# Problem Statement

# Multi-Class Text Classification

## Problem Formulation

The problem is supervised text classification problem, and our goal is to investigate which supervised machine learning methods are best suited to solve it.

### Data Exploration

```
import pandas as pd
from google.colab import drive
drive.mount('/content/gdrive')
```

```
Mounted at /content/gdrive
```

```
[2] df = pd.read_csv("/content/gdrive/MyDrive/Colab Notebooks/root ai internship work/root2ai - Data.csv")
```

```
df.head()
```

| | Text | Target |
|---|---|---|
| 0 | reserve bank forming expert committee based in... | Blockchain |
| 1 | director could play role financial system | Blockchain |
| 2 | preliminary discuss secure transaction study r... | Blockchain |
| 3 | security indeed prove essential transforming f... | Blockchain |
| 4 | bank settlement normally take three days based... | Blockchain |

- **Input: Text**

- **Output: Target**

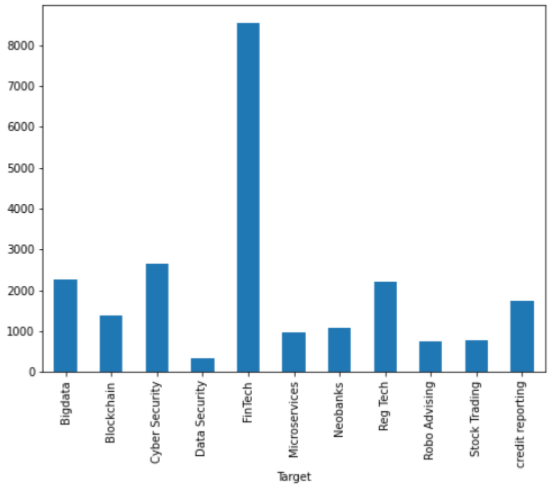
## Perform Mapping

```
from io import StringIO
col = ['Target', 'Text']
df = df[col]
df = df[pd.notnull(df['Text'])]
df.columns = ['Target', 'Text']
df['category_id'] = df['Target'].factorize()[0]
category_id_df = df[['Target', 'category_id']].drop_duplicates().sort_values('category_id')
category_to_id = dict(category_id_df.values)
id_to_category = dict(category_id_df[['category_id', 'Target']].values)
df.head()
```

| | Target | Text | category_id |
|---|---|---|---|
| 0 | Blockchain | reserve bank forming expert committee based in... | 0 |
| 1 | Blockchain | director could play role financial system | 0 |
| 2 | Blockchain | preliminary discuss secure transaction study r... | 0 |
| 3 | Blockchain | security indeed prove essential transforming f... | 0 |
| 4 | Blockchain | bank settlement normally take three days based... | 0 |

## Imbalance Classes

```
#Imbalanced Classes
import matplotlib.pyplot as plt
fig = plt.figure(figsize=(8,6))
df.groupby('Target').Text.count().plot.bar(ylim=0)
plt.show()
```

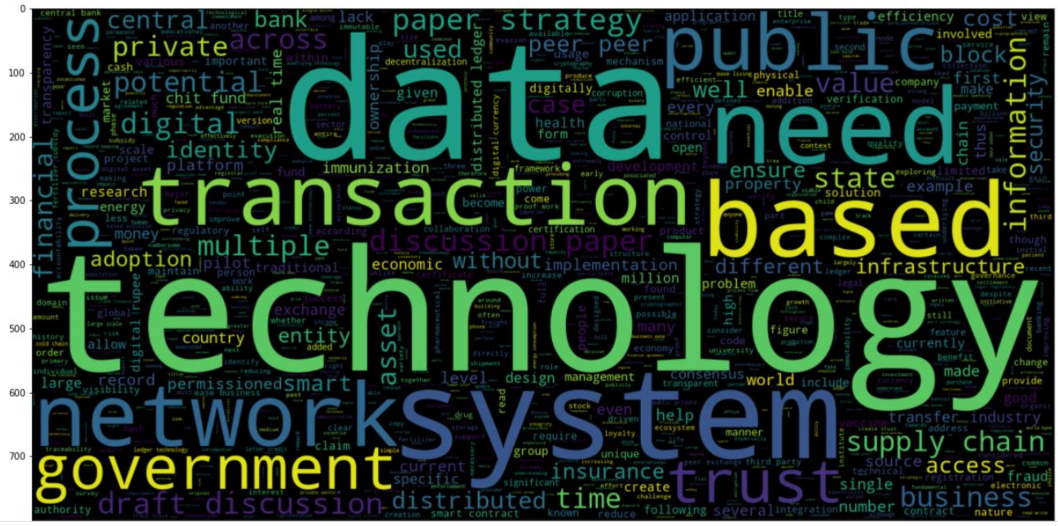



in our case of learning imbalanced data, the majority classes might be of our great interest. It is desirable to have a classifier that gives high prediction accuracy over the majority class, while maintaining reasonable accuracy for the minority classes. Therefore, we will leave it as it is.

## Word cloud:

```
plt.figure(figsize = (20,20))
wc = WordCloud(max_words = 2000 , width = 1600 , height = 800).generate(" ".join(df[df.Target == 'Blockchain'].Text))
plt.imshow(wc , interpolation = 'bilinear')
```

```
<matplotlib.image.AxesImage at 0x7f920a273110>
```




## Text Representation

- Naive Bayes Classifier: the one most suitable for word counts is the multinomial variant:

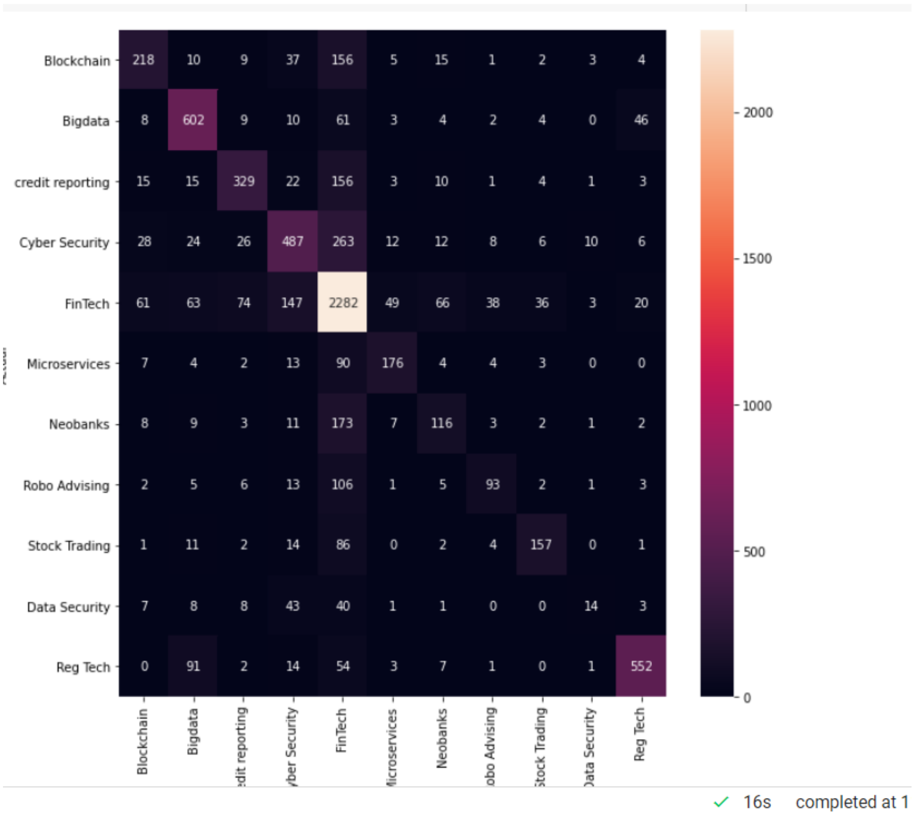
```python
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.naive_bayes import MultinomialNB
X_train, X_test, y_train, y_test = train_test_split(df['Text'], df['Target'], random_stat
count_vect = CountVectorizer()
X_train_counts = count_vect.fit_transform(X_train)
tfidf_transformer = TfidfTransformer()
X_train_tfidf = tfidf_transformer.fit_transform(X_train_counts)
clf = MultinomialNB().fit(X_train_tfidf, y_train)
```

# Model Selection

```
[19] cv_df.groupby('model_name').accuracy.mean()
```

```
model_name
LinearSVC                 0.559582
LogisticRegression        0.571211
MultinomialNB             0.545440
RandomForestClassifier    0.376768
Name: accuracy, dtype: float64
```

# Model Evaluation



```python
from sklearn import metrics
print(metrics.classification_report(y_test, y_pred, target_names=df['Target'].unique()))
```

```
                   precision    recall  f1-score   support

       Blockchain       0.61      0.47      0.53       460
          Bigdata       0.71      0.80      0.76       749
 credit reporting       0.70      0.59      0.64       559
   Cyber Security       0.60      0.55      0.58       882
          FinTech       0.66      0.80      0.72      2839
    Microservices       0.68      0.58      0.63       303
         Neobanks       0.48      0.35      0.40       335
    Robo Advising       0.60      0.39      0.47       237
    Stock Trading       0.73      0.56      0.64       278
    Data Security       0.41      0.11      0.18       125
         Reg Tech       0.86      0.76      0.81       725

         accuracy                           0.67      7492
        macro avg       0.64      0.54      0.58      7492
     weighted avg       0.67      0.67      0.66      7492
```

We can further tune is model with various things because Conventional algorithms are often biased towards the majority class, not taking the data distribution into consideration.

For some cases, such as fraud detection or cancer prediction, we would need to carefully configure our model or artificially balance the dataset, for example by under sampling or oversampling each class.