## Importing the requried libraries

In [1]:

```
import nltk
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
import re
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
```

## Importing the data set

In [2]:

```
JOB = pd.read_csv("C:/Users/naikc/Downloads/Job titles and industries.csv")
```

# Exploratory Data Analysis

In [3]:

```
### Finding the first 10 columns and row's of the data set ###
```

In [4]:

```
JOB.head(10)
```

Out[4]:

| | Job Title | Industry |
|---|---|---|
| **0** | technical support and helpdesk supervisor - co... | IT |
| **1** | senior technical support engineer | IT |
| **2** | head of it services | IT |
| **3** | js front end engineer | IT |
| **4** | network and telephony controller | IT |
| **5** | privileged access management expert | IT |
| **6** | devops engineers x 3 - global brand | IT |
| **7** | devops engineers x 3 - global brand | IT |
| **8** | data modeller | IT |
| **9** | php web developer £45,000 based in london | IT |

In [5]:

```
### Finding the last 10 columns & row's of the given data set ###
```

In [6]:

```
JOB.tail(10)
```

Out[6]:

| | Job Title | Industry |
|---|---|---|
| **8576** | marketing & social media specialist | Marketing |
| **8577** | senior php developer | Marketing |
| **8578** | social media graphic designer | Marketing |
| **8579** | sponsorship sales executive | Marketing |
| **8580** | marketing specialist | Marketing |
| **8581** | data entry clerk | Marketing |
| **8582** | content creator | Marketing |
| **8583** | sales & marketing manager | Marketing |
| **8584** | marketing & digital marketing consultant | Marketing |
| **8585** | creative copywriter (arabic/english) | Marketing |

In [7]:

```
### Find the total number of colmun's & row's in the data set ###
```

In [8]:

```
JOB.shape
```

Out[8]:

```
(8586, 2)
```

In [9]:

```
## Number of Columns and Row's ##
```

In [10]:

```
print('Count of columns in the data is:  ', len(JOB.columns))
print('Count of rows in the data is:  ', len(JOB))
```

```
Count of columns in the data is:   2
Count of rows in the data is:   8586
```

In [11]:

```
### Find the data types of the given data set ###
```

In [12]:

```
JOB.dtypes
```

Out[12]:

```
Job Title    object
Industry     object
dtype: object
```

In [13]:

```
## Checking the unique variables in data set ##
```

In [14]:

```
JOB.nunique()
```

Out[14]:

```
Job Title    3890
Industry        4
dtype: int64
```

In [15]:

```
## Chekcing the unique variable in Industry vaiable ##
```

In [16]:

```
JOB['Industry'].unique()
```

Out[16]:

```
array(['IT', 'Marketing', 'Education', 'Accountancy'], dtype=object)
```

In [17]:

```
## Chekcing the unique variable in Job Title vaiable ##
```

In [18]:

```
JOB['Job Title'].unique()
```

Out[18]:

```
array(['technical support and helpdesk supervisor - county buildings, ayr soa04086',
       'senior technical support engineer', 'head of it services', ...,
       'sales & marketing manager',
       'marketing & digital marketing consultant',
       'creative copywriter (arabic/english)'], dtype=object)
```

In [19]:

```
## Checking the number of counts in Job Title variable ##
```

In [20]:

```
JOB['Job Title'].value_counts()
```

Out[20]:

```
marketing executive                                                     91
php developer                                                           54
trainee network technician                                              53
software developer                                                      53
marketing manager                                                       49
                                                                        ..
c# developer (web apps)                                                  1
start up recruitment consultancy - 2 x trainee recruitment consultant    1
devops engineer - aws - machine learning & ai company!                   1
2nd line desktop support engineer, build laptops/desktops, sccm          1
german language teacher                                                  1
```

```
Name: Job Title, Length: 3890, dtype: int64
```

In [21]:

```
## Checking the number of counts in Industry variable ##
```

In [22]:

```
JOB['Industry'].value_counts()
```

Out[22]:

```
IT             4746
Marketing      2031
Education      1435
Accountancy     374
Name: Industry, dtype: int64
```

# Data Cleaning or Data Wrangling

In [23]:

```
### Finding the Null values or the missing values in data set ###
```

In [24]:

```
JOB.isnull().sum()
```

Out[24]:

```
Job Title    0
Industry     0
dtype: int64
```

In [25]:

```
def cleaner(text):
    text = text.lower()
    text = re.sub("german[^\s]+","",text)
    text = re.sub("bournemouth[^\s]+","",text)
    text = re.sub("international[^\s]+","",text)
    text = re.sub("flex[^\s]+","",text)
    text = re.sub("15[^\s]+","",text)
    text = re.sub("flexible[^\s]+","",text)
    text = re.sub("numerous[^\s]+","",text)
    text = re.sub("belfast[^\s]+","",text)
    text = re.sub("on[^\s]+","",text)
    text = re.sub("in[^\s]+","",text)
    text = re.sub("up[^\s]+","",text)
    text = re.sub("45[^\s]+","",text)
    text = re.sub("west[^\s]+","",text)
    text = re.sub("london[^\s]+","",text)
    text = re.sub("part[^\s]+","",text)
    text = re.sub("must[^\s]+","",text)
    text = re.sub("2[^\s]+","",text)
    text = re.sub("1/2[^\s]+","",text)
    text = re.sub("no[^\s]+","",text)
    text = re.sub("Â[^\s]+","",text)
    text = re.sub("12[^\s]+","",text)
    text = text.replace("1st","")
    text = re.sub("leading [^\s]+","",text)
    text = re.sub("1st[^\s]+","",text)
    text = re.sub("3rd[^\s]+","",text)
    text = re.sub("2nd[^\s]+","",text)
    text = re.sub("bristol[^\s]+","",text)
    text = re.sub("healthcare[^\s]+","",text)
    text = re.sub("good[^\s]+","",text)
    text = re.sub("pool[^\s]+","",text)
    text = re.sub("6 months[^\s]+","",text)
    text = re.sub("free[^\s]+","",text)
    text = re.sub("invest[^\s]+","",text)
    text = text.replace("o365","")
    text = text.replace("remote","")
    text = text.replace("-"," ")
    text = text.replace("/"," ")
    text = text.replace("("," ")
    text = text.replace(")"," ")
    text = text.replace("soa04086"," ")
    return text
```

In [26]:

```
### Removing the commas, semi colons, & slash's ###
```

In [27]:

```
def remove_stop_words(text):
    sw = stopwords.words("english")
    clean_words = []
    text = text.split()
    for word in text:
        if word not in sw:
            clean_words.append(word)
    return " ".join(clean_words)
```

In [28]:

```
### Stemming process or changing the words ###
```

In [29]:

```
def stemming(text):
    ps = PorterStemmer()
    text = text.split()
    stemmed_words = []
    for word in text :
        stemmed_words.append(ps.stem(word))
    return " ".join(stemmed_words)
```

In [30]:

```
### Running the changed words for the given data set ###
```

In [31]:

```
def run(text):
    text = cleaner(text)
    text = remove_stop_words(text)
    text = stemming(text)
    return text
```

In [32]:

```
### Checking with the Job Title variable ###
```

In [33]:

```
JOB['Job Title'] = JOB['Job Title'].apply(run)
```

In [ ]:

```
### Checking with the first 10 columns of the given data set ###
```

In [34]:

```
JOB.head(10)
```

Out[34]:

| | Job Title | Industry |
|---|---|---|
| **0** | technic helpdesk counti build ayr | IT |
| **1** | senior technic eng | IT |
| **2** | head servic | IT |
| **3** | js fr end eng | IT |
| **4** | network teleph c | IT |
| **5** | privileg access manag expert | IT |
| **6** | devop eng x 3 global brand | IT |
| **7** | devop eng x 3 global brand | IT |
| **8** | data model | IT |
| **9** | php web develop £ base l | IT |

In [35]:

```
### Converting words to vector ###
```

In [36]:

```
tfidf = TfidfVectorizer()
x = tfidf.fit_transform(JOB["Job Title"]).toarray()
```

In [37]:

```
JOB['Industry'] = JOB['Industry'].replace("IT",0)
JOB['Industry'] = JOB['Industry'].replace("Marketing",1)
JOB['Industry'] = JOB['Industry'].replace("Education",2)
JOB['Industry'] = JOB['Industry'].replace("Accountancy",3)
```

In [38]:

```
y = JOB['Industry'].values
y
```

Out[38]:

```
array([0, 0, 0, ..., 1, 1, 1], dtype=int64)
```

## Splitting the Data Set

## Splitting the Data Set

In [39]:

```
### Splitting up the data set into Train & Test data set respectivelly ###
```

In [40]:

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.25)
```

In [ ]:

```

```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js