1. Importing the data

In [1]:
```python
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report

data=pd.read_csv("C:/Users/User/Desktop/job_descriptions.csv")
```

In [2]:
```python
data.head()
```

Out[2]:

	Job Id	Experience	Qualifications	Salary Range	location	Country	latitude	longitude	Work Type	Company Size	...	Contact	Job Title	Role	Job Portal	Job Description	Benefits	skills
0	1089843540111562	5 to 15 Years	M.Tech	59K – 99K	Douglas	Isle of Man	54.2361	-4.5481	Intern	26801	...	001-381-930-7517x737	Digital Marketing Specialist	Social Media Manager	Snagajob	Social Media Managers oversee an organizations...	{'Flexible Spending Accounts (FSAs), Relocatio...	Social media platforms (e.g., Facebook, Twitte...
1	398454096642776	2 to 12 Years	BCA	56K – 116K	Ashgabat	Turkmenistan	38.9697	59.5563	Intern	100340	...	461-509-4216	Web Developer	Frontend Web Developer	Idealist	Frontend Web Developers design and implement u...	{'Health Insurance, Retirement Plans, Paid Tim...	HTML, CSS, JavaScript Frontend frameworks (e.g...
2	481640072963533	0 to 12 Years	PhD	61K – 104K	Macao	Macao SAR, China	22.1987	113.5439	Temporary	84525	...	9687619505	Operations Manager	Quality Control Manager	Jobs2Careers	Quality Control Managers establish and enforce...	{'Legal Assistance, Bonuses and Incentive Prog...	Quality control processes and methodologies St...
3	688192671473044	4 to 11 Years	PhD	65K – 91K	Porto-Novo	Benin	9.3077	2.3158	Full-Time	129896	...	+1-820-643-5431x47576	Network Engineer	Wireless Network Engineer	FlexJobs	Wireless Network Engineers design, implement, ...	{'Transportation Benefits, Professional Develo...	Wireless network design and architecture Wi-Fi...
4	117057806156508	1 to 12 Years	MBA	64K – 87K	Santiago	Chile	-35.6751	-71.5429	Intern	53944	...	343.975.4702x9340	Event Manager	Conference Manager	Jobs2Careers	A Conference Manager coordinates and manages c...	{'Flexible Spending Accounts (FSAs), Relocatio...	Event planning Conference logistics Budget man...

5 rows × 23 columns

In [3]:
```python
print(data.shape)
```

```
(1615940, 23)
```

In [4]:
```python
# Reduce the number of rows to around 3000
data = data.sample(n=3000, random_state=1)
```

In [5]:
```python
print(data.columns)
```

```
Index(['Job Id', 'Experience', 'Qualifications', 'Salary Range', 'location',
       'Country', 'latitude', 'longitude', 'Work Type', 'Company Size',
       'Job Posting Date', 'Preference', 'Contact Person', 'Contact',
       'Job Title', 'Role', 'Job Portal', 'Job Description', 'Benefits',
       'skills', 'Responsibilities', 'Company', 'Company Profile'],
      dtype='object')
```

In [6]:
```python
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 3000 entries, 1451761 to 1528009
Data columns (total 23 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Job Id            3000 non-null   int64  
 1   Experience        3000 non-null   object 
 2   Qualifications    3000 non-null   object 
 3   Salary Range      3000 non-null   object 
 4   location          3000 non-null   object 
 5   Country           3000 non-null   object 
 6   latitude          3000 non-null   float64
 7   longitude         3000 non-null   float64
 8   Work Type         3000 non-null   object 
 9   Company Size      3000 non-null   int64  
 10  Job Posting Date  3000 non-null   object 
 11  Preference        3000 non-null   object 
 12  Contact Person    3000 non-null   object 
 13  Contact           3000 non-null   object 
 14  Job Title         3000 non-null   object 
 15  Role              3000 non-null   object 
 16  Job Portal        3000 non-null   object 
 17  Job Description   3000 non-null   object 
 18  Benefits          3000 non-null   object 
 19  skills            3000 non-null   object 
 20  Responsibilities  3000 non-null   object 
 21  Company           3000 non-null   object 
 22  Company Profile   2991 non-null   object 
dtypes: float64(2), int64(2), object(19)
memory usage: 562.5+ KB
```

Cleaning the data

In [8]:
```python
# checking for missing values
data.isna().sum()
```

Out[8]:
```
Job Id              0
Experience          0
Qualifications      0
Salary Range        0
location            0
Country             0
latitude            0
longitude           0
Work Type           0
Company Size        0
Job Posting Date    0
Preference          0
Contact Person      0
Contact             0
Job Title           0
Role                0
Job Portal          0
Job Description     0
Benefits            0
skills              0
Responsibilities    0
Company             0
Company Profile     9
dtype: int64
```

In [9]:
```python
# Dropping unnecessary variables
# Drop unnecessary columns
unnecessary_columns = ['Job Id', 'Salary Range', 'location', 'Country', 'latitude', 'longitude',
                       'Work Type', 'Job Posting Date', 'Preference', 'Contact Person', 'Contact',
                       'Role', 'Job Portal', 'Benefits', 'Company', 'Company Profile']

data.drop(columns=unnecessary_columns, inplace=True)
```

In [15]:
```python
data.head(1)
```

Out[15]:

	Experience	Qualifications	Company Size	Job Title	Job Description	skills	Responsibilities	Combined_Text
1451761	1 to 9 Years	BA	96517	Human Resources Manager	Talent Acquisition Managers oversee the recrui...	Talent sourcing Interviewing Onboarding Recrui...	Lead talent acquisition efforts, including sou...	talent acquisition manager oversee recruitment...

pre-processing the data

In [10]:
```python
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
import string

# Download stopwords and initialize lemmatizer
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')
stop_words = set(stopwords.words('english'))
lemmatizer = WordNetLemmatizer()

# text preprocessing
def preprocess_text(text):
    # Remove punctuation marks
    text = text.translate(str.maketrans('', '', string.punctuation))
    # Tokenization
    tokens = word_tokenize(text.lower())
    # Removing stopwords and lemmatization
    tokens = [lemmatizer.lemmatize(token) for token in tokens if token.isalnum() and token not in stop_words]
    return ' '.join(tokens)
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\User\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\User\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\User\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

Feature Engineering

In [11]:
```python
# Apply preprocessing to relevant columns and concatenate text
data['Combined_Text'] = data['Job Description'] + ' ' + data['Responsibilities'] + ' ' + data['skills']
data['Combined_Text'] = data['Combined_Text'].apply(preprocess_text)

# Feature extraction
# TF-IDF vectorization for combined text data
tfidf_vectorizer = TfidfVectorizer(max_features=1000)
X_text = tfidf_vectorizer.fit_transform(data['Combined_Text']).toarray()

# Encoding target variable
label_encoder = LabelEncoder()
y = label_encoder.fit_transform(data['Job Title'])
```

Model training

In [13]:
```python
# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X_text, y, test_size=0.2, random_state=42)

# Model training
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)

# Model evaluation
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Logistic regression Accuracy:", accuracy)
print(classification_report(y_test, y_pred))

from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier

# Train and evaluate Support Vector Machine (SVM) model
svm_model = SVC()
svm_model.fit(X_train, y_train)
svm_y_pred = svm_model.predict(X_test)
svm_accuracy = accuracy_score(y_test, svm_y_pred)
print("SVM Accuracy:", svm_accuracy)
print(classification_report(y_test, svm_y_pred))

# Train and evaluate Random Forest model
rf_model = RandomForestClassifier()
rf_model.fit(X_train, y_train)
rf_y_pred = rf_model.predict(X_test)
rf_accuracy = accuracy_score(y_test, rf_y_pred)
print("Random Forest Accuracy:", rf_accuracy)
print(classification_report(y_test, rf_y_pred))
```

```
Logistic regression Accuracy: 0.9083333333333333
              precision    recall  f1-score   support

           0       1.00      0.50      0.67         6
           1       0.60      1.00      0.75         3
           2       1.00      1.00      1.00         9
           3       1.00      0.33      0.50         3
           4       1.00      1.00      1.00        11
           5       1.00      1.00      1.00         2
           6       1.00      1.00      1.00         7
           7       1.00      1.00      1.00         2
           8       1.00      1.00      1.00         2
           9       1.00      1.00      1.00         6
          10       1.00      1.00      1.00         3
          11       0.67      1.00      0.80         2
          12       1.00      1.00      1.00         5
          13       1.00      1.00      1.00         2
          14       0.50      1.00      0.67         3
          15       1.00      1.00      1.00         5
          16       0.50      1.00      0.67         2
          17       1.00      1.00      1.00         5
          18       1.00      1.00      1.00         6
          19       1.00      1.00      1.00         2
          20       1.00      1.00      1.00         8
          21       1.00      1.00      1.00         2
          22       1.00      1.00      1.00         6
          23       1.00      1.00      1.00         7
          24       1.00      1.00      1.00         9
          25       1.00      1.00      1.00         6
          26       1.00      1.00      1.00         3
          27       1.00      1.00      1.00         3
          28       1.00      1.00      1.00         5
          29       1.00      1.00      1.00         2
          30       1.00      1.00      1.00         5
          31       0.53      1.00      0.69        10
          32       1.00      1.00      1.00         5
          33       1.00      1.00      1.00         4
          34       1.00      1.00      1.00         2
          35       0.50      1.00      0.67         2
          36       0.00      0.00      0.00         1
          37       1.00      1.00      1.00         7
          38       0.00      0.00      0.00         1
          39       0.71      1.00      0.83         5
          40       0.64      1.00      0.78         7
          41       1.00      1.00      1.00         1
          42       1.00      0.62      0.77         8
          43       1.00      1.00      1.00         2
          44       1.00      1.00      1.00         6
          45       1.00      1.00      1.00         1
          46       1.00      1.00      1.00         5
          48       0.00      0.00      0.00         1
          49       1.00      1.00      1.00         1
          50       1.00      1.00      1.00         6
          51       0.89      1.00      0.94         8
          52       0.00      0.00      0.00         1
          53       1.00      1.00      1.00         4
          54       1.00      1.00      1.00         4
          57       1.00      1.00      1.00         5
          58       1.00      1.00      1.00         4
          59       0.00      0.00      0.00         1
          61       1.00      1.00      1.00         8
          62       1.00      1.00      1.00         1
          63       1.00      1.00      1.00         5
          64       0.00      0.00      0.00         2
          65       0.75      1.00      0.86         3
          66       0.00      0.00      0.00         1
          67       1.00      0.71      0.83         7
          68       0.86      1.00      0.92         6
          69       1.00      1.00      1.00         4
          70       1.00      1.00      1.00         2
          71       0.82      1.00      0.90         9
          73       1.00      1.00      1.00         5
          74       1.00      0.67      0.80         3
          75       0.55      1.00      0.71         6
          76       0.33      0.20      0.25         5
          77       1.00      0.67      0.80         3
          78       1.00      1.00      1.00         8
          79       1.00      1.00      1.00         1
          80       1.00      1.00      1.00         5
          81       1.00      1.00      1.00         3
          82       0.00      0.00      0.00         1
          83       0.67      0.86      0.75         7
          84       0.83      1.00      0.91         5
          85       1.00      0.50      0.67         4
          86       1.00      1.00      1.00         3
          87       1.00      1.00      1.00         5
          88       1.00      1.00      1.00         5
          89       1.00      1.00      1.00         7
          90       1.00      0.78      0.88         9
          91       1.00      0.80      0.89         5
          92       0.73      1.00      0.84         8
          93       0.00      0.00      0.00         4
          94       1.00      1.00      1.00         4
          95       1.00      1.00      1.00         3
          96       1.00      1.00      1.00         1
          97       1.00      1.00      1.00         4
          99       1.00      1.00      1.00         4
         100       1.00      1.00      1.00         9
         102       1.00      1.00      1.00         2
         103       1.00      1.00      1.00         6
         105       1.00      1.00      1.00         4
         106       1.00      0.67      0.80         3
         107       0.89      1.00      0.94         8
         108       1.00      0.33      0.50         3
         109       0.00      0.00      0.00         2
         110       0.50      0.50      0.50         4
         111       1.00      1.00      1.00         4
         112       1.00      1.00      1.00         6
         113       1.00      1.00      1.00         4
         114       1.00      1.00      1.00         2
         115       1.00      0.50      0.67         4
         116       1.00      0.36      0.53        11
         117       0.00      0.00      0.00         1
         118       1.00      1.00      1.00         1
         119       1.00      1.00      1.00         2
         120       1.00      1.00      1.00         9
         121       1.00      1.00      1.00         3
         122       1.00      1.00      1.00         5
         123       1.00      1.00      1.00         4
         124       1.00      1.00      1.00         8
         125       1.00      1.00      1.00         2
         126       1.00      1.00      1.00         8
         127       0.60      1.00      0.75         6
         128       1.00      1.00      1.00         2
         129       1.00      1.00      1.00         1
         130       1.00      1.00      1.00         2
         131       1.00      1.00      1.00         2
         132       1.00      1.00      1.00         2
         133       1.00      1.00      1.00         8
         134       1.00      1.00      1.00         2
         135       1.00      1.00      1.00         2
         136       0.50      1.00      0.67         2
         137       1.00      1.00      1.00         5
         138       1.00      1.00      1.00         3
         139       1.00      0.75      0.86         4
         140       1.00      1.00      1.00         9
         141       0.90      1.00      0.95        19
         142       1.00      0.67      0.80         3
         145       1.00      1.00      1.00         2
         146       0.00      0.00      0.00         1

    accuracy                           0.91       600
   macro avg       0.86      0.86      0.84       600
weighted avg       0.91      0.91      0.89       600
```

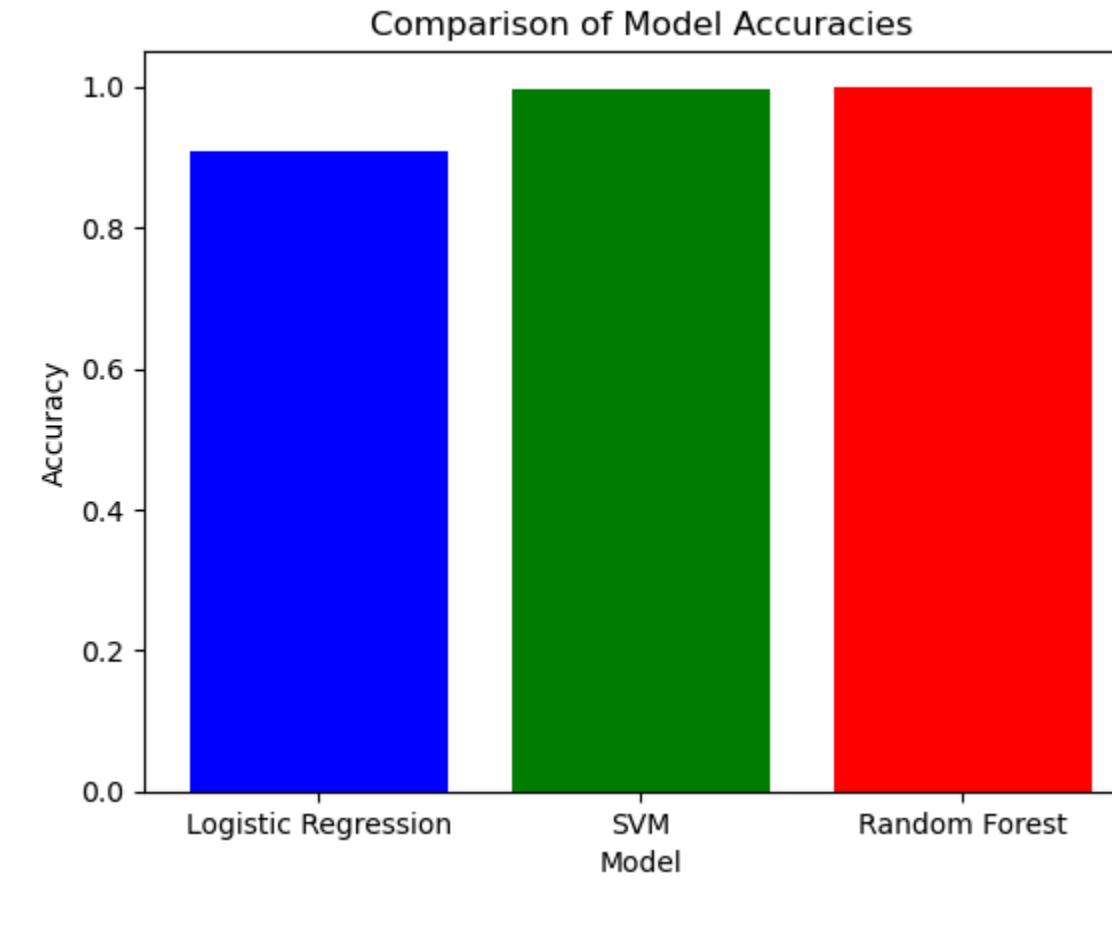
```
C:\Users\User\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1469: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\User\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1469: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\User\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1469: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

```
SVM Accuracy: 0.9966666666666667
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         6
           1       1.00      1.00      1.00         3
           2       1.00      1.00      1.00         9
           3       1.00      1.00      1.00         3
           4       1.00      1.00      1.00        11
           5       1.00      1.00      1.00         2
           6       1.00      1.00      1.00         7
           7       1.00      1.00      1.00         2
           8       1.00      1.00      1.00         2
           9       1.00      1.00      1.00         6
          10       1.00      1.00      1.00         3
          11       1.00      1.00      1.00         2
          12       1.00      1.00      1.00         5
          13       1.00      1.00      1.00         2
          14       1.00      1.00      1.00         3
          15       1.00      1.00      1.00         5
          16       1.00      1.00      1.00         2
          17       1.00      1.00      1.00         5
          18       1.00      1.00      1.00         6
          19       1.00      1.00      1.00         2
          20       1.00      1.00      1.00         8
          21       1.00      1.00      1.00         2
          22       1.00      1.00      1.00         6
          23       1.00      1.00      1.00         7
          24       1.00      1.00      1.00         9
          25       1.00      1.00      1.00         6
          26       1.00      1.00      1.00         3
          27       1.00      1.00      1.00         3
          28       1.00      1.00      1.00         5
          29       1.00      1.00      1.00         2
          30       1.00      1.00      1.00         5
          31       1.00      1.00      1.00        10
          32       1.00      1.00      1.00         5
          33       1.00      1.00      1.00         4
          34       1.00      1.00      1.00         2
          35       1.00      1.00      1.00         2
          36       1.00      1.00      1.00         1
          37       1.00      1.00      1.00         7
          38       1.00      1.00      1.00         1
          39       1.00      1.00      1.00         5
          40       1.00      1.00      1.00         7
          41       1.00      1.00      1.00         1
          42       1.00      1.00      1.00         8
          43       1.00      1.00      1.00         2
          44       1.00      1.00      1.00         6
          45       1.00      1.00      1.00         1
          46       1.00      1.00      1.00         5
          48       1.00      1.00      1.00         1
          49       1.00      1.00      1.00         1
          50       1.00      1.00      1.00         6
          51       1.00      1.00      1.00         8
          52       1.00      1.00      1.00         1
          53       1.00      1.00      1.00         4
          54       1.00      1.00      1.00         4
          57       1.00      1.00      1.00         5
          58       1.00      1.00      1.00         4
          59       1.00      1.00      1.00         1
          61       1.00      1.00      1.00         8
          62       1.00      1.00      1.00         1
          63       1.00      1.00      1.00         5
          64       1.00      1.00      1.00         2
          65       1.00      1.00      1.00         3
          66       1.00      1.00      1.00         1
          67       1.00      1.00      1.00         7
          68       1.00      1.00      1.00         6
          69       1.00      1.00      1.00         4
          70       1.00      1.00      1.00         2
          71       1.00      1.00      1.00         9
          73       1.00      1.00      1.00         5
          74       1.00      1.00      1.00         3
          75       1.00      1.00      1.00         6
          76       1.00      1.00      1.00         5
          77       1.00      1.00      1.00         3
          78       1.00      1.00      1.00         8
          79       1.00      1.00      1.00         1
          80       1.00      1.00      1.00         5
          81       1.00      1.00      1.00         3
          82       1.00      1.00      1.00         1
          83       1.00      1.00      1.00         7
          84       1.00      1.00      1.00         5
          85       1.00      1.00      1.00         4
          86       1.00      1.00      1.00         3
          87       1.00      1.00      1.00         5
          88       1.00      1.00      1.00         5
          89       1.00      1.00      1.00         7
          90       1.00      1.00      1.00         9
          91       1.00      1.00      1.00         5
          92       1.00      1.00      1.00         8
          93       1.00      1.00      1.00         4
          94       1.00      1.00      1.00         4
          95       1.00      1.00      1.00         3
          96       1.00      1.00      1.00         1
          97       1.00      1.00      1.00         4
          99       1.00      1.00      1.00         4
         100       1.00      1.00      1.00         9
         102       1.00      1.00      1.00         2
         103       1.00      1.00      1.00         6
         105       1.00      1.00      1.00         4
         106       1.00      1.00      1.00         3
         107       1.00      1.00      1.00         8
         108       1.00      1.00      1.00         3
         109       0.00      0.00      0.00         2
         110       0.67      1.00      0.80         4
         111       1.00      1.00      1.00         4
         112       1.00      1.00      1.00         6
         113       1.00      1.00      1.00         4
         114       1.00      1.00      1.00         2
         115       1.00      1.00      1.00         4
         116       1.00      1.00      1.00        11
         117       1.00      1.00      1.00         1
         118       1.00      1.00      1.00         1
         119       1.00      1.00      1.00         2
         120       1.00      1.00      1.00         9
         121       1.00      1.00      1.00         3
         122       1.00      1.00      1.00         5
         123       1.00      1.00      1.00         4
         124       1.00      1.00      1.00         8
         125       1.00      1.00      1.00         2
         126       1.00      1.00      1.00         8
         127       1.00      1.00      1.00         6
         128       1.00      1.00      1.00         2
         129       1.00      1.00      1.00         1
         130       1.00      1.00      1.00         2
         131       1.00      1.00      1.00         2
         132       1.00      1.00      1.00         2
         133       1.00      1.00      1.00         8
         134       1.00      1.00      1.00         2
         135       1.00      1.00      1.00         2
         136       1.00      1.00      1.00         2
         137       1.00      1.00      1.00         5
         138       1.00      1.00      1.00         3
         139       1.00      1.00      1.00         4
         140       1.00      1.00      1.00         9
         141       1.00      1.00      1.00        19
         142       1.00      1.00      1.00         3
         145       1.00      1.00      1.00         2
         146       1.00      1.00      1.00         1

    accuracy                           1.00       600
   macro avg       0.99      0.99      0.99       600
weighted avg       0.99      1.00      1.00       600
```

```
C:\Users\User\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1469: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\User\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1469: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\User\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1469: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

```
Random Forest Accuracy: 1.0
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         6
           1       1.00      1.00      1.00         3
           2       1.00      1.00      1.00         9
           3       1.00      1.00      1.00         3
           4       1.00      1.00      1.00        11
           5       1.00      1.00      1.00         2
           6       1.00      1.00      1.00         7
           7       1.00      1.00      1.00         2
           8       1.00      1.00      1.00         2
           9       1.00      1.00      1.00         6
          10       1.00      1.00      1.00         3
          11       1.00      1.00      1.00         2
          12       1.00      1.00      1.00         5
          13       1.00      1.00      1.00         2
          14       1.00      1.00      1.00         3
          15       1.00      1.00      1.00         5
          16       1.00      1.00      1.00         2
          17       1.00      1.00      1.00         5
          18       1.00      1.00      1.00         6
          19       1.00      1.00      1.00         2
          20       1.00      1.00      1.00         8
          21       1.00      1.00      1.00         2
          22       1.00      1.00      1.00         6
          23       1.00      1.00      1.00         7
          24       1.00      1.00      1.00         9
          25       1.00      1.00      1.00         6
          26       1.00      1.00      1.00         3
          27       1.00      1.00      1.00         3
          28       1.00      1.00      1.00         5
          29       1.00      1.00      1.00         2
          30       1.00      1.00      1.00         5
          31       1.00      1.00      1.00        10
          32       1.00      1.00      1.00         5
          33       1.00      1.00      1.00         4
          34       1.00      1.00      1.00         2
          35       1.00      1.00      1.00         2
          36       1.00      1.00      1.00         1
          37       1.00      1.00      1.00         7
          38       1.00      1.00      1.00         1
          39       1.00      1.00      1.00         5
          40       1.00      1.00      1.00         7
          41       1.00      1.00      1.00         1
          42       1.00      1.00      1.00         8
          43       1.00      1.00      1.00         2
          44       1.00      1.00      1.00         6
          45       1.00      1.00      1.00         1
          46       1.00      1.00      1.00         5
          48       1.00      1.00      1.00         1
          49       1.00      1.00      1.00         1
          50       1.00      1.00      1.00         6
          51       1.00      1.00      1.00         8
          52       1.00      1.00      1.00         1
          53       1.00      1.00      1.00         4
          54       1.00      1.00      1.00         4
          57       1.00      1.00      1.00         5
          58       1.00      1.00      1.00         4
          59       1.00      1.00      1.00         1
          61       1.00      1.00      1.00         8
          62       1.00      1.00      1.00         1
          63       1.00      1.00      1.00         5
          64       1.00      1.00      1.00         2
          65       1.00      1.00      1.00         3
          66       1.00      1.00      1.00         1
          67       1.00      1.00      1.00         7
          68       1.00      1.00      1.00         6
          69       1.00      1.00      1.00         4
          70       1.00      1.00      1.00         2
          71       1.00      1.00      1.00         9
          73       1.00      1.00      1.00         5
          74       1.00      1.00      1.00         3
          75       1.00      1.00      1.00         6
          76       1.00      1.00      1.00         5
          77       1.00      1.00      1.00         3
          78       1.00      1.00      1.00         8
          79       1.00      1.00      1.00         1
          80       1.00      1.00      1.00         5
          81       1.00      1.00      1.00         3
          82       1.00      1.00      1.00         1
          83       1.00      1.00      1.00         7
          84       1.00      1.00      1.00         5
          85       1.00      1.00      1.00         4
          86       1.00      1.00      1.00         3
          87       1.00      1.00      1.00         5
          88       1.00      1.00      1.00         5
          89       1.00      1.00      1.00         7
          90       1.00      1.00      1.00         9
          91       1.00      1.00      1.00         5
          92       1.00      1.00      1.00         8
          93       1.00      1.00      1.00         4
          94       1.00      1.00      1.00         4
          95       1.00      1.00      1.00         3
          96       1.00      1.00      1.00         1
          97       1.00      1.00      1.00         4
          99       1.00      1.00      1.00         4
         100       1.00      1.00      1.00         9
         102       1.00      1.00      1.00         2
         103       1.00      1.00      1.00         6
         105       1.00      1.00      1.00         4
         106       1.00      1.00      1.00         3
         107       1.00      1.00      1.00         8
         108       1.00      1.00      1.00         3
         109       1.00      1.00      1.00         2
         110       1.00      1.00      1.00         4
         111       1.00      1.00      1.00         4
         112       1.00      1.00      1.00         6
         113       1.00      1.00      1.00         4
         114       1.00      1.00      1.00         2
         115       1.00      1.00      1.00         4
         116       1.00      1.00      1.00        11
         117       1.00      1.00      1.00         1
         118       1.00      1.00      1.00         1
         119       1.00      1.00      1.00         2
         120       1.00      1.00      1.00         9
         121       1.00      1.00      1.00         3
         122       1.00      1.00      1.00         5
         123       1.00      1.00      1.00         4
         124       1.00      1.00      1.00         8
         125       1.00      1.00      1.00         2
         126       1.00      1.00      1.00         8
         127       1.00      1.00      1.00         6
         128       1.00      1.00      1.00         2
         129       1.00      1.00      1.00         1
         130       1.00      1.00      1.00         2
         131       1.00      1.00      1.00         2
         132       1.00      1.00      1.00         2
         133       1.00      1.00      1.00         8
         134       1.00      1.00      1.00         2
         135       1.00      1.00      1.00         2
         136       1.00      1.00      1.00         2
         137       1.00      1.00      1.00         5
         138       1.00      1.00      1.00         3
         139       1.00      1.00      1.00         4
         140       1.00      1.00      1.00         9
         141       1.00      1.00      1.00        19
         142       1.00      1.00      1.00         3
         145       1.00      1.00      1.00         2
         146       1.00      1.00      1.00         1

    accuracy                           1.00       600
   macro avg       1.00      1.00      1.00       600
weighted avg       1.00      1.00      1.00       600
```

In [17]:
```python
import matplotlib.pyplot as plt

# Assuming that the accuracy of the models are stored in the variables `accuracy`, `svm_accuracy`, and `rf_accuracy`

# Create a list of the accuracies
accuracies = [accuracy, svm_accuracy, rf_accuracy]

# Create a list of the model names
models = ['Logistic Regression', 'SVM', 'Random Forest']

# Create a bar plot
plt.bar(models, accuracies, color=['blue', 'green', 'red'])

# Add title and labels
plt.title('Comparison of Model Accuracies')
plt.xlabel('Model')
plt.ylabel('Accuracy')

# Show the plot
plt.show()
```



Testing the model

In [18]:
```python
# Let's say we have a new individual with the following skills, responsibilities, and job description
individual_skills = "Talent sourcing Interviewing Onboarding Recruiting"
individual_responsibilities = "Lead talent acquisition efforts, including sourcing"
individual_job_description = "Talent Acquisition Managers oversee the recruitment"

# We need to preprocess the text and convert it to the format that our model expects
individual_combined_text = individual_job_description + ' ' + individual_responsibilities + ' ' + individual_skills
individual_combined_text = preprocess_text(individual_combined_text)

# Convert the text to TF-IDF vectors
individual_X = tfidf_vectorizer.transform([individual_combined_text]).toarray()

# Use the Random Forest model to predict the job title
individual_y_pred = rf_model.predict(individual_X)

# The model's prediction is in the form of a label-encoded integer, so we need to transform it back to the original job title
individual_job_title_pred = label_encoder.inverse_transform(individual_y_pred)

# Print the predicted job title
print(f"The predicted job title for the individual is: {individual_job_title_pred[0]}")
```

```
The predicted job title for the individual is: Human Resources Manager
```

In [ ]: