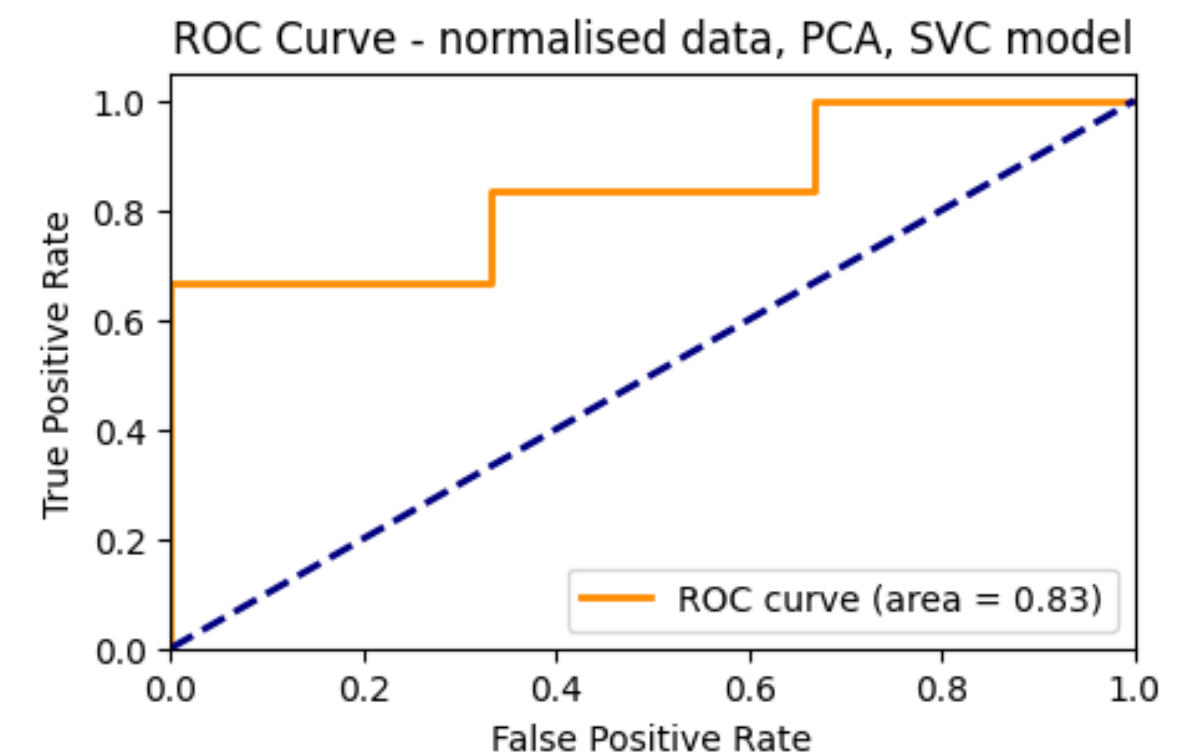


## Conclusion:

- Model #1 approach may work well if the data does not suffer from high-dimensional noise, especially when data has high number of features, in this case it is number of distinct words.
- Model #2 should be chosen if data diverges, meaning if more data are being used to train then normalisation and PCA is necessary to scale data into a normal distribution and reduce the curse of dimensionality. This approach is contributive if the original data has high correlations among features or if removing dimensions supports generalisation.

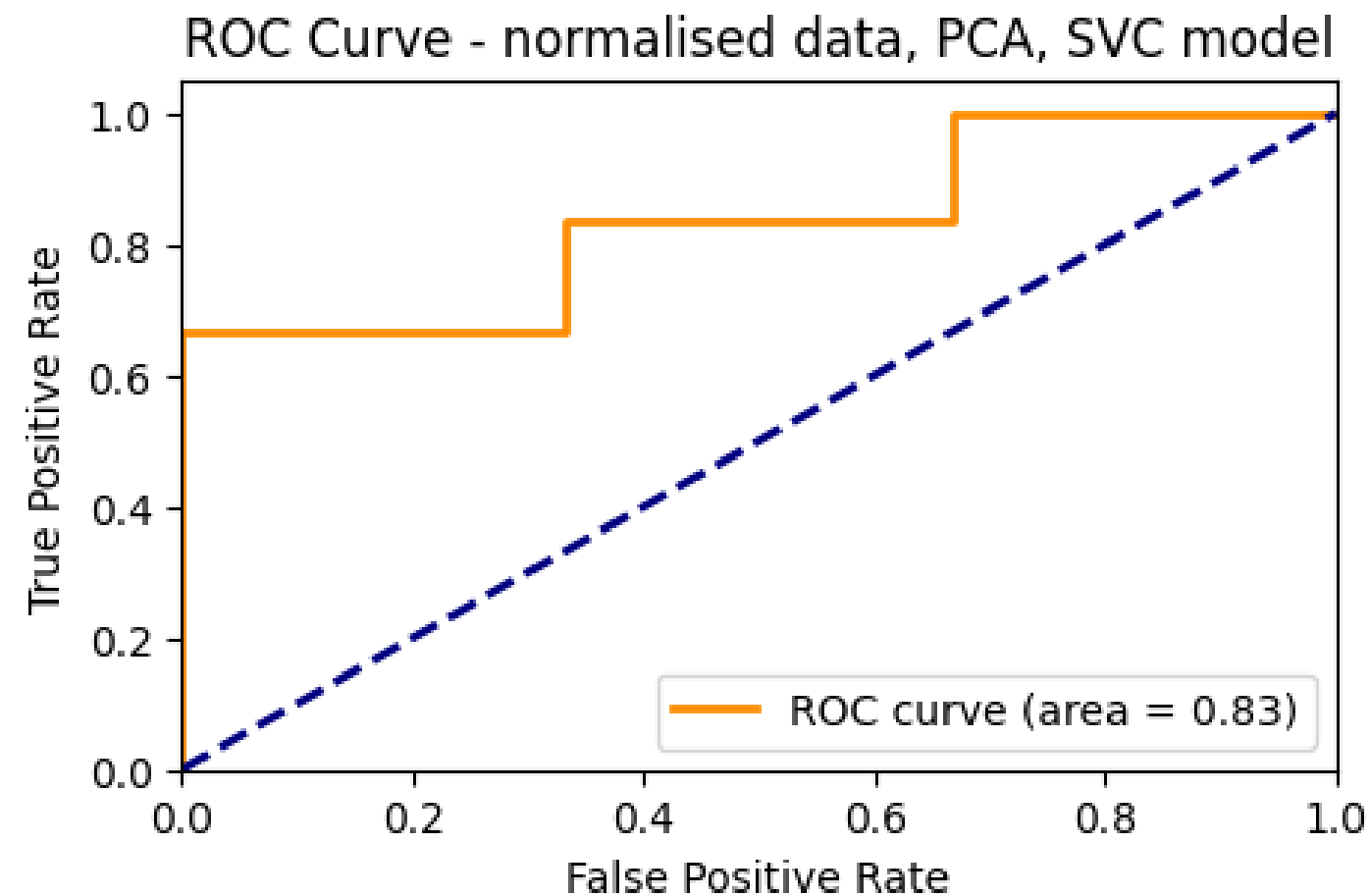
NOTE: the approach done in this experiment was trained with a subset of 50,000 reviews, which is only around 60 reviews. Therefore, Model #2 is exemplified to be used in this approach to estimate the resulting values.

## ROC Curve with normalised data, PCA, SVC Model



# Nexy | Conclusion & Recommendation

## Reasons for choosing the models



- SVC Model excels in classification tasks since it has the ability to search for optimal hyperplanes and classify non-linear data through kernel function. This means for a training models with many variables. With its available kernel functions (polynomial, RBF, and sigmoid) to map data into higher-dimensional space, SVC can search for the best hyperplanes that enable it to classify more accurately. This makes it easier to capture feasible non-linear decision boundaries than with linear models.
- PCA ensures reducing the complexity and noise of data while highlighting the most important features and relationships. For our models, this means as we go on in longer term data assessment, there will be more data to be trained. This means the number of variables could exponentially increase if each text files contains a new word. Therefore, PCA is there to release data from the curse of dimensionality while capturing the important features only.
- Normalisation improves data quality, and performance of ML algorithms. This step is important during and after training the dataset. Since after we obtain a fixed training models we will not be going through the ETL with Hadoop MapReduce to obtain the normalised values anymore. Hence the normalised values we obtained after the ETL steps will be applied to the unforeseen data.

Therefore, **model #2 [normalisation, PCA, SVC]** is better for application than model #1. As number of data observations increase, the more data being used to train the models means that the more needs for data to be rescaled and reduced in number of dimensions, hence eliminating the curse of dimensionality and reduce the size it takes to train models.

# Nexy | Next steps

	Final actions
Data Abundance	Consistently provides the company with more diverse, classified data (i.e. hashtags, topics, posts, comments, etc) for data assessments & model buildings.
Data redundancy & quality assurance	Avoid / Remove spam, or emoji-based texts,
Task-based models	Categorises data into separated models serving for separated tasks



For further information, please contact the following student:

- (University) Quang Dong Nguyen - 20744696@student.westernsydney.edu.au
- (Personal email) Quang Dong Nguyen - dongnguyen12122003@gmail.com

Thanks