```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
```

## Connect to Google Drive

```
!rm -rf /root/.config/Google/DriveFS
!rm -rf /root/.config/Google/Drive/
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
File_1 = pd.read_csv('/content/drive/My Drive/Resume data set/clean_resume_data.csv')
```

## Import Clean DataSet

```
File_1.head()
```

| | ID | Category | Feature |
|---|---|---|---|
| **0** | 16852973 | HR | hr administrator marketing associate hr admini... |
| **1** | 22323967 | HR | hr specialist hr operations summary media prof... |
| **2** | 33176873 | HR | hr director summary years experience recruitin... |
| **3** | 27018550 | HR | hr specialist summary dedicated driven dynamic... |
| **4** | 17812897 | HR | hr manager skill highlights hr skills hr depar... |

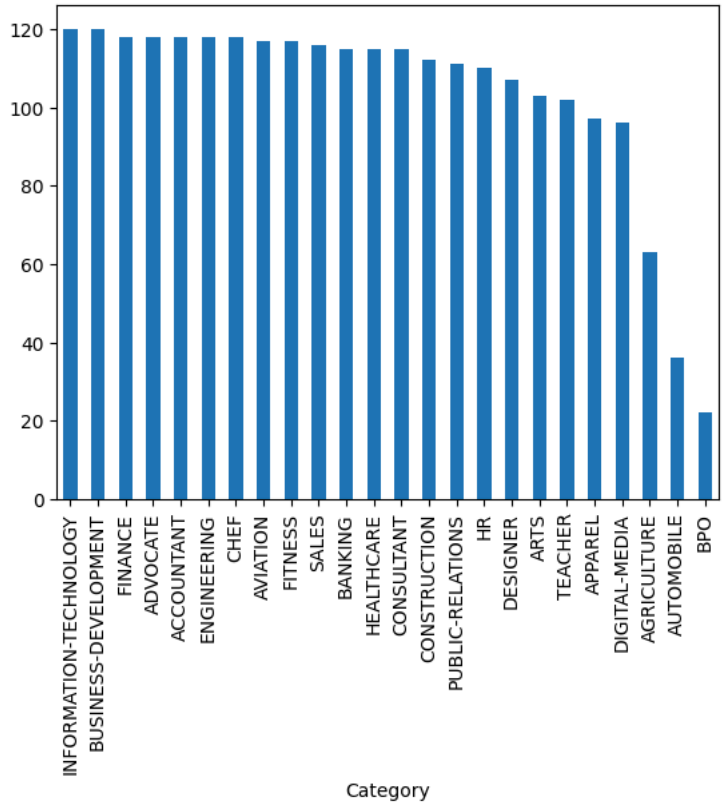
```
File_1.shape
```

```
(2484, 3)
```

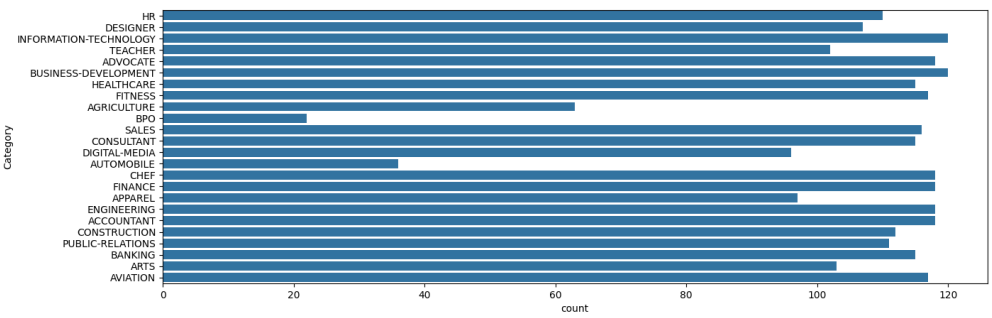
## Exploratory Data Analysis

```
File_1['Category'].value_counts().plot(kind='bar')
```

<Axes: xlabel='Category'>




```
plt.figure(figsize=(15,5))
sns.countplot(File_1['Category'])
plt.show()
```




```
counts = File_1['Category'].value_counts()
labels = File_1['Category'].unique()
plt.figure(figsize=(15,10))

plt.pie(counts,labels=labels,autopct='%1.1f%%',shadow=True, colors=plt.cm.plasma(np.linspace(0,1,3)))
plt.show()
```

Show hidden output

## Balance DataSet

```
from sklearn.utils import resample

# Define the maximum count among all categories
max_count = File_1['Category'].value_counts().max()

# Resample each category to match the maximum count
balanced_data = []
for category in File_1['Category'].unique():
    category_data = File_1[File_1['Category'] == category]
    if len(category_data) < max_count:
        # Perform oversampling for categories with fewer samples
        balanced_category_data = resample(category_data, replace=True, n_samples=max_count, random_state=42)
    else:
        # Perform undersampling for categories with more samples
        balanced_category_data = resample(category_data, replace=False, n_samples=max_count, random_state=42)
    balanced_data.append(balanced_category_data)

# Concatenate the balanced data for all categories
balanced_df = pd.concat(balanced_data)
```

```
balanced_df['Category'].value_counts()
```

```
Category
HR                        120
DESIGNER                  120
ARTS                      120
BANKING                   120
PUBLIC-RELATIONS          120
CONSTRUCTION              120
ACCOUNTANT                120
ENGINEERING               120
APPAREL                   120
FINANCE                   120
CHEF                      120
AUTOMOBILE                120
DIGITAL-MEDIA             120
CONSULTANT                120
SALES                     120
BPO                       120
AGRICULTURE               120
FITNESS                   120
HEALTHCARE                120
BUSINESS-DEVELOPMENT      120
ADVOCATE                  120
TEACHER                   120
INFORMATION-TECHNOLOGY    120
AVIATION                  120
Name: count, dtype: int64
```

```
balanced_df.isnull().sum()
```

```
ID          0
Category    0
Feature     1
dtype: int64
```

```
balanced_df.dropna(inplace=True)
```

## Train & Test data

```
X = balanced_df['Feature']
y = balanced_df['Category']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

## Encoding (TF-IDF)

```
X_train
```

```
228     director information technology professional p...
1557    senior finance manager summary highly driven f...
1194    consultant summary media planner media buyer r...
916     adult education teacher accomplishments cal po...
2150    banking summary high energy manager successful...
                              ...
```

```
1328    director executive profile forward thinker exp...
973     bpo team leader summary seeking provide top qu...
979     manager administration facilities executive su...
1027    pmc psm sales manager sales associate career f...
793     fitness specialist summary energetic personal ...
Name: Feature, Length: 2303, dtype: object
```

```
tfidf_vectorizer = TfidfVectorizer()
X_train_tfidf = tfidf_vectorizer.fit_transform(X_train)
X_test_tfidf = tfidf_vectorizer.transform(X_test)
```

## Train Random Forest Classifier

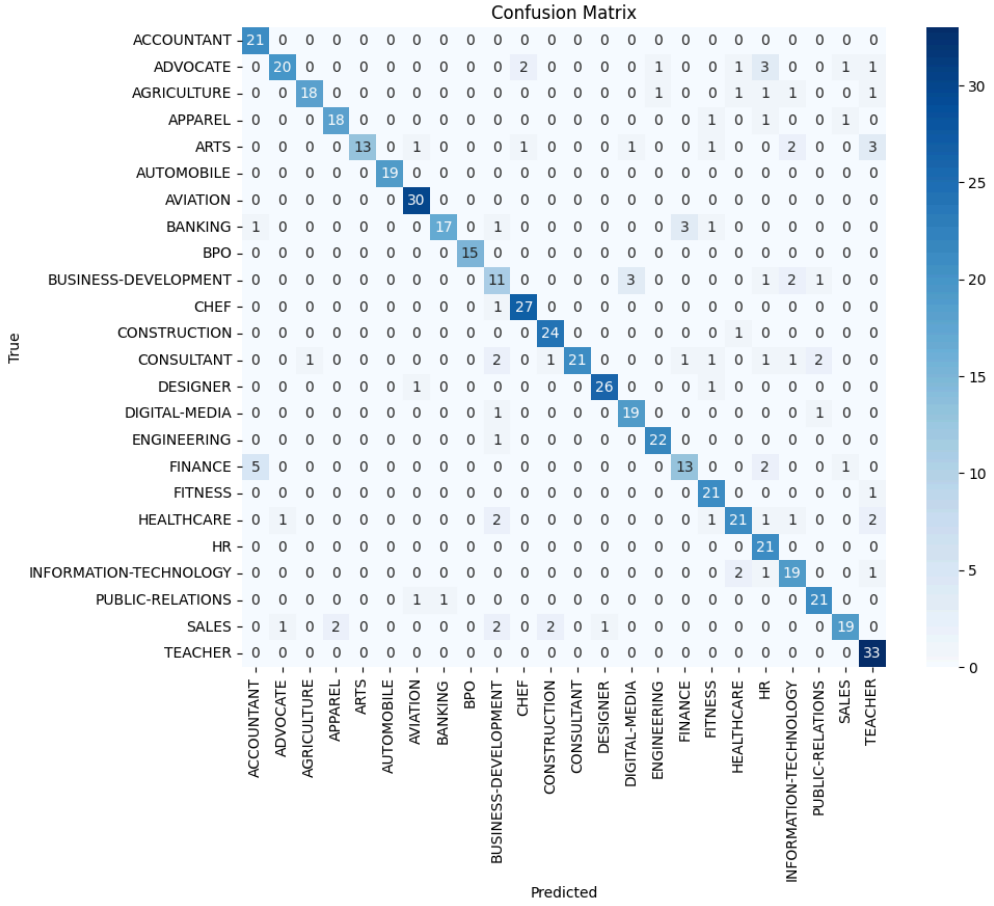
```
from sklearn.metrics import classification_report

rf_classifier = RandomForestClassifier()
rf_classifier.fit(X_train_tfidf, y_train)

# Step 4: Accuracy Evaluation
y_pred = rf_classifier.predict(X_test_tfidf)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
print(classification_report(y_test,y_pred))
# Confusion Matrix
conf_matrix = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(10, 8))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=rf_classifier.classes_, yticklabels=rf_classifier.classes_)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')
plt.show()
```

Accuracy: 0.8489583333333334

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| ACCOUNTANT | 0.78 | 1.00 | 0.88 | 21 |
| ADVOCATE | 0.91 | 0.69 | 0.78 | 29 |
| AGRICULTURE | 0.95 | 0.78 | 0.86 | 23 |
| APPAREL | 0.90 | 0.86 | 0.88 | 21 |
| ARTS | 1.00 | 0.59 | 0.74 | 22 |
| AUTOMOBILE | 1.00 | 1.00 | 1.00 | 19 |
| AVIATION | 0.91 | 1.00 | 0.95 | 30 |
| BANKING | 0.94 | 0.74 | 0.83 | 23 |
| BPO | 1.00 | 1.00 | 1.00 | 15 |
| BUSINESS-DEVELOPMENT | 0.52 | 0.61 | 0.56 | 18 |
| CHEF | 0.90 | 0.96 | 0.93 | 28 |
| CONSTRUCTION | 0.89 | 0.96 | 0.92 | 25 |
| CONSULTANT | 1.00 | 0.68 | 0.81 | 31 |
| DESIGNER | 0.96 | 0.93 | 0.95 | 28 |
| DIGITAL-MEDIA | 0.83 | 0.90 | 0.86 | 21 |
| ENGINEERING | 0.92 | 0.96 | 0.94 | 23 |
| FINANCE | 0.76 | 0.62 | 0.68 | 21 |
| FITNESS | 0.78 | 0.95 | 0.86 | 22 |
| HEALTHCARE | 0.81 | 0.72 | 0.76 | 29 |
| HR | 0.66 | 1.00 | 0.79 | 21 |
| INFORMATION-TECHNOLOGY | 0.73 | 0.83 | 0.78 | 23 |
| PUBLIC-RELATIONS | 0.84 | 0.91 | 0.87 | 23 |
| SALES | 0.86 | 0.70 | 0.78 | 27 |
| TEACHER | 0.79 | 1.00 | 0.88 | 33 |
| accuracy | | | 0.85 | 576 |
| macro avg | 0.86 | 0.85 | 0.85 | 576 |
| weighted avg | 0.86 | 0.85 | 0.85 | 576 |



```
#Clean the resume
import re
def cleanResume(txt):
    cleanText = re.sub('http\S+\s', ' ', txt)
    cleanText = re.sub('RT|cc', ' ', cleanText)
    cleanText = re.sub('#\S+\s', ' ', cleanText)
    cleanText = re.sub('@\S+', '  ', cleanText)
    cleanText = re.sub('[%s]' % re.escape("""!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~"""), ' ', cleanText)
    cleanText = re.sub(r'[^\x00-\x7f]', ' ', cleanText)
    cleanText = re.sub('\s+', ' ', cleanText)
    return cleanText


# Prediction and Category Name
def predict_category(resume_text):
    resume_text= cleanResume(resume_text)
    resume_tfidf = tfidf_vectorizer.transform([resume_text])
    predicted_category = rf_classifier.predict(resume_tfidf)[0]
    return predicted_category
```

```
#Example For the same
resume_file = """hr assistant summary hard worker dedicated detailed oriented experienced organized highly motivated many attributes possess ind
"""

predicted_category = predict_category(resume_file)
print("Predicted Category:", predicted_category)
```

```
Predicted Category: HR
```

```
#one more example
# Example Usage
resume_file = """Objective:
A creative and detail-oriented Designer with a passion for visual communication and brand identity seeking opportunities to leverage design skil

Education:
- Bachelor of Fine Arts in Graphic Design, XYZ College, GPA: 3.7/4.0
- Diploma in Web Design, ABC Institute, GPA: 3.9/4.0

Skills:
- Proficient in Adobe Creative Suite (Photoshop, Illustrator, InDesign)
- Strong understanding of typography, layout, and color theory
- Experience in both print and digital design
- Ability to conceptualize and execute design projects from concept to completion
- Excellent attention to detail and time management skills

Experience:
Graphic Designer | XYZ Design Studio
- Created visually appealing graphics for various marketing materials, including brochures, flyers, and social media posts
- Collaborated with clients to understand their design needs and deliver creative solutions that align with their brand identity
- Worked closely with the marketing team to ensure consistency in brand messaging across all platforms

Freelance Designer
- Designed logos, branding materials, and website layouts for small businesses and startups
- Managed multiple projects simultaneously while meeting tight deadlines and maintaining quality standards
- Established and maintained strong client relationships through clear communication and exceptional service
```