notebook

January 19, 2023

1 Fraudulent Job Posting Prediction

1.1 Business Problem

According to the FBI's Internet Crime Complaint Center (IC3), 16,012 people reported being victims of employment scams in 2020, with losses totaling more than $59 million. Fake Job Scams have existed for a long time but technology has made this scam easier and more lucrative. Cyber criminals now pose as legitimate employers by spoofing company websites and posting fake job openings on popular online job boards. They conduct false interviews with unsuspecting applicant victims, then request PII and/or money from these individuals. The PII can be used for any number of nefarious purposes, including taking over the victims' accounts, opening new financial accounts, or using the victims' identity for another deception scam (such as obtaining fake driver's licenses or passports).

Criminals first spoof a legitimate company's website by creating a domain name similar in appearance to a legitimate company. Then they post fake job openings on popular job boards that direct applicants to the spoofed sites. Applicants can apply on the spoofed company websites or directly on the job boards. Applicants are contacted by email to conduct an interview using a teleconference application. According to victims, cyber criminals impersonate personnel from different departments, including recruiters, talent acquisition, human resources, and department managers. The average reported loss was nearly $3,000 per victim, in addition to damage to the victims' credit scores.

1.1.1 Project Task

- Perform Exploratory Data Analysis on the dataset to identify interesting insights from this dataset.
- Create classification model that uses text data features and meta-features and predict which job description are fraudulent or real.
- Identify key traits/features (words, entities, phrases) of job descriptions which are fraudulent in nature.
- Run a contextual embedding model to identify the most similar job descriptions.

1.1.2 Dataset:

This dataset contains 18K job descriptions out of which about 800 are fake. The data consists of both textual information and meta-information about the jobs. There are 17880 rows * 18 columns.

- job_id: int 64, 17880
- title: object, 17880
- location: object, 17534, missing value
- department: object, 6333, missing value
- salaray_range: object, 2868, missing value
- company_profile: object, 14572, missing value
- description, object, 17879, only 1 missing value
- requirements, object 15185, missing value
- benefits, object, 10670, missing value
- telecommuting, int64, 17880
- has_company_logo, int64,17880
- has_questions, int 64,17880
- employment_type, object, 14409,missing value
- required_experience,object, 10830,missing value

- required_education, object, 9775, missing value
- industry, object, 12977, missing value
- function, object,11425, missing value
- fraudulent, int64, 17880, '0' for real, '1' for fraud, target value, 4.84%, imbalanced dataset.

[24]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import CountVectorizer
import plotly.express as px
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator

# Nlp library
import re
import nltk
from nltk.corpus import stopwords
import nltk as nlp
from sklearn.feature_extraction.text import CountVectorizer

# sklearn Library
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import r2_score
from sklearn.metrics import confusion_matrix, classification_report,␣
↪plot_confusion_matrix
from sklearn.metrics import accuracy_score, f1_score, recall_score,␣
↪precision_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import LogisticRegression
from xgboost import XGBClassifier
from sklearn.metrics import explained_variance_score

#Tenserflow Library
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Activation, Dropout
from keras.layers import LSTM
from tensorflow.keras.layers import Embedding,  Bidirectional
from tensorflow.keras.preprocessing.sequence import pad_sequences

import warnings
warnings.filterwarnings("ignore")
```

```
2023-01-18 19:17:08.357555: W
tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load
```

```
dynamic library 'libcudart.so.11.0'; dlerror: libcudart.so.11.0: cannot open
shared object file: No such file or directory
2023-01-18 19:17:08.357580: I tensorflow/stream_executor/cuda/cudart_stub.cc:29]
Ignore above cudart dlerror if you do not have a GPU set up on your machine.
```

[2]:
```
data = pd.read_csv('fake_job_postings.csv')
data.head()
```

[2]:
```
   job_id  … fraudulent
0       1  …          0
1       2  …          0
2       3  …          0
3       4  …          0
4       5  …          0

[5 rows x 18 columns]
```

[3]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17880 entries, 0 to 17879
Data columns (total 18 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   job_id               17880 non-null  int64
 1   title                17880 non-null  object
 2   location             17534 non-null  object
 3   department           6333 non-null   object
 4   salary_range         2868 non-null   object
 5   company_profile      14572 non-null  object
 6   description          17879 non-null  object
 7   requirements         15185 non-null  object
 8   benefits             10670 non-null  object
 9   telecommuting        17880 non-null  int64
 10  has_company_logo     17880 non-null  int64
 11  has_questions        17880 non-null  int64
 12  employment_type      14409 non-null  object
 13  required_experience  10830 non-null  object
 14  required_education   9775 non-null   object
 15  industry             12977 non-null  object
 16  function             11425 non-null  object
 17  fraudulent           17880 non-null  int64
dtypes: int64(5), object(13)
memory usage: 2.5+ MB
```

[4]:
```
data.describe()
```

[4]:
```
             job_id  telecommuting  …  has_questions    fraudulent
count  17880.000000   17880.000000  …   17880.000000  17880.000000
mean    8940.500000       0.042897  …       0.491723      0.048434
std     5161.655742       0.202631  …       0.499945      0.214688
min        1.000000       0.000000  …       0.000000      0.000000
25%     4470.750000       0.000000  …       0.000000      0.000000
50%     8940.500000       0.000000  …       0.000000      0.000000
75%    13410.250000       0.000000  …       1.000000      0.000000
max    17880.000000       1.000000  …       1.000000      1.000000

[8 rows x 5 columns]
```

[5]:
```
data.shape
```

[5]:
```
(17880, 18)
```

[6]:
```
data.isnull().sum()
```

[6]:
```
job_id                     0
title                      0
location                 346
department             11547
salary_range           15012
company_profile         3308
description                1
requirements            2695
benefits                7210
telecommuting              0
has_company_logo           0
has_questions              0
employment_type         3471
required_experience     7050
required_education      8105
industry                4903
function                6455
fraudulent                 0
dtype: int64
```

[7]:
```
data.description[0]
```

[7]:
```
'Food52, a fast-growing, James Beard Award-winning online food community and
crowd-sourced and curated recipe hub, is currently interviewing full- and part-
time unpaid interns to work in a small team of editors, executives, and
developers in its New York City headquarters.Reproducing and/or repackaging
existing Food52 content for a number of partner sites, such as Huffington Post,
Yahoo, Buzzfeed, and more in their various content management systemsResearching
blogs and websites for the Provisions by Food52 Affiliate ProgramAssisting in
```

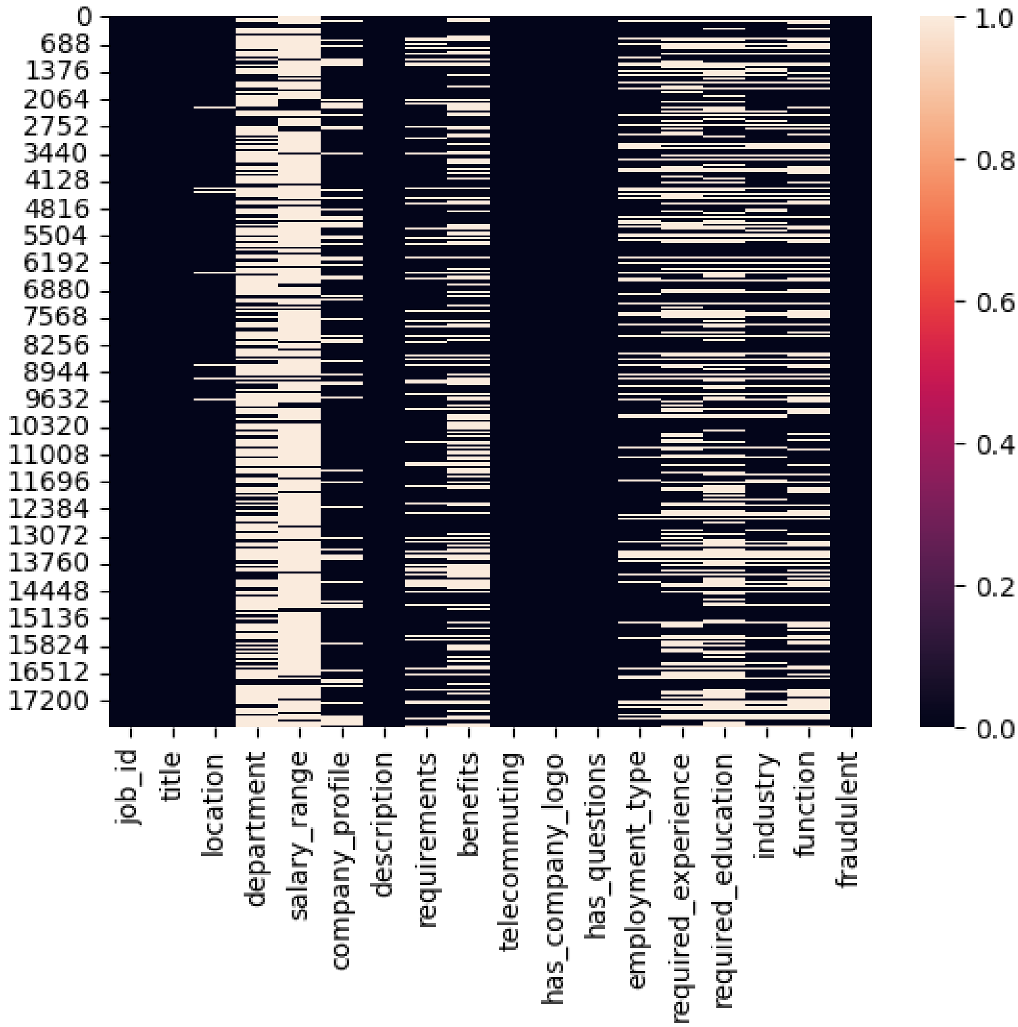
```
day-to-day affiliate program support, such as screening affiliates and assisting
in any affiliate inquiriesSupporting with PR &amp; Events when neededHelping
with office administrative work, such as filing, mailing, and preparing for
meetingsWorking with developers to document bugs and suggest improvements to the
siteSupporting the marketing and executive staff'
```

[8]:
```
data.company_profile[0]
```

[8]:
```
"We're Food52, and we've created a groundbreaking and award-winning cooking
site. We support, connect, and celebrate home cooks, and give them everything
they need in one place.We have a top editorial, business, and engineering team.
We're focused on using technology to find new and better ways to connect people
around their specific food interests, and to offer them superb, highly curated
information about food and cooking. We attract the most talented home cooks and
contributors in the country; we also publish well-known professionals like Mario
Batali, Gwyneth Paltrow, and Danny Meyer. And we have partnerships with Whole
Foods Market and Random House.Food52 has been named the best food website by the
James Beard Foundation and IACP, and has been featured in the New York Times,
NPR, Pando Daily, TechCrunch, and on the Today Show.We're located in Chelsea, in
New York City."
```

1.2 Data Validation and Data Wrangling

[9]:
```
sns.heatmap(data.isnull())
plt.show()

# Insight: Department, salary_range, company_profile, requirements, benefits,
 ↪employment_type, required_experience, required_education, industry, function
 ↪columns have very high percentage of missing value
```



[10]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17880 entries, 0 to 17879
Data columns (total 18 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   job_id               17880 non-null  int64
 1   title                17880 non-null  object
 2   location             17534 non-null  object
 3   department           6333 non-null   object
 4   salary_range         2868 non-null   object
 5   company_profile      14572 non-null  object
 6   description          17879 non-null  object
 7   requirements         15185 non-null  object
```

```
 8   benefits             10670 non-null  object
 9   telecommuting        17880 non-null  int64
 10  has_company_logo     17880 non-null  int64
 11  has_questions        17880 non-null  int64
 12  employment_type      14409 non-null  object
 13  required_experience  10830 non-null  object
 14  required_education   9775 non-null   object
 15  industry             12977 non-null  object
 16  function             11425 non-null  object
 17  fraudulent           17880 non-null  int64
dtypes: int64(5), object(13)
memory usage: 2.5+ MB
```

[11]:
```
data.location.value_counts()
```

[11]:
```
GB, LND, London          718
US, NY, New York         658
US, CA, San Francisco    472
GR, I, Athens            464
US, ,                    339
                        …
GB, SFK, Leiston           1
GB, LND, Hammersmith       1
US, WA, Seattle            1
BE                         1
GB, WSX, Chichester        1
Name: location, Length: 3105, dtype: int64
```

[12]:
```
data.location.fillna('Unknown',inplace = True)
```

[13]:
```
data.location.value_counts()
```

[13]:
```
GB, LND, London          718
US, NY, New York         658
US, CA, San Francisco    472
GR, I, Athens            464
Unknown                  346
                        …
GB, SFK, Leiston           1
GB, LND, Hammersmith       1
US, WA, Seattle            1
BE                         1
GB, WSX, Chichester        1
Name: location, Length: 3106, dtype: int64
```

[14]:
```
data.department.value_counts()
```

[14]:
```
Sales                                             551
Engineering                                       487
Marketing                                         401
Operations                                        270
IT                                                225
                                                 ...
Commercial Management / Contract Management         1
Exec                                                1
Marcomm                                             1
CRM                                                 1
Hospitality                                         1
Name: department, Length: 1337, dtype: int64
```

[15]:
```python
data.department.fillna("Unknown", inplace = True)
```

[16]:
```python
data.salary_range.value_counts()
```

[16]:
```
0-0            142
40000-50000     66
30000-40000     55
25000-30000     37
45000-67000     37
              ...
15-25            1
60-75            1
27500-36000      1
20-22            1
3700-3800        1
Name: salary_range, Length: 874, dtype: int64
```

[17]:
```python
data.salary_range.fillna("Unknown", inplace = True)
```

[18]:
```python
data.company_profile.value_counts()
```

[18]:
```
We help teachers get safe &amp; secure jobs abroad :)
726
We Provide Full Time Permanent Positions for many medium to large US companies.
We are interested in finding/recruiting high quality candidates in IT,
Engineering, Manufacturing and other highly technical and non-technical jobs.
674
Novitex Enterprise Solutions, formerly Pitney Bowes Management Services,
delivers innovative document and communications management solutions that help
companies around the world drive business process efficiencies, increase
productivity, reduce costs and improve customer satisfaction. For almost 30
years, clients have turned to us to integrate and optimize their enterprise-wide
business processes to empower employees, increase productivity and maximize
results. As a trusted partner, we continually focus on delivering secure,
```

technology-enabled document and communications solutions that improve our clients' work processes, enhance their customer interactions and drive growth.
574
Established on the principles that full time education is not for everyone Spectrum Learning is made up of a team of passionate consultants with the drive for putting people who wish to grow themselves through education whilst working into long term and relevant job roles.We also are official re-sellers for The Institute of Recruiters/ Study Course professional courses in HR Practice, In-House Recruitment and Agency RecruitmentIt is our mission to help anyone wishing to pursue an apprenticeship onto the right qualification and into the right job.We work closely with both the candidate and the employer to ensure when the learner is enrolled they are at the start of a long and successful career.We have great relationships with a number of national training providers to ensure we can cover any apprenticeship available.
450
Applied Memetics LLC is a professional services company dedicated to integrating and delivering best practice communication and information solutions in pre-conflict, conflict, or post-conflict areas. The world has changed: 'always on' brands require a new way of thinking to engage and manage their consumers. Our purpose is to inspire original thinking through a deeper understanding of technology and human behaviour. From strategy through to implementation, our teams of connected specialists - all experts in their respective fields - work together to help our clients maximise the opportunities created by the changing digital world and create a multi faceted digital strategy through to implementation.Our work explores a new model of journalism that is based around a global story - in this case, the struggle for human rights and democracy around the world. Our goal is to build a better user experience of these stories by adding context to content, using the latest digital tools of the day. Over time, we hope to add greater clarity, deeper understanding, and more sustained engagement to the conversations surrounding global events. As such, our content is transcribed and translated into English for broadcast to a global audience.
185

…

ISBS Hellas is a modern and growing company developing innovative websites and applications.
1
T USEUM
(#URL_04ac7232026f06ca6a32948470abce692e6921c271b55005ae682a6dec34e345#)
«T » 11,000
P E Γ
96 .H USEUM LTD start-up
Λ ,
.
1
Founded in Athens, Greece in 1995, Relational Technology SA has evolved over the years to an international software developer & integrator with offices in Albania and Romania and a client base all over Europe, Middle East and North

```
Africa. Its strategy is focused on expansion to markets outside Greece such as
SE Europe, Middle East and North Africa. The company has an average annual
revenue of € 12.000.000 the last three years and employs an average of 130
persons.Relational offers leading IT solutions for Process Automation, Business
Process Management, Business Intelligence, Data Warehouse, Data Mining, Data
Collection, Management Information Systems and reporting, as well as
product/project related services (i.e. requirement analysis, design of
architectural and technical solution, installation, configuration, project
management and technical support).Relational develops mission critical software
assets for Financial Institutions, Government, Retail, Telecoms and SMEs, while
in addition represents actively and exclusively a number of International
software vendors and their respective product portfolios, namely SAP AG,
Informatica SA, Microsoft, BMC Software and UC4.
1
Focus Camera is a family owned and operated nationwide specialty on-line
retailer.  Our roots are in a brick-and-mortar retail, operating in the same New
York neighborhood location for over 40 years. We have taken our consumer-
oriented philosophy that has served us well in our local community, and expanded
it to serve customers nationwide.Customer Service is our #1 priority. We make
customers; not just sales. Our customers are our investment. "Treat every
customer as if they sign your paycheck…because they do." This has been our
credo, and it is for this reason that many of our orders are from repeat
customers.We carry a wide range of all brand-name photographic products,
including digital cameras and camcorders, digital frames, and photo printers.
In addition, we carry consumer electronics, optics equipment such as binoculars
and telescopes, and some kitchen and home appliances.Our prices are very
competitive. We sell only top-quality merchandise, and our customer service is
second to none. Join our family of satisfied customers, and you will be glad you
did. In fact, our best advertisers are our satisfied customers, more than
400,000 strong, and growing daily.
1
Stories by RELFrom 5,000 feet, we look like a media company. If you look close,
our greatest strength is producing video. If you find our sweet spot, we are
really into agriculture. If you pin us down, we value quality and work hard to
deliver as much as we can. If we had a wish, we would be doing this (with an
unlimited budget and no deadlines). If you sum us up, we make stories.Produce
the story.Every project starts as an idea and ends as a finished product.
Producing starts someplace after "idea" and ends when the client is happy with
the finished product. Everything in-between is a variable. Our focus is taking
that idea and helping our clients get it to an end product they can be proud
of.Let’s get our hands dirty.We believe in hard work, but we also believe in
working smart. If we have learned anything in nearly 20 years…you need to have
processes, technology and workflows…that can be managed and operated
efficiently, that are client friendly and are able handle most anything thrown
at them. You will understand it when you see it. The tour is free.
1
Name: company_profile, Length: 1709, dtype: int64
```

[19]:
```python
data.company_profile.fillna("", inplace = True)
```

[20]:
```python
data.requirements.value_counts()
```

[20]:
```
University degree required. TEFL / TESOL / CELTA or teaching experience
preferred but not necessaryCanada/US passport holders only
410
University degree required. TEFL / TESOL / CELTA or teaching experience
preferred but not necessaryPositive attitude required. Canada/US passport
holders only
163
16-18 year olds only due to government funding.Full time availability.
117
Minimum Requirements:Minimum of 6 months customer service related experience
requiredHigh school diploma or equivalent (GED) requiredPreferred
Qualifications:Keyboarding and windows environment PC skills required (Word,
Excel and PowerPoint preferred)Experience running mail posting equipment a
plusExcellent communication skills both verbal and writtenLifting up to 55 lbs
with or without accommodationsWillingness and availability to work additional
hours if assignedWillingness to submit to a pre-employment drug screening and
criminal background checkAbility to effectively work individually or in a team
environmentCompetency in performing multiple functional tasksAbility to meet
employer's attendance policy
80
University degree required. TEFL / TESOL / CELTA, and/or teaching experience
preferredCanada/US passport holders only
42
                                                                   …
About You2 or more years managing analytics infrastructure in a startup
environment.Highly proficient in SQL and a scripting language of choice (PHP,
Python, Ruby)Deep understanding of data structures and schema design.Expertise
in low-latency data stores (Vertica, Redshift)Experience with developing,
maintaining and/or supporting business intelligence / reporting tools.Bias
towards most efficient solution for the problem (e.g. experience and willingness
to rely on third party tools rather than developing in-house).Familiarity with
commonly-used third party tools for analytics and event-driven marketing (Google
Analytics, Kissmetrics, Mixpanel,
#URL_48c8e248f7ad35fdccda4a20a3f3f3951f2624a277ba771de21dc8cb3ad211d0#,
Optimizely, Tableau).Understanding of common user acquisition / retention /
revenue metrics in a SaaS company.ResponsibilitiesFully manage our data
pipeline: instrumentation of our website, data collection, ETL, low latency
storageOwn data quality and integrityMaintain and support querying / business
reporting toolsExpect to spend 25% of one's time consuming and reporting data
and insights (not only to drive the business forward, but also to understand how
to make data analysis automated or easier for others in the company).      1
Understands what a startup is and is willing to work in such
environmentUnderstanding of the sales processIs passionate about growing
```

```
businesses and closing deals with clientsUnderstands digital products and how
they are sold and grownHas a clear interest in marketing and
technologyPreference for candidates finishing / who've finished Marketing, Sales
or related degrees.Fluent in EnglishCan work in a fast-paced environment and in
a small team
1
RequirementsMarket knowledge about the UK real estate sector having lived and/or
studied in the UK beforeFluency in written and spoken English and preferably
another European languageInternational experience, having lived worked or
studied outside your home countryPast sales experience is good but more
important is your motivation &amp; energyAvailable to start within a month
1
Job requirements and essential functions: Able to type a minimum of 45 WPM
Computer savvy Basic knowledge of Microsoft Office (especially Word and Excel)
Time management skills Hard working Minimum of 40 hours per week Overtime
available (and occasionally required) Willing to work legal holidays and
weekends as required Comfortable in an open office environment Applicants with
college degrees and/or college students preferred but not required
1
1. Must be fluent in the latest versions of Corel &amp; Adobe CC (Esp.
Photoshop, Illustrator &amp; Indesign) 2. Have a strong interest in
interactive/interface design 3. Understand color theory, typography, composition
and photo retouching 4. Be able to take design direction 5. Must think
creatively and step outside of the norm 6. Be willing to put in the extra time
and effort on projects 7. Eager to learn and have a great attitude 8. Be self-
sufficient and able to figure problems out on your own
1
Name: requirements, Length: 11968, dtype: int64
```

[21]:
```python
data.description.fillna("", inplace= True)
```

[22]:
```python
data.requirements.fillna("", inplace = True)
```

[23]:
```python
data.benefits.value_counts()
```

[23]:
```
See job description
726
Career prospects.
158
CSD offers a competitive benefits package for full-time employees. For a full
list of benefits and perks, please visit the career page.Communication Service
for the Deaf, Inc. is an Equal Opportunity Affirmative Action Employer and drug
free and tobacco free workplace. All qualified applicants will receive
consideration for employment without regard to race, color, religion, sex, or
national origin, including individuals with a disability and protected veterans.
70
Our company offers a competitive salary plus BONUSES as well as a comprehensive
```

benefits package to our full-time employees including:40 vacation hours after 6 months of employment, 80 vacation hours after 1 year of employment6 paid holidays as well as an anniversary holiday benefitPaid personal and sick leave after 90 days of employmentHealth, dental, life, and disability insurance as well as AFLAC supplemental insuranceA 401K plan with a company match after six months of employment, however, we have quarterly enrollment periods.
60
Plenty of perksAs well as the opportunity to solve complex problems in this exciting new era of big data, here’s what we offer:Realistic performance related bonusesGenerous equity options mean you’ll own a piece of the pieExcellent health and dental insurance packagesA relaxed approach to time off and enough holidays to see several corners of the worldFridge fully stocked with healthy snacks and the ultimate espresso machine for your java fixA competitive office where we play foosball, football, scrabble, go-karting… you name it, we’ll play itThank Qubit it’s Friday - we have lots of creative ways to let off steam at the end of the weekPlenty of opportunities for training and development 58
…
Salary:Based on Qualifications and Experience
1
Novation offers an empowered work environment that encourages creativity, initiative and professional growth and provides a competitive salary and benefits package. Novation is an Equal Employment Opportunity/Affirmative Action Employer and maintains a Drug-Free Workplace. We are fully committed to employing a diverse workforce and creating an inclusive work environment that embraces everyone’s unique contributions, experiences and values. Please apply on our website for consideration.
1
Benefits • Competitive salary • Vehicle allowance • Mobile phone and Laptop • Uncapped commission • Career development opportunities All successful candidates will be required to complete a Ministry of Justice criminal record check and drug screen. If you can provide innovative solutions which meet the needs for this business and have a track record of establishing relationships at all levels with proven sales history. I want to hear from you. Immediate start. Please submit your application in strict confidence
1
This is an opportunity to work with one of the most exciting high tech companies globally that is turning science fiction into an accessible technology. We hire the best in the wireless power technology field globally. If you are someone highly motivated in developing your career in the power electronics industry and wish to become a subject matter expert in this field, this is a great opportunity for you to advance your career. Only overseas applicants with experience from relevant sectors (Induction Power, Wireless Power, Power Electronics) will be considered.
1
Competitive salary (compensation will be based on experience) Casual attire At Nemsia Studios you are assured of a pleasant, enthusiastic, fast paced work environment with a lot of great people who love what they do!

```
1
Name: benefits, Length: 6205, dtype: int64
```

[24]:
```
data.benefits.fillna("", inplace = True)
```

[25]:
```
data.employment_type.value_counts()
```

[25]:
```
Full-time    11620
Contract      1524
Part-time      797
Temporary      241
Other          227
Name: employment_type, dtype: int64
```

[26]:
```
data.employment_type.fillna("Other", inplace = True)
```

[27]:
```
data.required_experience.value_counts()
```

[27]:
```
Mid-Senior level    3809
Entry level         2697
Associate           2297
Not Applicable      1116
Director             389
Internship           381
Executive            141
Name: required_experience, dtype: int64
```

[28]:
```
data.required_experience.fillna("Not Applicable",inplace=True)
```

[29]:
```
data.required_education.value_counts()
```

[29]:
```
Bachelor's Degree                    5145
High School or equivalent            2080
Unspecified                          1397
Master's Degree                       416
Associate Degree                      274
Certification                         170
Some College Coursework Completed     102
Professional                           74
Vocational                             49
Some High School Coursework            27
Doctorate                              26
Vocational - HS Diploma                 9
Vocational - Degree                     6
Name: required_education, dtype: int64
```

[30]:
```
data.required_education.fillna("Unspecified", inplace = True)
```

[31]:
```
data.industry.value_counts().head(50)
```

[31]:
```
Information Technology and Services    1734
Computer Software                      1376
Internet                               1062
Marketing and Advertising               828
Education Management                    822
Financial Services                      779
Hospital & Health Care                  497
Consumer Services                       358
Telecommunications                      342
Oil & Energy                            287
Retail                                  223
Real Estate                             175
Accounting                              159
Construction                            158
E-Learning                              139
Management Consulting                   130
Design                                  129
Health, Wellness and Fitness            127
Staffing and Recruiting                 127
Insurance                               123
Automotive                              120
Logistics and Supply Chain              112
Human Resources                         108
Online Media                            101
Apparel & Fashion                        97
Legal Services                           97
Facilities Services                      94
Hospitality                              88
Computer Games                           86
Banking                                  84
Building Materials                       78
Leisure, Travel & Tourism                76
Nonprofit Organization Management        76
Entertainment                            74
Electrical/Electronic Manufacturing      73
Food & Beverages                         72
Cosmetics                                65
Airlines/Aviation                        63
Consumer Goods                           63
Consumer Electronics                     62
Medical Practice                         60
Public Relations and Communications      58
Civic & Social Organization              55
Market Research                          54
Transportation/Trucking/Railroad         53
```

```
Restaurants                                    52
Warehousing                                    51
Events Services                                50
Broadcast Media                                50
Computer & Network Security                    49
Name: industry, dtype: int64
```

[32]:
```python
data.industry.fillna("Unspecified", inplace = True)
```

[33]:
```python
data.function.value_counts()
```

[33]:
```
Information Technology    1749
Sales                     1468
Engineering               1348
Customer Service          1229
Marketing                  830
Administrative             630
Design                     340
Health Care Provider       338
Other                      325
Education                  325
Management                 317
Business Development       228
Accounting/Auditing        212
Human Resources            205
Project Management         183
Finance                    172
Consulting                 144
Writing/Editing            132
Art/Creative               132
Production                 116
Product Management         114
Quality Assurance          111
Advertising                 90
Business Analyst            84
Data Analyst                82
Public Relations            76
Manufacturing               74
General Business            68
Research                    50
Legal                       47
Strategy/Planning           46
Training                    38
Supply Chain                36
Financial Analyst           33
Distribution                24
Purchasing                  15
```

```
Science                          14
Name: function, dtype: int64
```

[34]:
```
data.function.fillna("Unspecified", inplace = True)
```

[35]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17880 entries, 0 to 17879
Data columns (total 18 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   job_id               17880 non-null  int64
 1   title                17880 non-null  object
 2   location             17880 non-null  object
 3   department           17880 non-null  object
 4   salary_range         17880 non-null  object
 5   company_profile      17880 non-null  object
 6   description          17880 non-null  object
 7   requirements         17880 non-null  object
 8   benefits             17880 non-null  object
 9   telecommuting        17880 non-null  int64
 10  has_company_logo     17880 non-null  int64
 11  has_questions        17880 non-null  int64
 12  employment_type      17880 non-null  object
 13  required_experience  17880 non-null  object
 14  required_education   17880 non-null  object
 15  industry             17880 non-null  object
 16  function             17880 non-null  object
 17  fraudulent           17880 non-null  int64
dtypes: int64(5), object(13)
memory usage: 2.5+ MB
```

[36]:
```
data.columns
```

[36]:
```
Index(['job_id', 'title', 'location', 'department', 'salary_range',
       'company_profile', 'description', 'requirements', 'benefits',
       'telecommuting', 'has_company_logo', 'has_questions', 'employment_type',
       'required_experience', 'required_education', 'industry', 'function',
       'fraudulent'],
      dtype='object')
```

[37]:
```
# generate a new column data['text'] to collect all information provided in job␣
↪posting
data['text'] = ""
```

```
text_columns = ['title', 'location', 'department','company_profile',␣
 ↪'description',␣
 ↪'requirements','benefits','employment_type','required_experience',␣
 ↪'required_education', 'industry', 'function']
for col in text_columns:
    data['text']= data['text'] + " " + data[col]
```

[38]:
```
data.text[0]
```

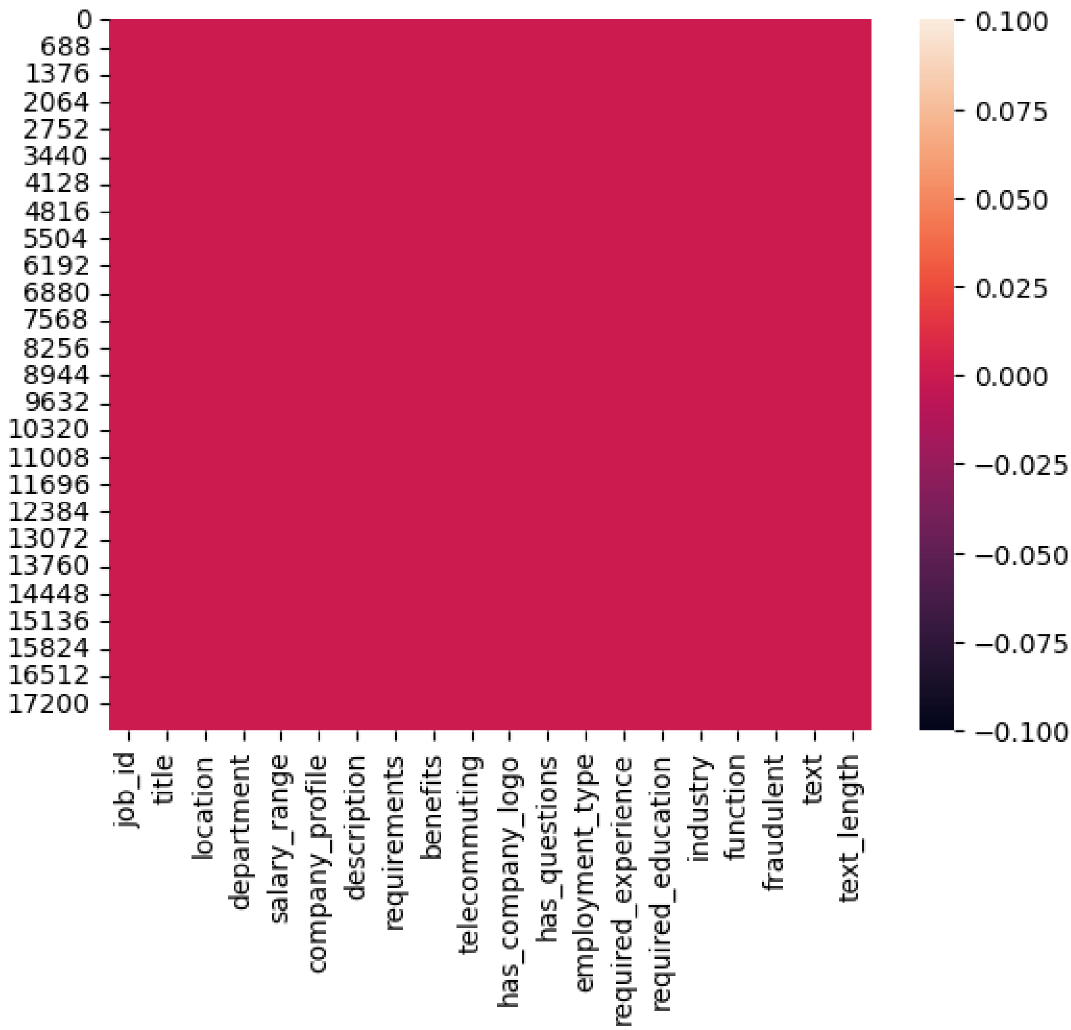
[38]:
```
" Marketing Intern US, NY, New York Marketing We're Food52, and we've created a
groundbreaking and award-winning cooking site. We support, connect, and
celebrate home cooks, and give them everything they need in one place.We have a
top editorial, business, and engineering team. We're focused on using technology
to find new and better ways to connect people around their specific food
interests, and to offer them superb, highly curated information about food and
cooking. We attract the most talented home cooks and contributors in the
country; we also publish well-known professionals like Mario Batali, Gwyneth
Paltrow, and Danny Meyer. And we have partnerships with Whole Foods Market and
Random House.Food52 has been named the best food website by the James Beard
Foundation and IACP, and has been featured in the New York Times, NPR, Pando
Daily, TechCrunch, and on the Today Show.We're located in Chelsea, in New York
City. Food52, a fast-growing, James Beard Award-winning online food community
and crowd-sourced and curated recipe hub, is currently interviewing full- and
part-time unpaid interns to work in a small team of editors, executives, and
developers in its New York City headquarters.Reproducing and/or repackaging
existing Food52 content for a number of partner sites, such as Huffington Post,
Yahoo, Buzzfeed, and more in their various content management systemsResearching
blogs and websites for the Provisions by Food52 Affiliate ProgramAssisting in
day-to-day affiliate program support, such as screening affiliates and assisting
in any affiliate inquiriesSupporting with PR &amp; Events when neededHelping
with office administrative work, such as filing, mailing, and preparing for
meetingsWorking with developers to document bugs and suggest improvements to the
siteSupporting the marketing and executive staff Experience with content
management systems a major plus (any blogging counts!)Familiar with the Food52
editorial voice and aestheticLoves food, appreciates the importance of home
cooking and cooking with the seasonsMeticulous editor, perfectionist, obsessive
attention to detail, maddened by typos and broken links, delighted by finding
and fixing themCheerful under pressureExcellent communication skillsA+ multi-
tasker and juggler of responsibilities big and smallInterested in and engaged
with social media like Twitter, Facebook, and PinterestLoves problem-solving and
collaborating to drive Food52 forwardThinks big picture but pitches in on the
nitty gritty of running a small company (dishes, shopping, administrative
support)Comfortable with the realities of working for a startup: being on call
on evenings and weekends, and working long hours  Other Internship Unspecified
Unspecified Marketing"
```

[39]:

```
# generate a new columns of text_length of the job posting
data["text_length"]= data["text"].str.len()
```

[40]:

```
sns.heatmap(data.isnull())
plt.show()
```



1.3 Data Visualization and Exploratory Data Analysis

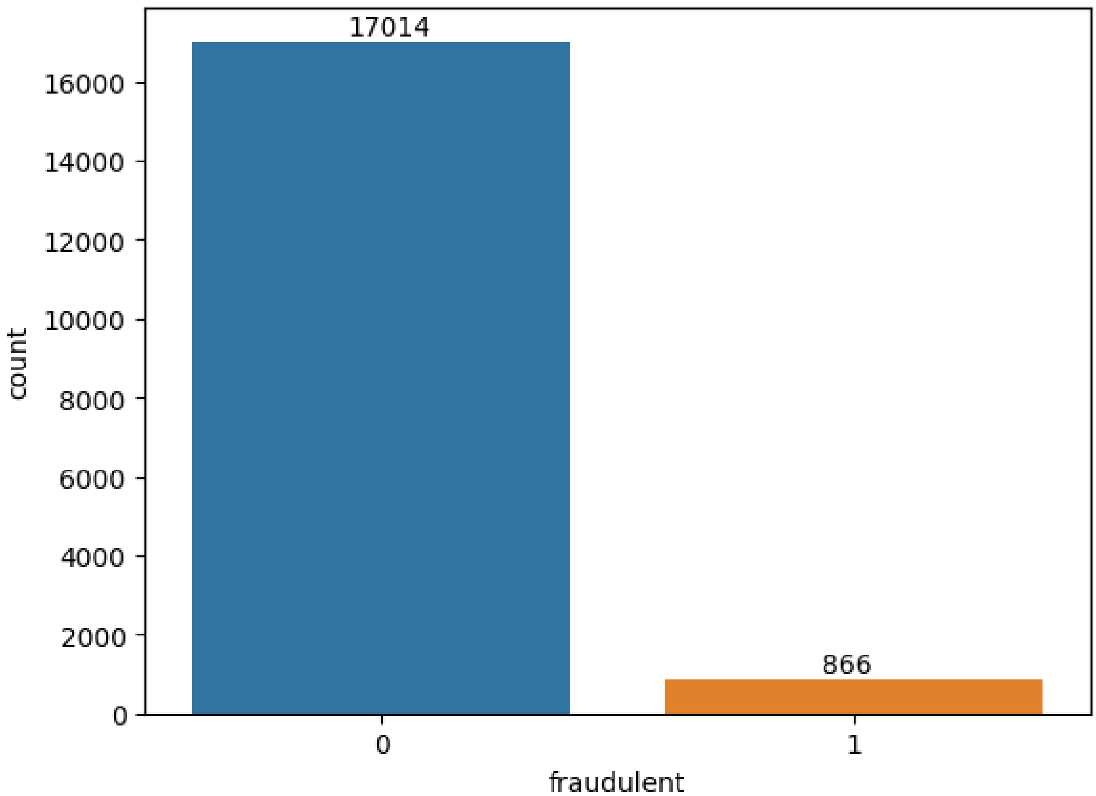
1.3.1 Target Variable Analysis

[41]:

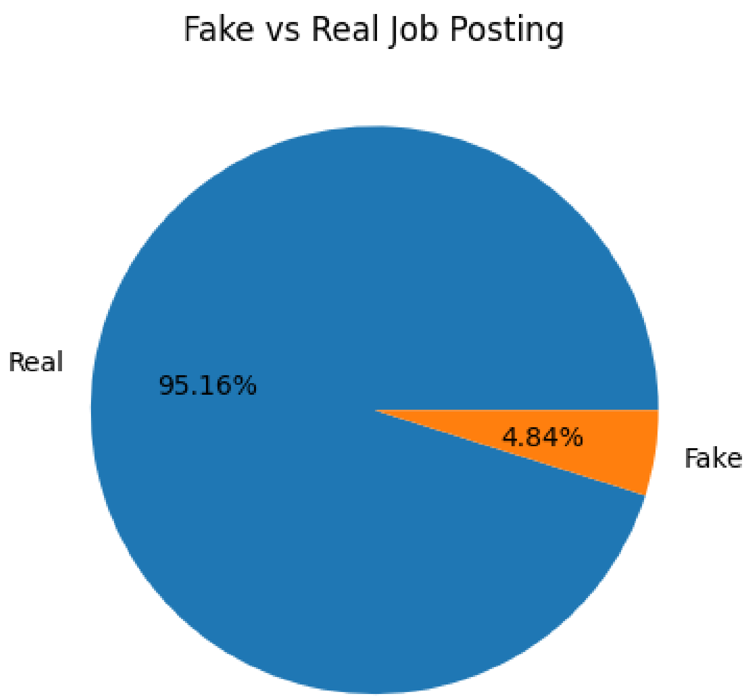
```
# Target Variable - fraudulent
ax = sns.countplot(x = 'fraudulent', data = data)
for p in ax.patches:
    ax.annotate('{:.0f}'.format(p.get_height()), (p.get_x()+0.33, p.
↪get_height()+160))
```

```python
data.fraudulent.mean()
# The fraud rate is 4.84%. It is an imbalanced dataset.
```

[41]: 0.048434004474272935



[42]:
```python
labels=["Real","Fake"]
real = data[data["fraudulent"] == 0]["fraudulent"].count()/
 ↪len(data["fraudulent"])
fake = data[data["fraudulent"] == 1]["fraudulent"].count()/
 ↪len(data["fraudulent"])
sizes = [real,fake]
p, tx, autotexts = plt.pie(sizes, labels=labels,autopct="")

for i, a in enumerate(autotexts):
    a.set_text("{:.2f}%".format(sizes[i]*100))
plt.title("Fake vs Real Job Posting")
plt.show()
```



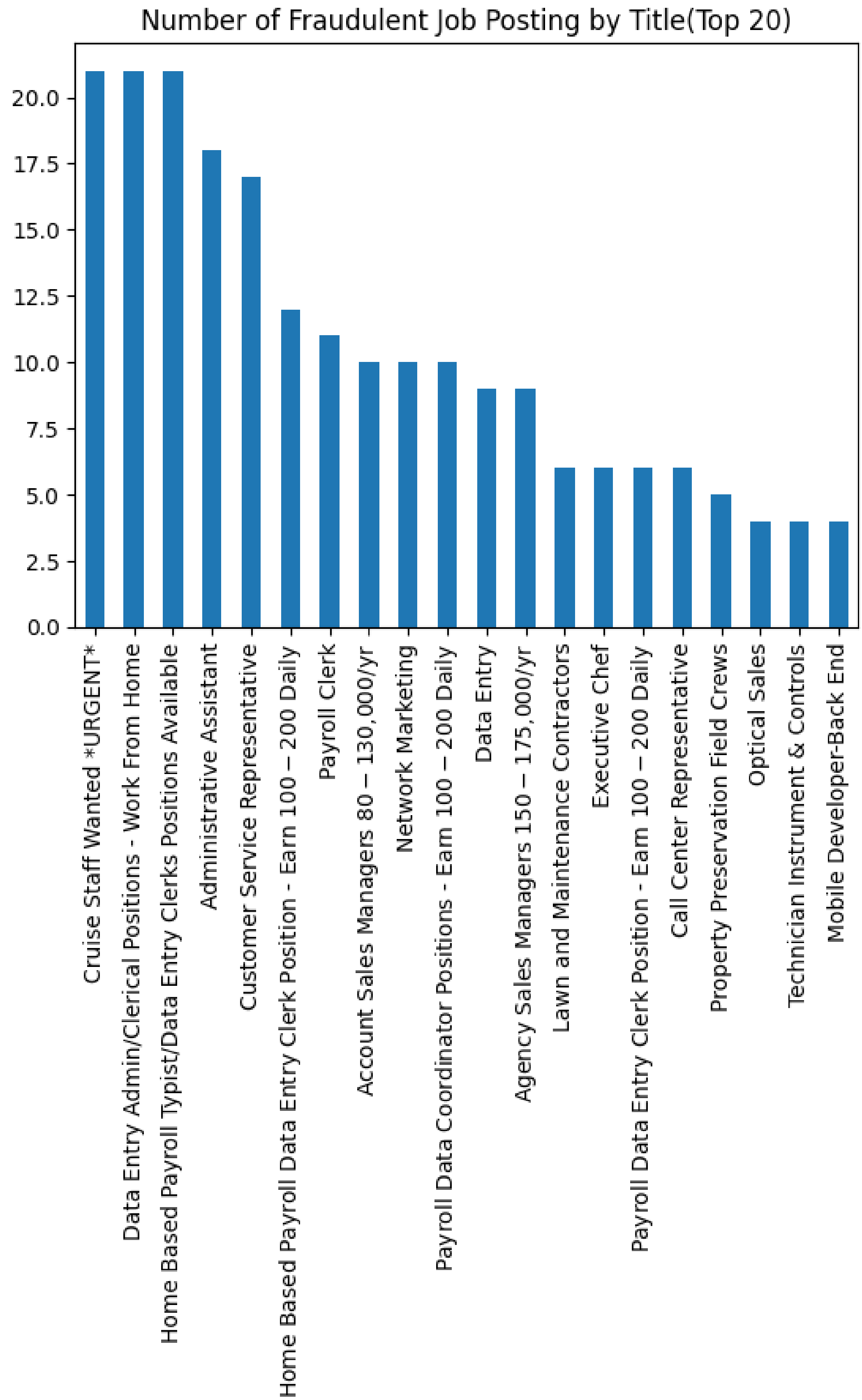
1.3.2 Categorical Variables Analysis

[43]:
```
data.title = data.title.str.strip()
data.title.value_counts().head(20)
```

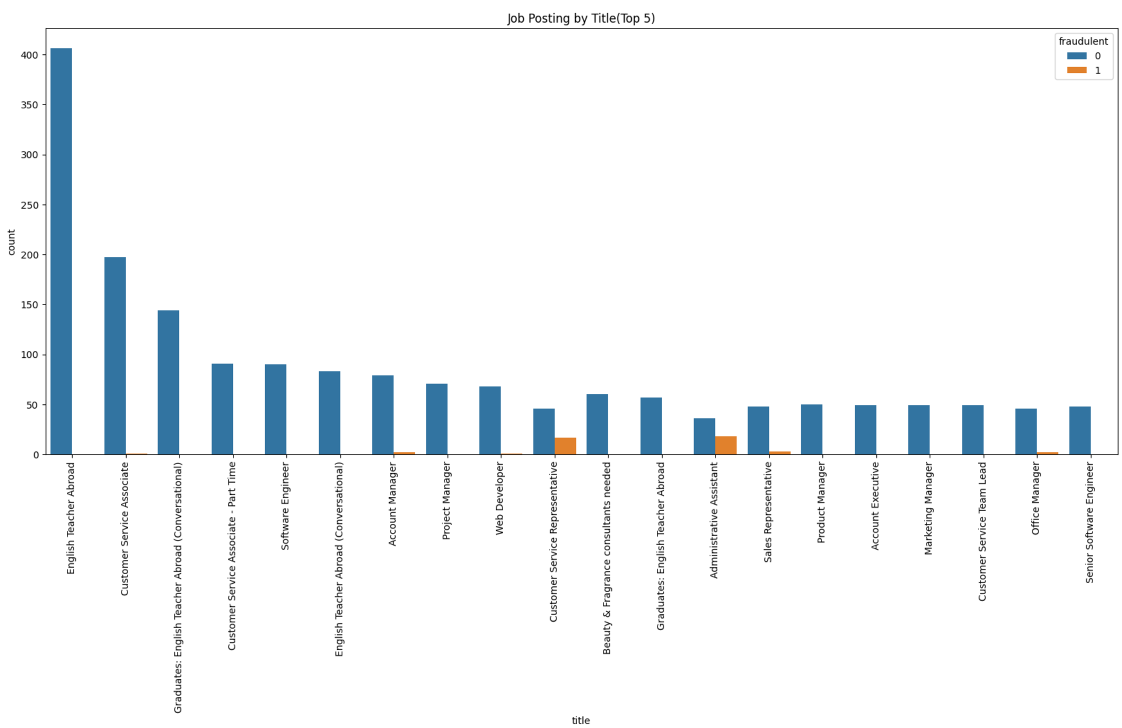
[43]:
```
English Teacher Abroad                                406
Customer Service Associate                            198
Graduates: English Teacher Abroad (Conversational)    144
Customer Service Associate - Part Time                 91
Software Engineer                                      90
English Teacher Abroad (Conversational)                83
Account Manager                                        81
Project Manager                                        71
Web Developer                                          69
Customer Service Representative                        63
Beauty & Fragrance consultants needed                  60
Graduates: English Teacher Abroad                      57
Administrative Assistant                               54
Sales Representative                                   51
Product Manager                                        50
Account Executive                                      49
Marketing Manager                                      49
```

```
Customer Service Team Lead                                        49
Office Manager                                                    48
Senior Software Engineer                                          48
Name: title, dtype: int64
```

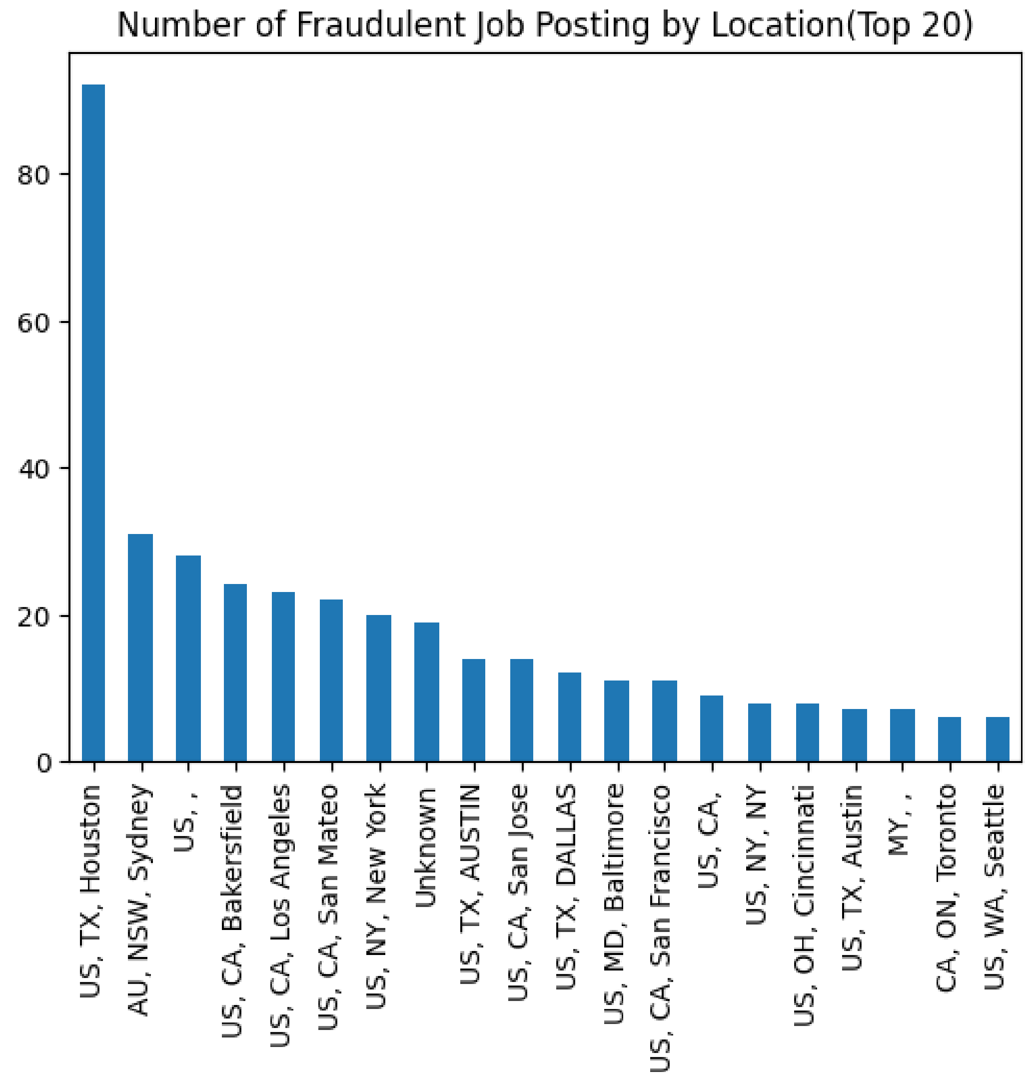
[44]:
```
# title
data[data['fraudulent'] == 1].title.value_counts(sort =True, ascending = False).
 ↪head(20).plot(kind = 'bar')
plt.title("Number of Fraudulent Job Posting by Title(Top 20)")
plt.show()
# Insight: title could be a predictor.
```


Number of Fraudulent Job Posting by Title(Top 20)
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Cruise Staff Wanted *URGENT*
Data Entry Admin/Clerical Positions - Work From Home
Home Based Payroll Typist/Data Entry Clerks Positions Available
Administrative Assistant
Customer Service Representative
Home Based Payroll Data Entry Clerk Position - Earn 100 – 200 Daily
Payroll Clerk
Account Sales Managers 80 – 130,000/yr
Network Marketing
Payroll Data Coordinator Positions - Earn 100 – 200 Daily
Data Entry
Agency Sales Managers 150 – 175,000/yr
Lawn and Maintenance Contractors
Executive Chef
Payroll Data Entry Clerk Position - Earn 100 – 200 Daily
Call Center Representative
Property Preservation Field Crews
Optical Sales
Technician Instrument & Controls
Mobile Developer-Back End

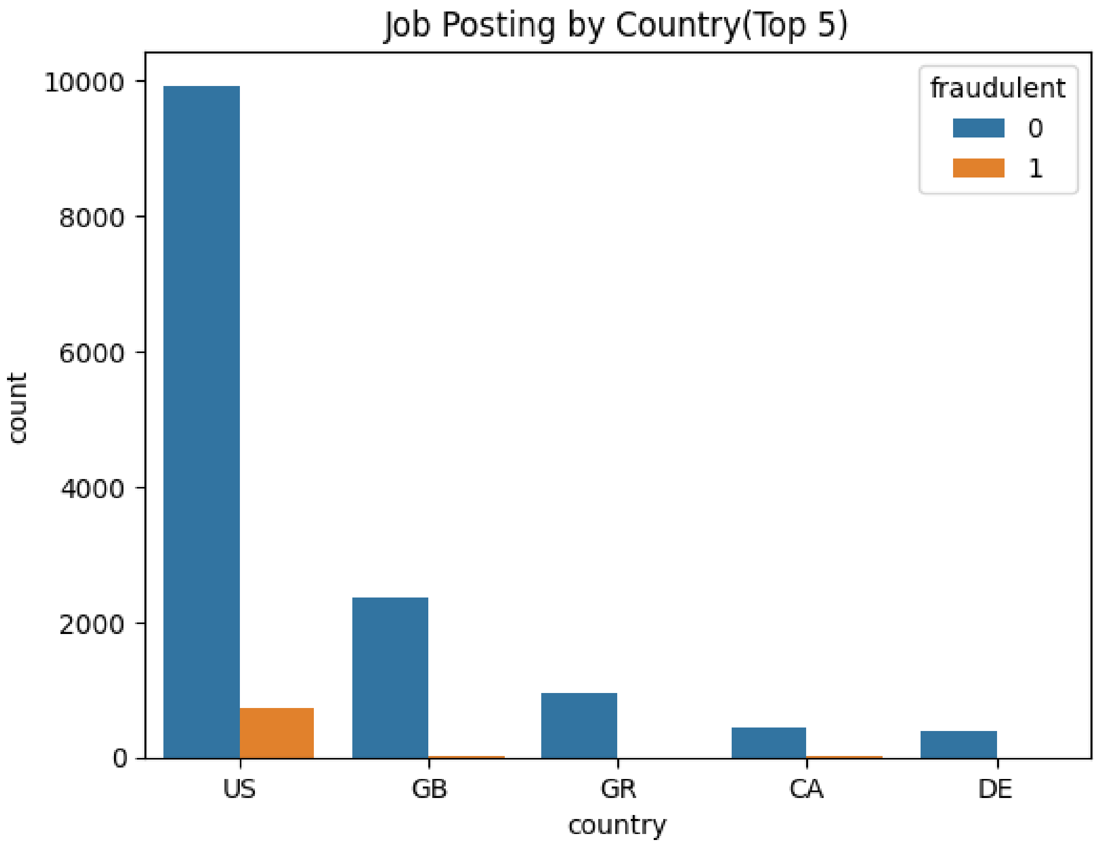
[45]:
```
data.title.value_counts().head(20)
title_grp = data.title.value_counts().head(20).index

plt.figure(figsize=(20,8))
sns.countplot(x="title",hue="fraudulent",data=data[data["title"].
 ↪isin(title_grp)],order=title_grp);
plt.title("Job Posting by Title(Top 5)")
plt.xticks(rotation=90)
plt.show()

# Insight: title could be a predictor
```



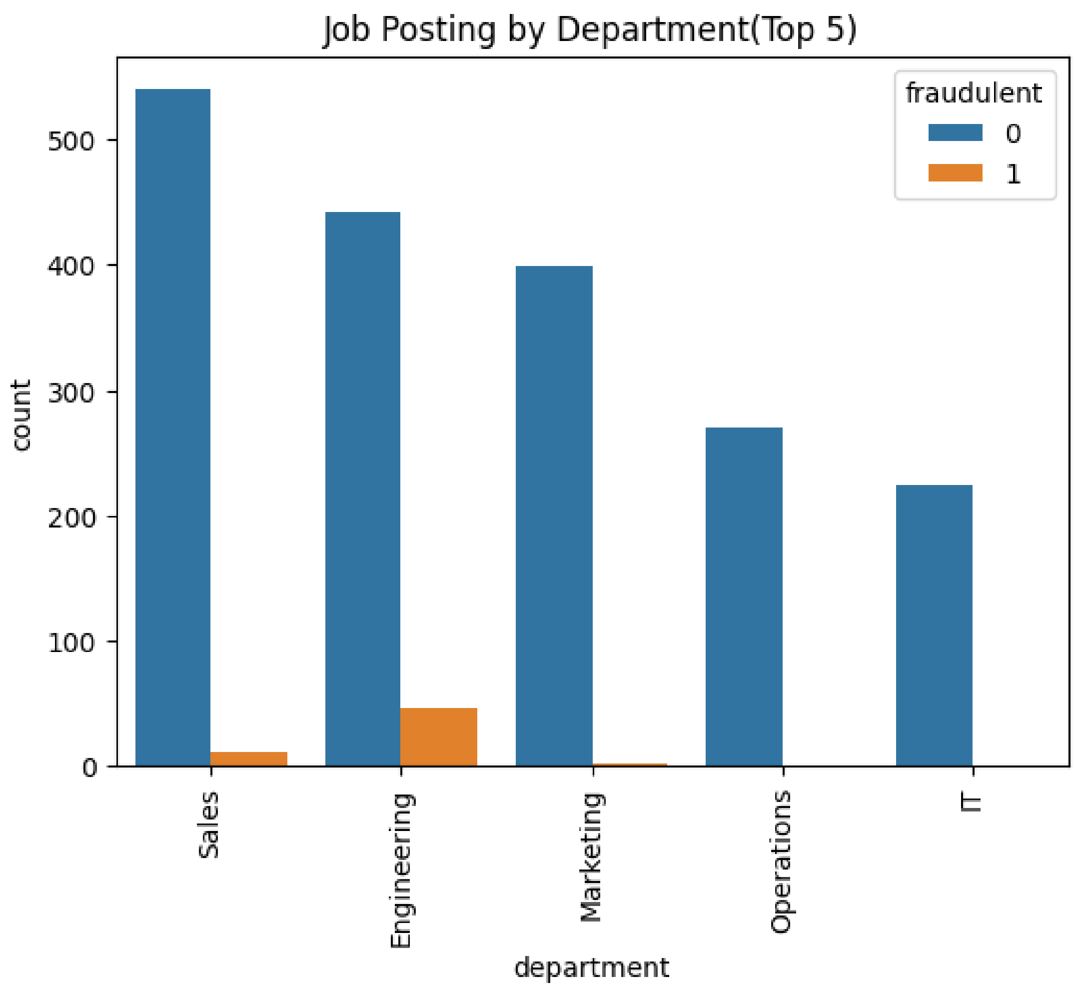
[46]:
```
# location
data[data['fraudulent'] == 1].location.value_counts(sort =True, ascending =␣
 ↪False).head(20).plot(kind = 'bar')
plt.title("Number of Fraudulent Job Posting by Location(Top 20)")
plt.show()

# Insight: title could be a predictor.
```



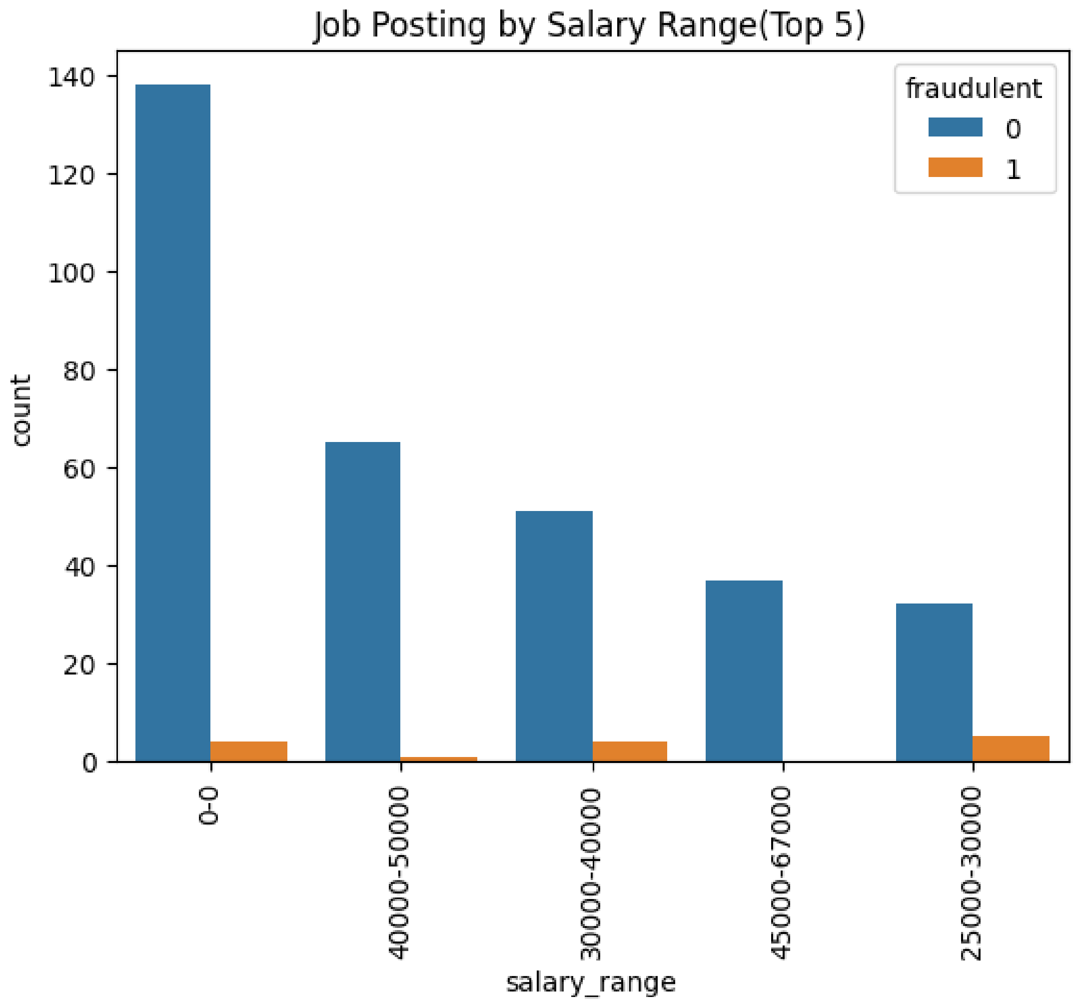
[47]:
```
# generat new columns from location: data['country']
def code(string):
    return string.split(",")[0]
data['country'] = data['location'].apply(code)

data.country.value_counts().head(5)
grp = data.country.value_counts().head(5).index
sns.countplot(x="country",hue="fraudulent",data=data[data["country"].
 ↪isin(grp)],order=grp);
plt.title("Job Posting by Country(Top 5)")
plt.show()
```



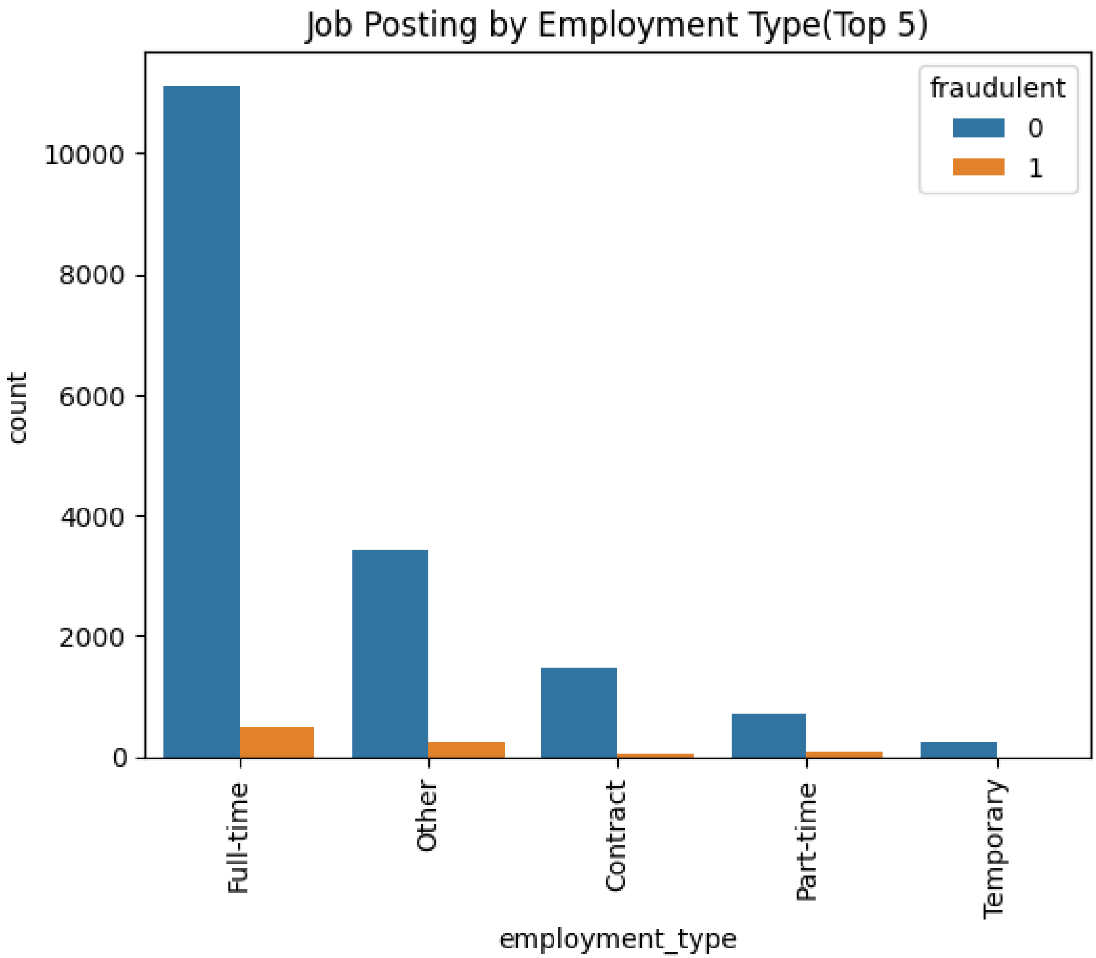
[48]:
```
# department
dp_grp = data.department.value_counts().head(6).index
dp_grp = dp_grp[1:]
sns.countplot(x="department",hue="fraudulent",data=data[data["department"].
    ↪isin(dp_grp)],order=dp_grp);
plt.title("Job Posting by Department(Top 5)")
plt.xticks(rotation=90)
plt.show()
```



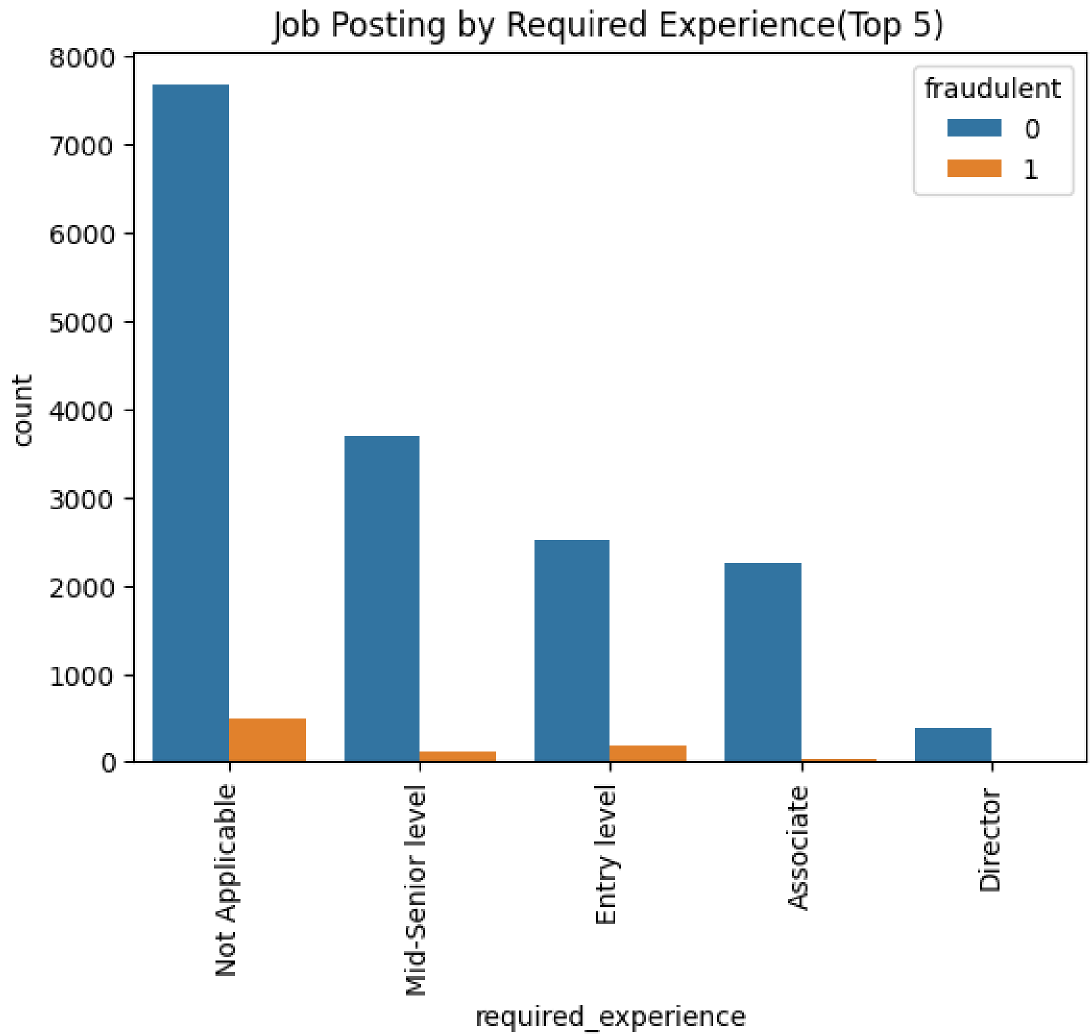
[49]:
```
# salary_range
sr_grp = data.salary_range.value_counts().head(6).index[1:]
sns.countplot(x="salary_range",hue="fraudulent",data=data[data["salary_range"].
 ↪isin(sr_grp)],order=sr_grp);
plt.title("Job Posting by Salary Range(Top 5)")
plt.xticks(rotation=90)
plt.show()
```



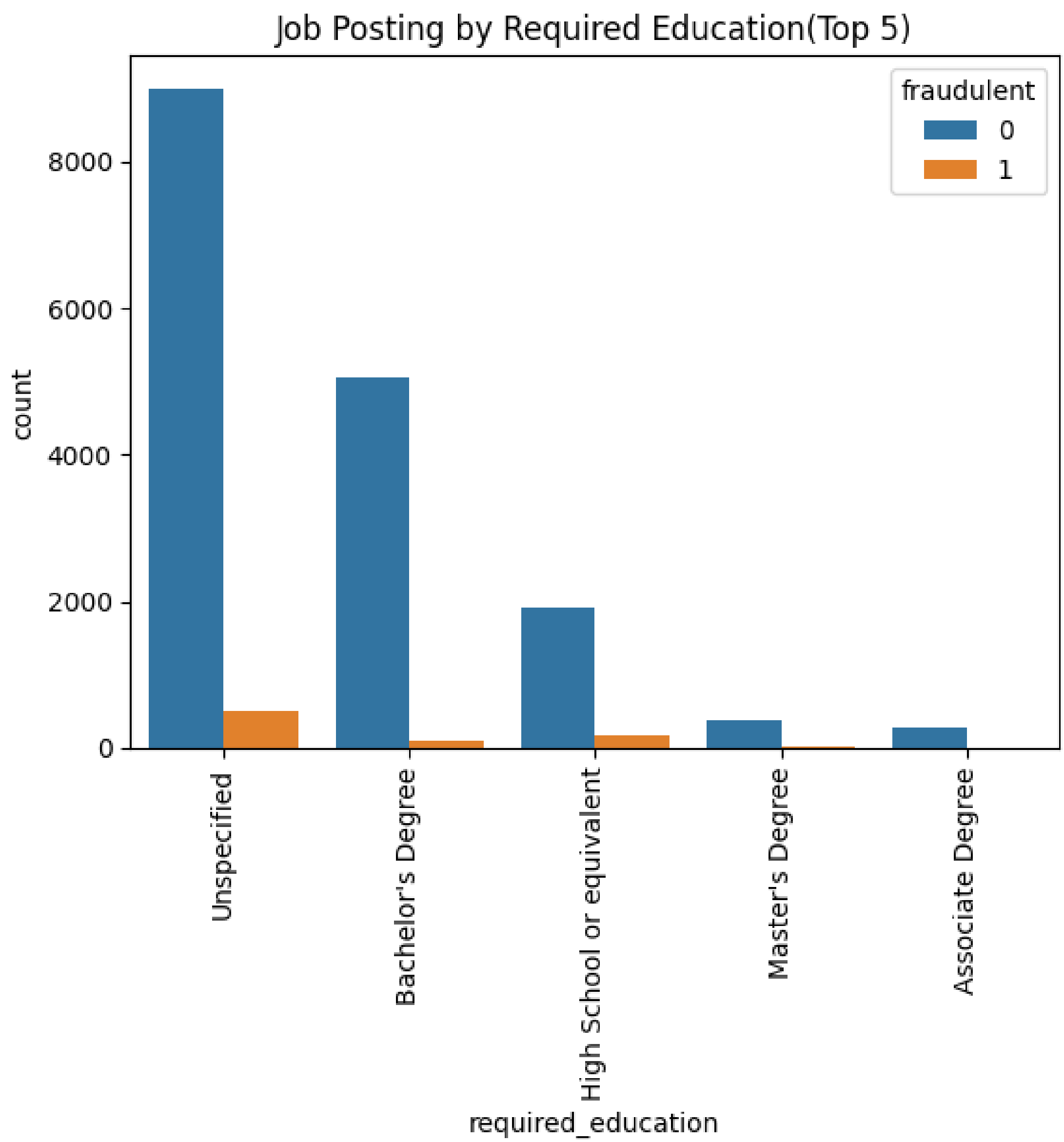
[50]:
```
# employment_type
et_grp = data.employment_type.value_counts().head(5).index
sns.
 ↪countplot(x="employment_type",hue="fraudulent",data=data[data["employment_type"].
 ↪isin(et_grp)],order=et_grp);
plt.title("Job Posting by Employment Type(Top 5)")
plt.xticks(rotation=90)
plt.show()#
```



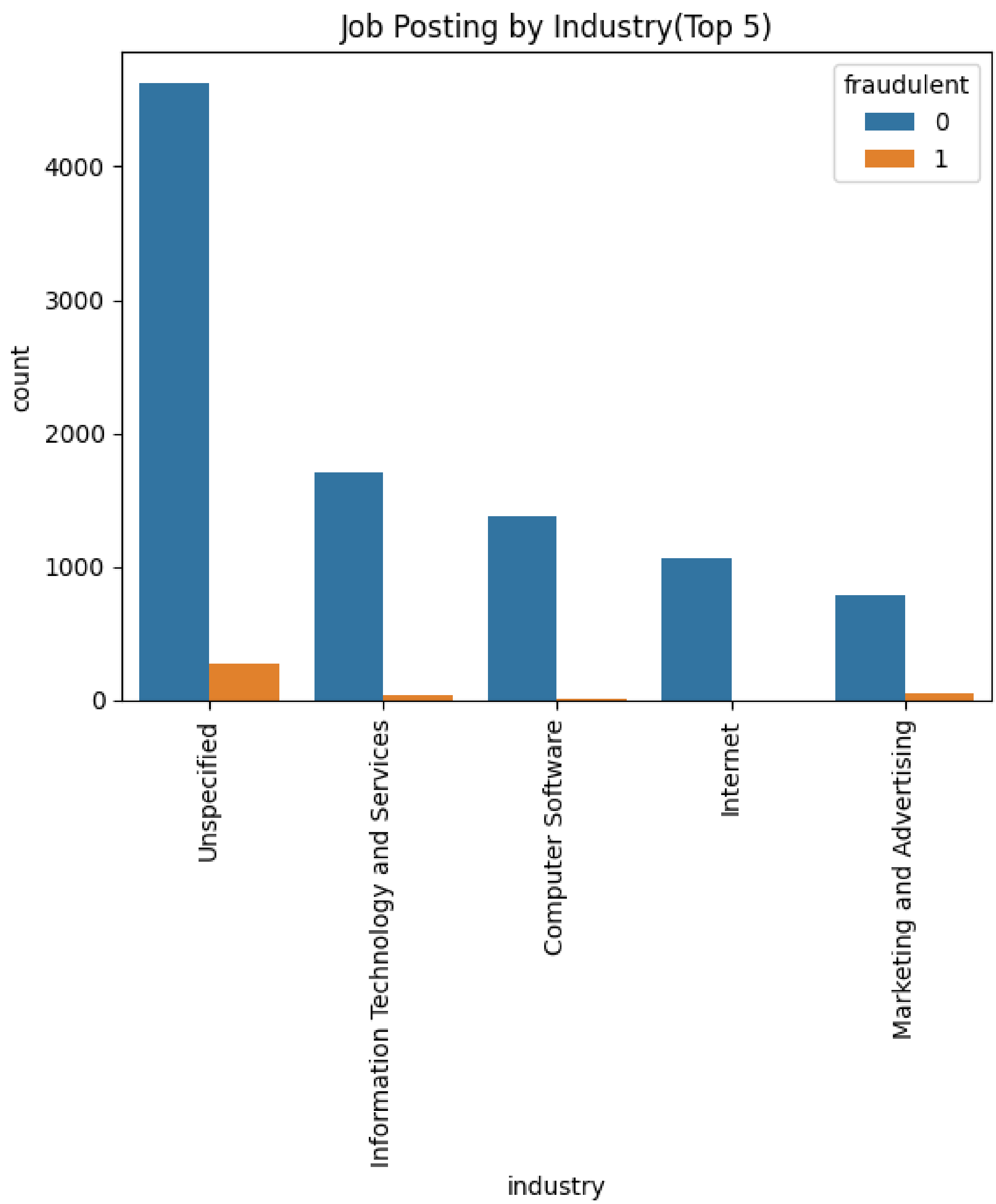
[51]:
```
# required_experience
rex_grp = data.required_experience.value_counts().head(5).index
sns.
  ↪countplot(x="required_experience",hue="fraudulent",data=data[data["required_experience"].
  ↪isin(rex_grp)],order=rex_grp);
plt.title("Job Posting by Required Experience(Top 5)")
plt.xticks(rotation=90)
plt.show()
```



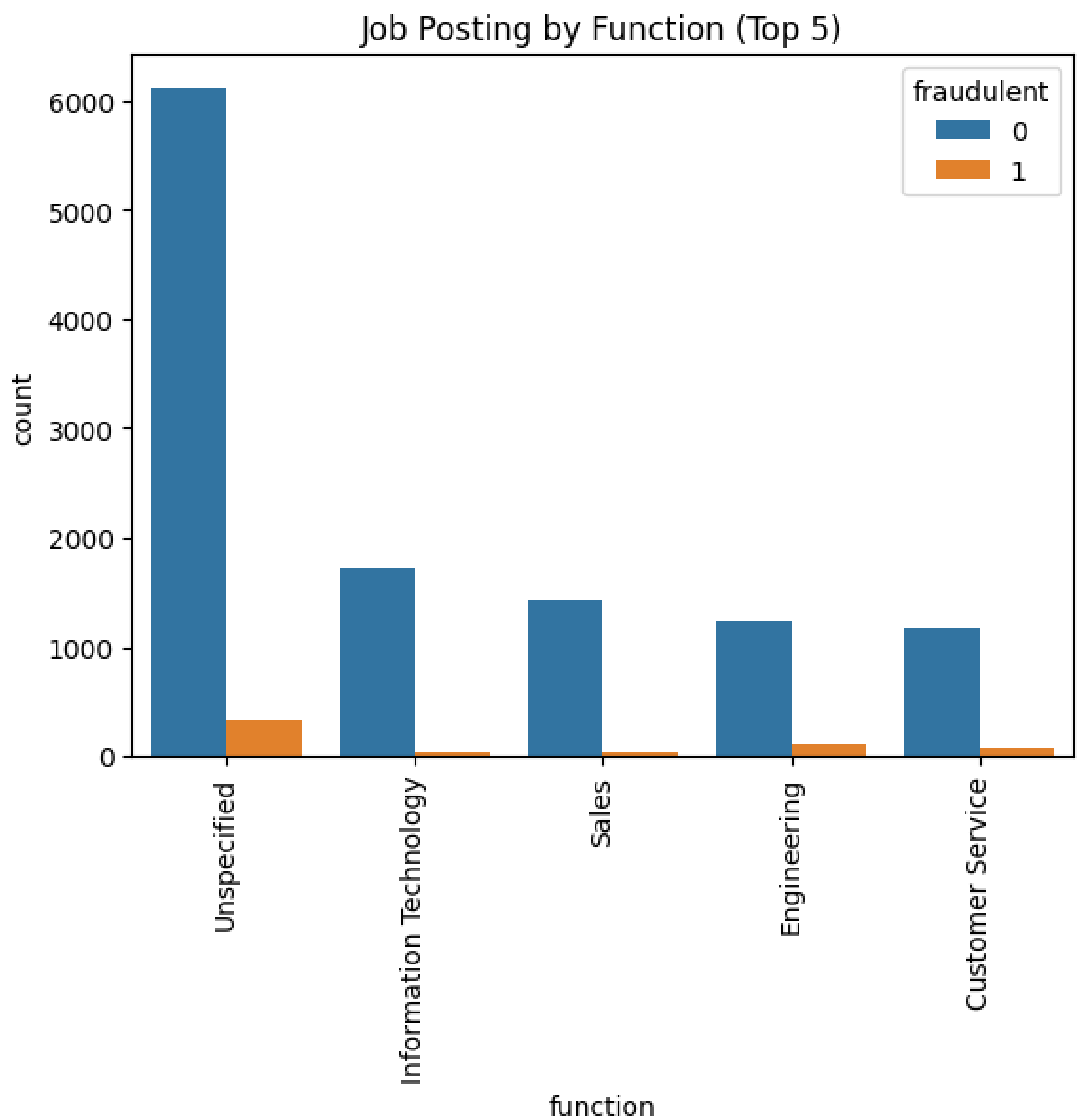
[52]:
```
# required_education
red_grp = data.required_education.value_counts().head(5).index
sns.
 ↪countplot(x="required_education",hue="fraudulent",data=data[data["required_education"].
 ↪isin(red_grp)],order=red_grp);
plt.title("Job Posting by Required Education(Top 5)")
plt.xticks(rotation=90)
plt.show()
```



[53]:
```
# industry
in_grp = data.industry.value_counts().head(5).index
sns.countplot(x="industry",hue="fraudulent",data=data[data["industry"].
  ↪isin(in_grp)],order=in_grp);
plt.title("Job Posting by Industry(Top 5)")
plt.xticks(rotation=90)
plt.show()
```

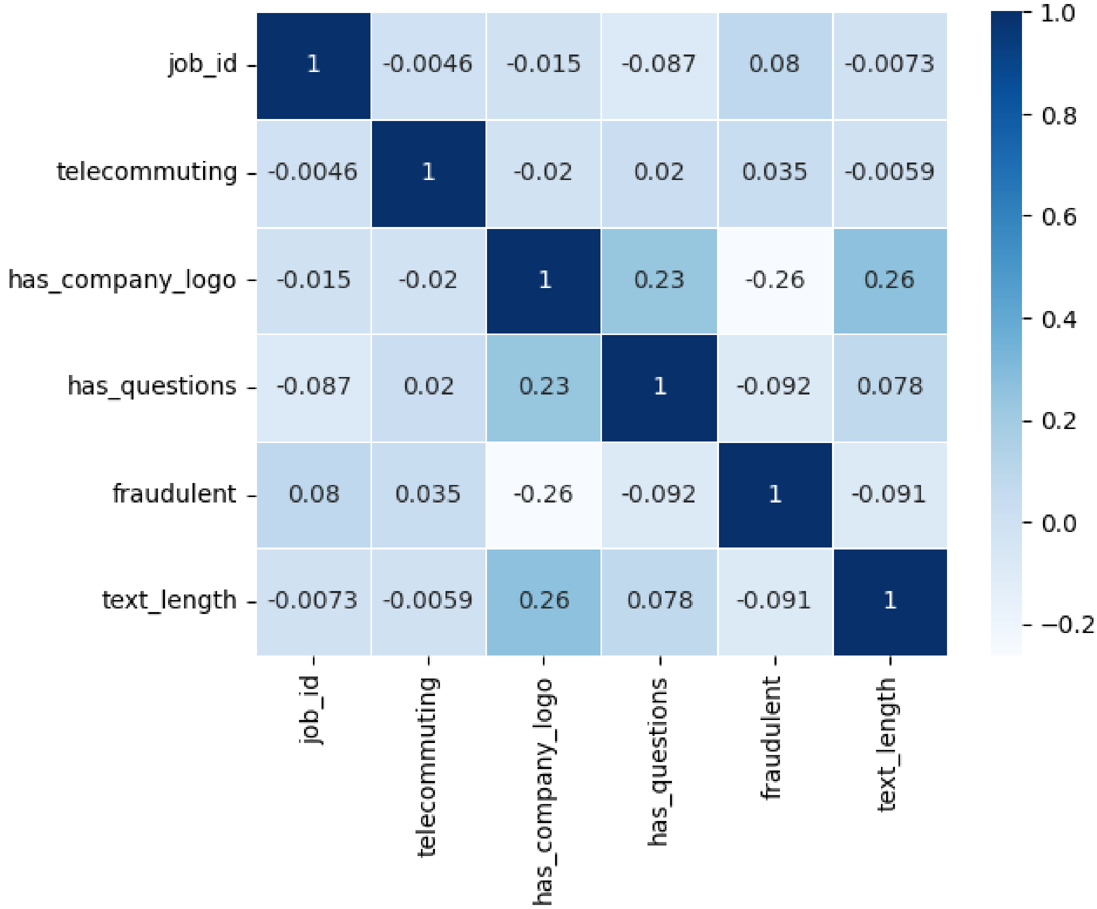


[54]:
```
# function
fu_grp = data.function.value_counts().head(5).index
sns.countplot(x="function",hue="fraudulent",data=data[data["function"].
    ↪isin(fu_grp)],order=fu_grp);
plt.title("Job Posting by Function (Top 5)")
plt.xticks(rotation=90)
plt.show()
```

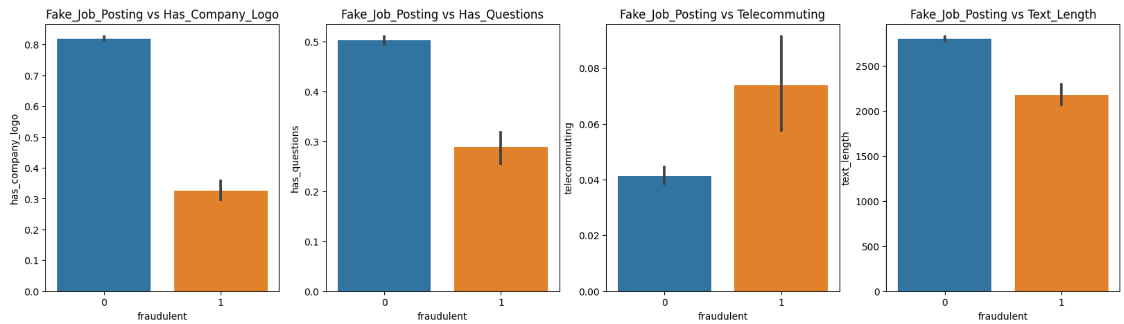


1.3.3 Numerical Variables Analysis

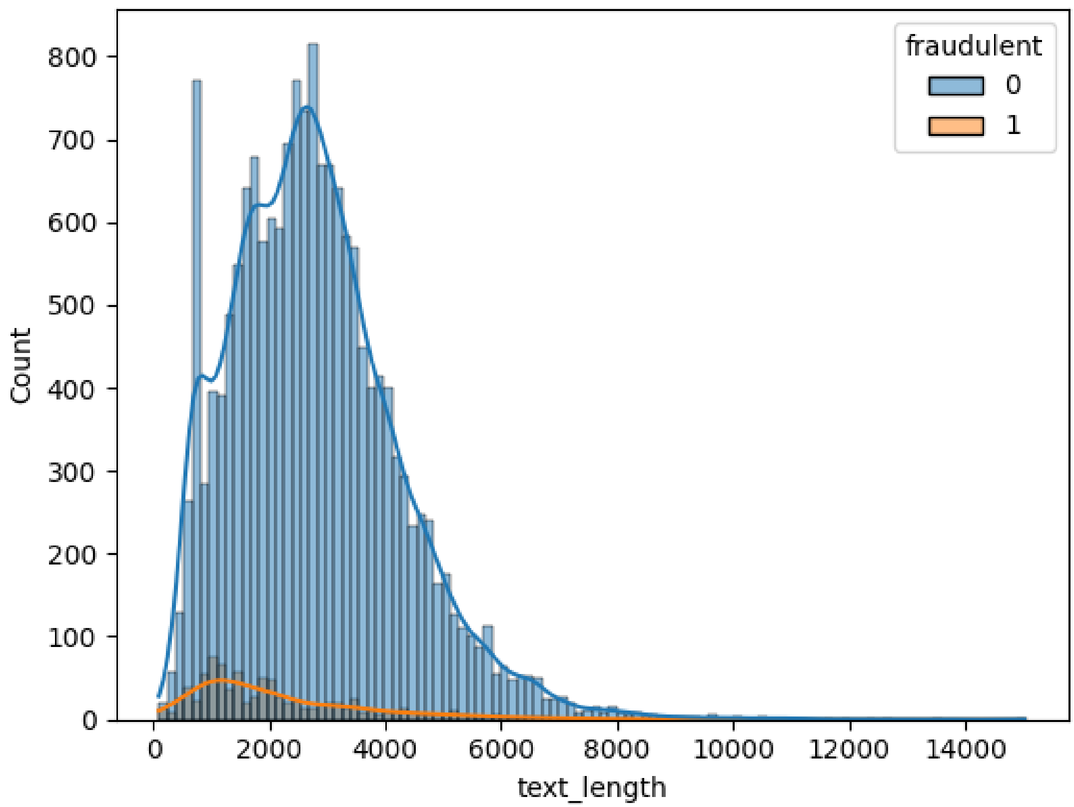
[55]:
```
#Plotting the heat map to find the correlation between the numerical columns
sns.heatmap(data.corr(),annot=True, linewidths=0.5, cmap = "Blues")
plt.show()

# Insight: we could see that has_company_logo is strongly correlated with␣
↪fradulent columns. has_company_logo could be a good predictor
```



[56]:
```
fig, axes = plt.subplots(1,4,figsize=(20,5))
sns.barplot(x ='fraudulent',y= 'has_company_logo', data = data, ax=axes[0]).
 ↪set(title='Fake_Job_Posting vs Has_Company_Logo')
sns.barplot(x = 'fraudulent',y= 'has_questions', data = data, ax=axes[1]).
 ↪set(title='Fake_Job_Posting vs Has_Questions')
sns.barplot(x = 'fraudulent',y='telecommuting',data = data,  ax = axes[2]).
 ↪set(title='Fake_Job_Posting vs Telecommuting')
sns.barplot(x = 'fraudulent',y='text_length',data = data,  ax = axes[3]).
 ↪set(title='Fake_Job_Posting vs Text_Length')
plt.show()
# Insights: We could see the differences of point estimates between real and␣
 ↪fake job posting. Fake job posting has lower prob. in has_company_logo,␣
 ↪has_questions. Fake job posting has less text on average. Fake job has a␣
 ↪higher prob. to be telecommuting.
# All of has_company_logo, has_questions,telecommuting,text_length could be␣
 ↪good predictors.
```



[57]:
```
sns.histplot(x='text_length', hue = 'fraudulent', data = data, kde=True)
plt.show()
```



[58]:
```
data1 = data
```

1.4 Data Modeling

[12]:
```python
import pandas as pd
data1 = pd.read_csv('data1.csv')
```

[13]:
```python
data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17880 entries, 0 to 17879
Data columns (total 21 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   job_id               17880 non-null  int64
 1   title                17880 non-null  object
 2   location             17880 non-null  object
 3   department           17880 non-null  object
 4   salary_range         17880 non-null  object
 5   company_profile      14572 non-null  object
 6   description          17879 non-null  object
 7   requirements         15185 non-null  object
 8   benefits             10670 non-null  object
 9   telecommuting        17880 non-null  int64
 10  has_company_logo     17880 non-null  int64
 11  has_questions        17880 non-null  int64
 12  employment_type      17880 non-null  object
 13  required_experience  17880 non-null  object
 14  required_education   17880 non-null  object
 15  industry             17880 non-null  object
 16  function             17880 non-null  object
 17  fraudulent           17880 non-null  int64
 18  text                 17880 non-null  object
 19  text_length          17880 non-null  int64
 20  country              17880 non-null  object
dtypes: int64(6), object(15)
memory usage: 2.9+ MB
```

[14]:
```python
#prepare data for modeling
drop_cols = ['job_id', 'location','salary_range','company_profile',␣
↪'description', 'requirements', 'benefits', 'text']
data1 = data1.drop(drop_cols, axis =1)
```

[15]:
```python
data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17880 entries, 0 to 17879
Data columns (total 13 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
```

```
 0   title                17880 non-null  object
 1   department           17880 non-null  object
 2   telecommuting        17880 non-null  int64
 3   has_company_logo     17880 non-null  int64
 4   has_questions        17880 non-null  int64
 5   employment_type      17880 non-null  object
 6   required_experience  17880 non-null  object
 7   required_education   17880 non-null  object
 8   industry             17880 non-null  object
 9   function             17880 non-null  object
 10  fraudulent           17880 non-null  int64
 11  text_length          17880 non-null  int64
 12  country              17880 non-null  object
dtypes: int64(5), object(8)
memory usage: 1.8+ MB
```

[16]:
```
data1.columns
```

[16]:
```
Index(['title', 'department', 'telecommuting', 'has_company_logo',
       'has_questions', 'employment_type', 'required_experience',
       'required_education', 'industry', 'function', 'fraudulent',
       'text_length', 'country'],
      dtype='object')
```

[17]:
```
from sklearn.preprocessing import LabelEncoder
labelencoder = LabelEncoder()
cat_cols = ['title', 'department', 'employment_type', 'required_experience',
       'required_education', 'industry', 'function', 'country']
for c in cat_cols:
    data1[c]= labelencoder.fit_transform(data1[c])
```

[18]:
```
data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17880 entries, 0 to 17879
Data columns (total 13 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   title                17880 non-null  int64
 1   department           17880 non-null  int64
 2   telecommuting        17880 non-null  int64
 3   has_company_logo     17880 non-null  int64
 4   has_questions        17880 non-null  int64
 5   employment_type      17880 non-null  int64
 6   required_experience  17880 non-null  int64
 7   required_education   17880 non-null  int64
 8   industry             17880 non-null  int64
 9   function             17880 non-null  int64
```

```
 10  fraudulent           17880 non-null  int64
 11  text_length          17880 non-null  int64
 12  country              17880 non-null  int64
dtypes: int64(13)
memory usage: 1.8 MB
```

1.4.1 Data Modeling: Part1

[19]:
```
from sklearn.linear_model import LogisticRegression # For Logistic Regression␣
 ↪Model
from sklearn.tree import DecisionTreeClassifier # For Desicion Tree␣
 ↪Classification Model
from sklearn.ensemble import RandomForestClassifier # For Random Forest␣
 ↪Classification Model
from sklearn.model_selection import GridSearchCV # For hyperparameters tuning
```

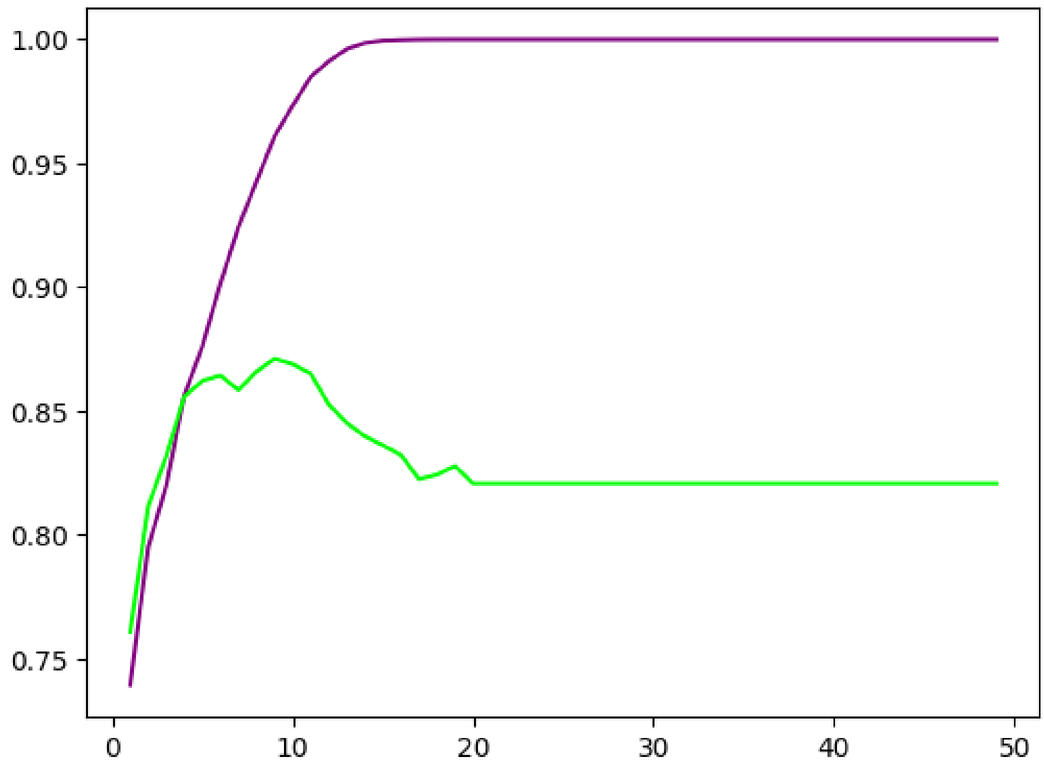
[34]:
```
# Baseline Model: Decision Tree,
# Comparision Model: LogisticRegression,Random Forest, GB

feature_cols = ['title', 'department', 'telecommuting', 'has_company_logo',
       'has_questions', 'employment_type', 'required_experience',
       'required_education', 'industry', 'function','text_length', 'country']
X = data1[feature_cols] # Features

y = data1["fraudulent"] # Target variable
```

[21]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,␣
 ↪random_state=42)
```

[26]:
```
# Hyperparameters tuning for Dasicion Tree model
from sklearn import metrics
from sklearn.metrics import␣
 ↪accuracy_score,f1_score,precision_score,recall_score,roc_auc_score
train_score = []
test_score = []
max_score = 0
max_pair = (0,0)

for i in range(1,50):
    tree = DecisionTreeClassifier(max_depth=i,random_state=42)
    tree.fit(X_train,y_train)
    y_pred = tree.predict_proba(X_train)[:,1]
    y_pred_t = tree.predict_proba(X_test)[:,1]
    train_score.append(metrics.roc_auc_score(y_train,y_pred))
    test_score.append(metrics.roc_auc_score(y_test, y_pred_t))
    test_pair = (i,metrics.roc_auc_score(y_test,y_pred_t))
```

```
    if test_pair[1] > max_pair[1]:
        max_pair = test_pair

fig, ax = plt.subplots()

ax.plot(np.arange(1,50), train_score, label = "roc_auc_score",color='purple')
ax.plot(np.arange(1,50), test_score, label = "roc_auc_score",color='lime')
print(f'Best max_depth is: {max_pair[0]} \nroc_auc_score is: {max_pair[1]}')
```

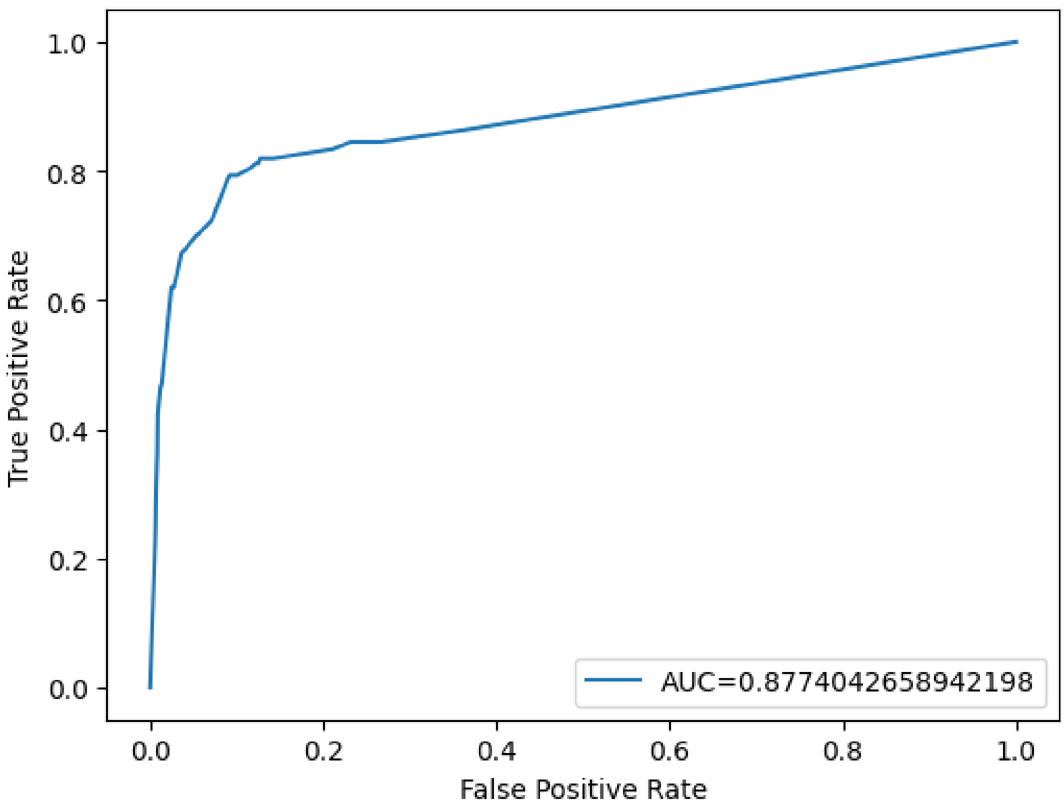
```
Best max_depth is: 9
roc_auc_score is: 0.8711138868702648
```



[27]:
```
# Create Decision Tree classifer object
dtm = DecisionTreeClassifier(criterion="entropy", max_depth=9)

# Train Decision Tree Classifer
dtm = dtm.fit(X_train,y_train)
```

[28]:
```
#define metrics
y_pred_proba = dtm.predict_proba(X_test)[:,1]
fpr, tpr, thresholds = metrics.roc_curve(y_test, y_pred_proba)
auc = metrics.roc_auc_score(y_test, y_pred_proba)
```

```python
#create ROC curve
plt.plot(fpr,tpr,label="AUC="+str(auc))
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.legend(loc=4)
plt.show()

# Calculate the G-mean
gmean = np.sqrt(tpr * (1 - fpr)) # using G-mean

# Find the optimal threshold
index = np.argmax(gmean)
thresholdOpt = round(thresholds[index], ndigits = 4)
gmeanOpt = round(gmean[index], ndigits = 4)
fprOpt = round(fpr[index], ndigits = 4)
tprOpt = round(tpr[index], ndigits = 4)
print('Best Threshold: {} with G-Mean: {}'.format(thresholdOpt, gmeanOpt))
print('FPR: {}, TPR: {}'.format(fprOpt, tprOpt))
```




```
Best Threshold: 0.0909 with G-Mean: 0.8486
```

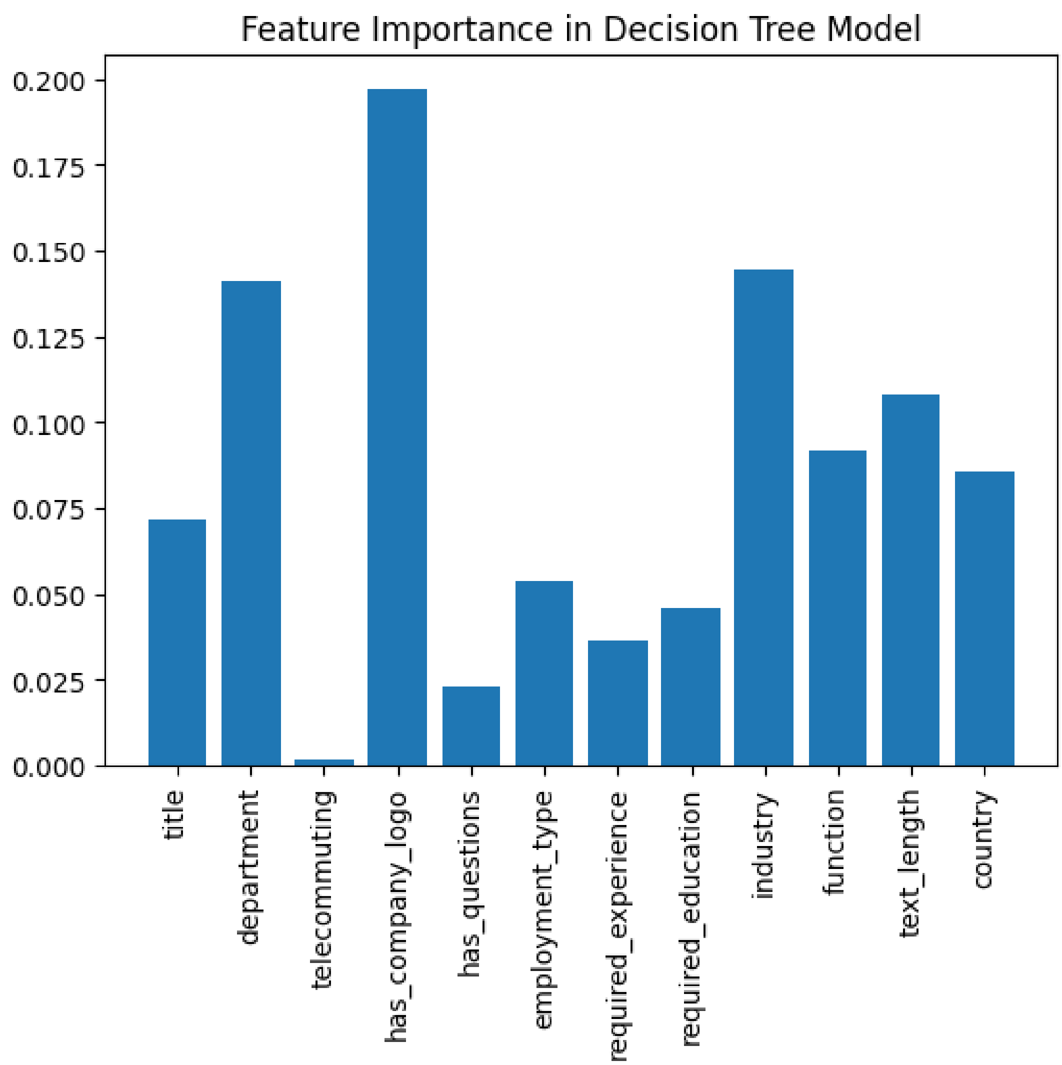
```
FPR: 0.0923, TPR: 0.7934
```

[30]:
```python
#Predict the response for test dataset
# select the right threshold to make sure the recall of "1" category is higher
threshold = 0.0909
y_pred = (dtm.predict_proba(X_test)[:, 1] > threshold).astype('float')

dtm_matrix = metrics.confusion_matrix(y_test, y_pred)
print(dtm_matrix)
dtm_report = metrics.classification_report(y_test,y_pred)
print(dtm_report)
```

```
[[4623  470]
 [  56  215]]
              precision    recall  f1-score   support

           0       0.99      0.91      0.95      5093
           1       0.31      0.79      0.45       271

    accuracy                           0.90      5364
   macro avg       0.65      0.85      0.70      5364
weighted avg       0.95      0.90      0.92      5364
```

[35]:
```python
resultdict = {}
for i in range(len(feature_cols)):
    resultdict[feature_cols[i]] = dtm.feature_importances_[i]

plt.bar(resultdict.keys(),resultdict.values())
plt.xticks(rotation='vertical')
plt.title('Feature Importance in Decision Tree Model')
```

[35]:
```
Text(0.5, 1.0, 'Feature Importance in Decision Tree Model')
```

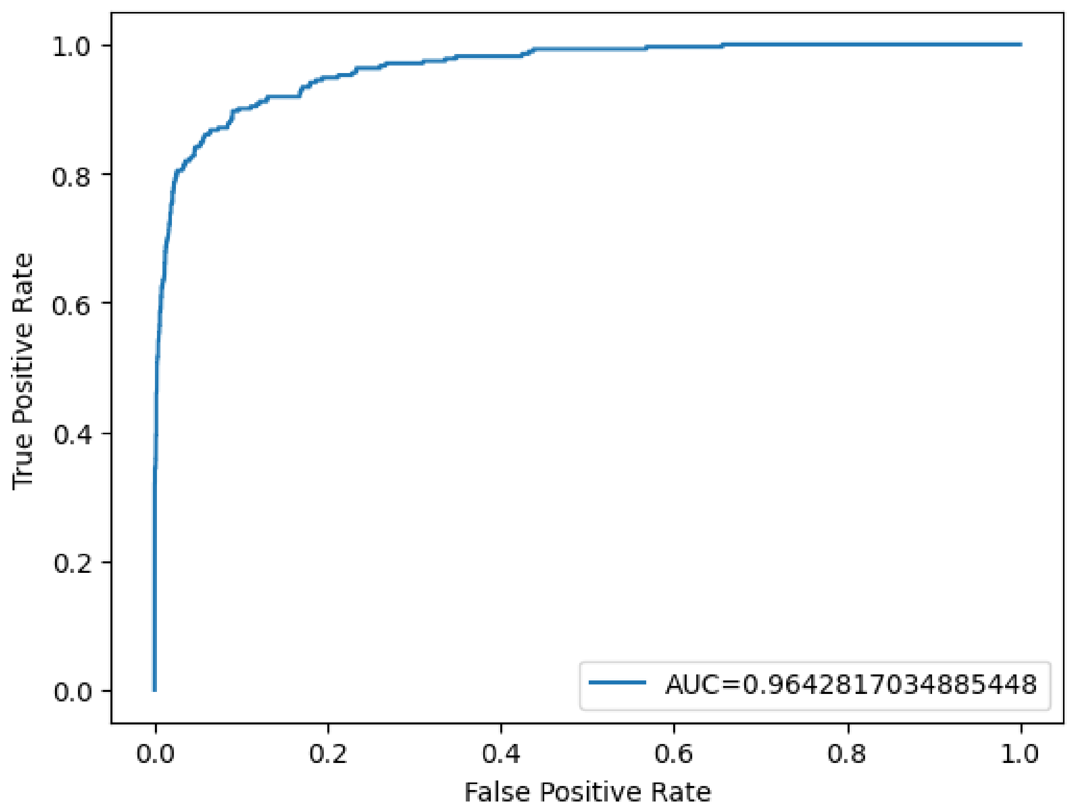


[36]:
```
rf = RandomForestClassifier(random_state = 42, n_estimators=100, bootstrap =␣
↪True, max_depth=10,criterion='entropy')
rf.fit(X_train, y_train)
```

[36]:
```
RandomForestClassifier(criterion='entropy', max_depth=10, random_state=42)
```

[37]:
```
#define metrics

y_pred_proba = rf.predict_proba(X_test)[:,1]
fpr, tpr, thresholds = metrics.roc_curve(y_test,  y_pred_proba)
auc = metrics.roc_auc_score(y_test, y_pred_proba)

#create ROC curve
plt.plot(fpr,tpr,label="AUC="+str(auc))
```

```
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.legend(loc=4)
plt.show()

# Calculate the G-mean
gmean = np.sqrt(tpr * (1 - fpr)) # using G-mean

# Find the optimal threshold
index = np.argmax(gmean)
thresholdOpt = round(thresholds[index], ndigits = 4)
gmeanOpt = round(gmean[index], ndigits = 4)
fprOpt = round(fpr[index], ndigits = 4)
tprOpt = round(tpr[index], ndigits = 4)
print('Best Threshold: {} with G-Mean: {}'.format(thresholdOpt, gmeanOpt))
print('FPR: {}, TPR: {}'.format(fprOpt, tprOpt))
```




```
Best Threshold: 0.0963 with G-Mean: 0.9031
FPR: 0.0905, TPR: 0.8967
```

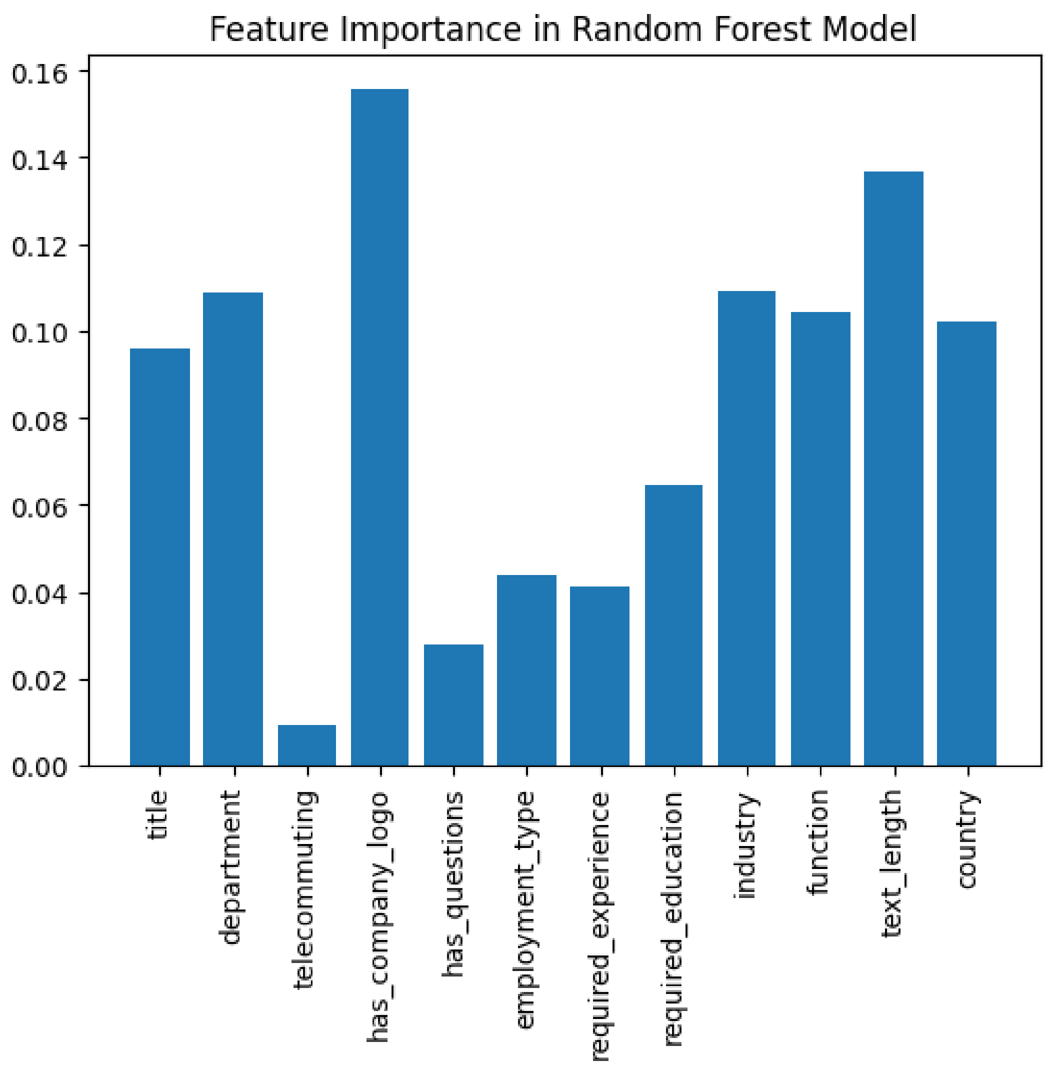
[38]:
```
# Create the parameter grid based on the results of random search

param_grid = {
    'bootstrap': [True],
    'max_depth': [5, 10, 20],
    'n_estimators': [100, 200, 300]
}
# Create a based model
rf_t = RandomForestClassifier(random_state = 42)
# Instantiate the grid search model
grid_search = GridSearchCV(estimator = rf_t, param_grid = param_grid,
                          cv = 3, n_jobs = -1, verbose = 2, scoring = 'roc_auc')
```

[39]:
```
y_pred2 = rf.predict(X_test)
rf_matrix = metrics.confusion_matrix(y_test, y_pred2)
print(rf_matrix)

rf_report = metrics.classification_report(y_test,y_pred2)
print(rf_report)
```

```
[[5088    5]
 [ 178   93]]
              precision    recall  f1-score   support

           0       0.97      1.00      0.98      5093
           1       0.95      0.34      0.50       271

    accuracy                           0.97      5364
   macro avg       0.96      0.67      0.74      5364
weighted avg       0.97      0.97      0.96      5364
```

[40]:
```
resultdict = {}
for i in range(len(feature_cols)):
    resultdict[feature_cols[i]] = rf.feature_importances_[i]

plt.bar(resultdict.keys(),resultdict.values())
plt.xticks(rotation='vertical')
plt.title('Feature Importance in Random Forest Model')
```

[40]: `Text(0.5, 1.0, 'Feature Importance in Random Forest Model')`



1.4.2 Data Modeling: part 2

1.4.3 Text Cleaning and Text Mining

[5]:
```
import re
import nltk
from nltk.corpus import stopwords
from nltk.stem import SnowballStemmer
from nltk.tokenize import word_tokenize
```

[11]:
```
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /home/repl/nltk_data…
[nltk_data]   Package stopwords is already up-to-date!
```

[11]: True

[13]:
```
stop=set(stopwords.words("english"))
```

[14]:
```
def clean(text):

    text=text.lower()
    obj=re.compile(r"<.*?>")                      #removing html tags
    text=obj.sub(r" ",text)
    obj=re.compile(r"https://\S+|http://\S+")     #removing url
    text=obj.sub(r" ",text)
    obj=re.compile(r"[^\w\s]")                    #removing punctuations
    text=obj.sub(r" ",text)
    obj=re.compile(r"\d{1,}")                     #removing digits
    text=obj.sub(r" ",text)
    obj=re.compile(r"_+")                         #removing underscore
    text=obj.sub(r" ",text)
    obj=re.compile(r"\s\w\s")                     #removing single character
    text=obj.sub(r" ",text)
    obj=re.compile(r"\s{2,}")                     #removing multiple spaces
    text=obj.sub(r" ",text)


    stemmer = SnowballStemmer("english")
    text=[stemmer.stem(word) for word in text.split() if word not in stop]

    return " ".join(text)
```

[15]:
```
import pandas as pd
data = pd.read_csv('data1.csv')
```

[16]:
```
data["text"]=data["text"].apply(clean)
```

[55]:
```
from wordcloud import WordCloud,STOPWORDS
from collections import defaultdict
from nltk import ngrams
```

[56]:
```
def generate(text,ngram):
    n_grams=ngrams(word_tokenize(text),ngram)
    grams=[" ".join(val) for val in n_grams]
    return grams
```

[57]:
```
real_job=data[data["fraudulent"]==0]["text"].values
```

[58]:
```
wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = STOPWORDS).generate(str(real_job))
```

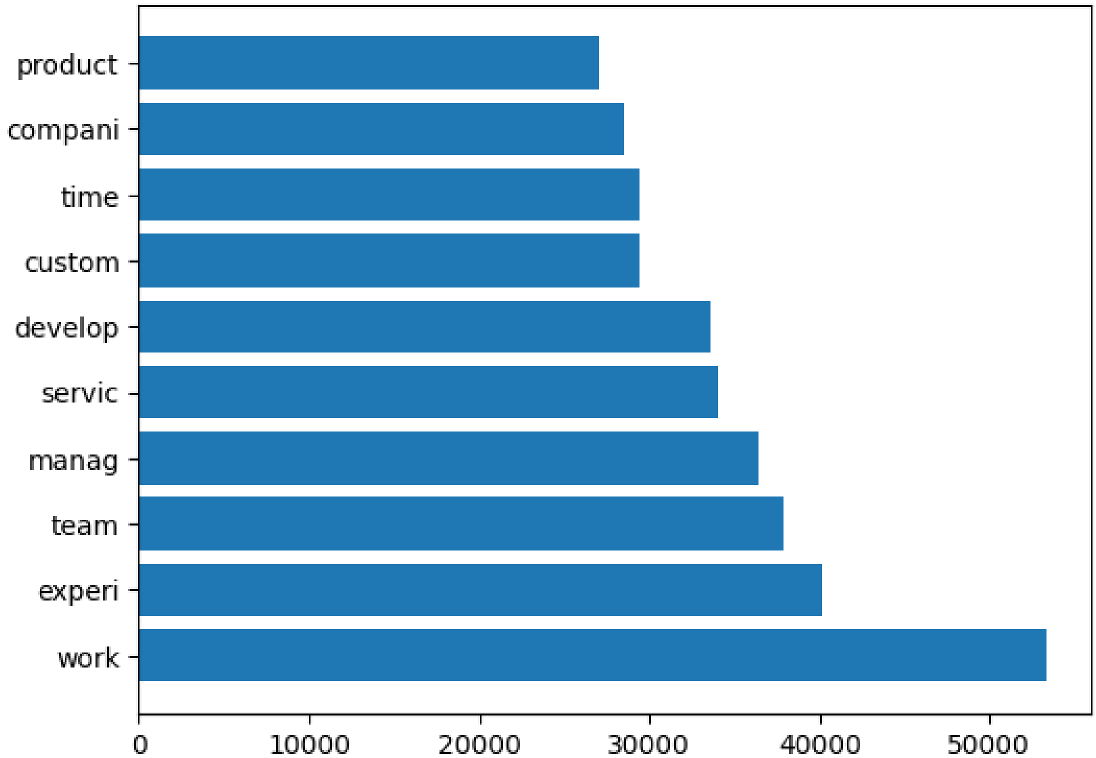
```
plt.imshow(wordcloud, interpolation = 'bilinear')
plt.axis('off');
```

[59]:
```
nltk.download('punkt')
```

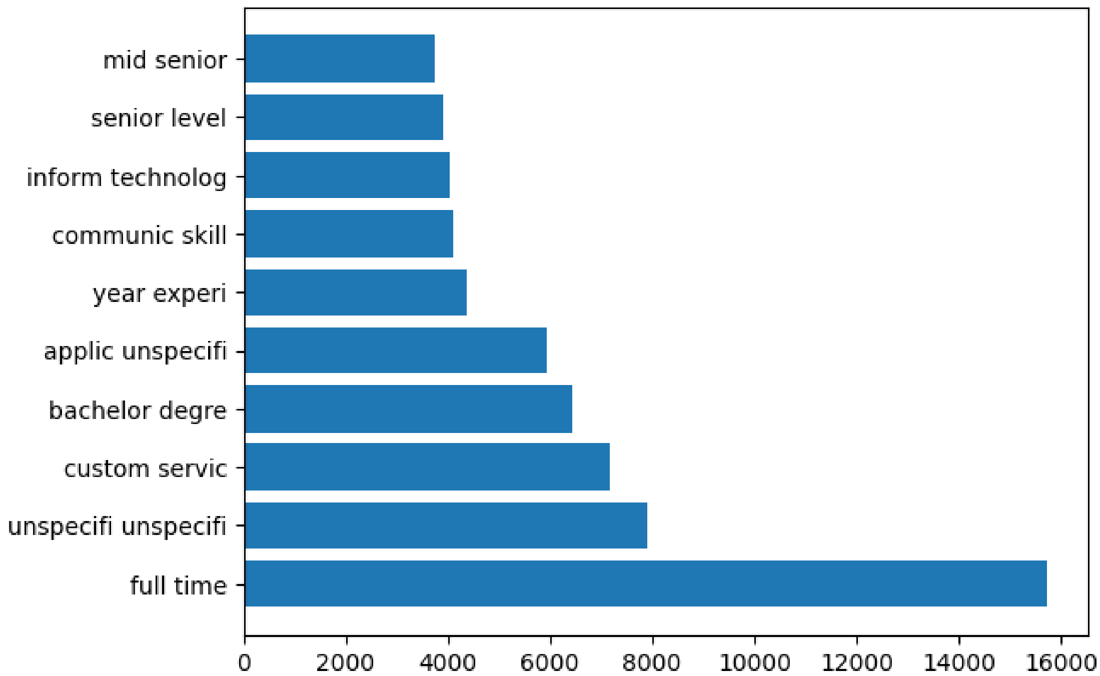
```
[nltk_data] Downloading package punkt to /home/repl/nltk_data…
[nltk_data]   Package punkt is already up-to-date!
```

[59]: True

[60]:
```
pos_1=defaultdict(int)

for text in data[data["fraudulent"]==0]["text"].values:
    for words in generate(text,1):
        pos_1[words]+=1

pos_1=pd.DataFrame(sorted(pos_1.items(),key=lambda x: x[1],reverse=True))
plt.barh(pos_1[0][:10],pos_1[1][:10])
plt.show()
```



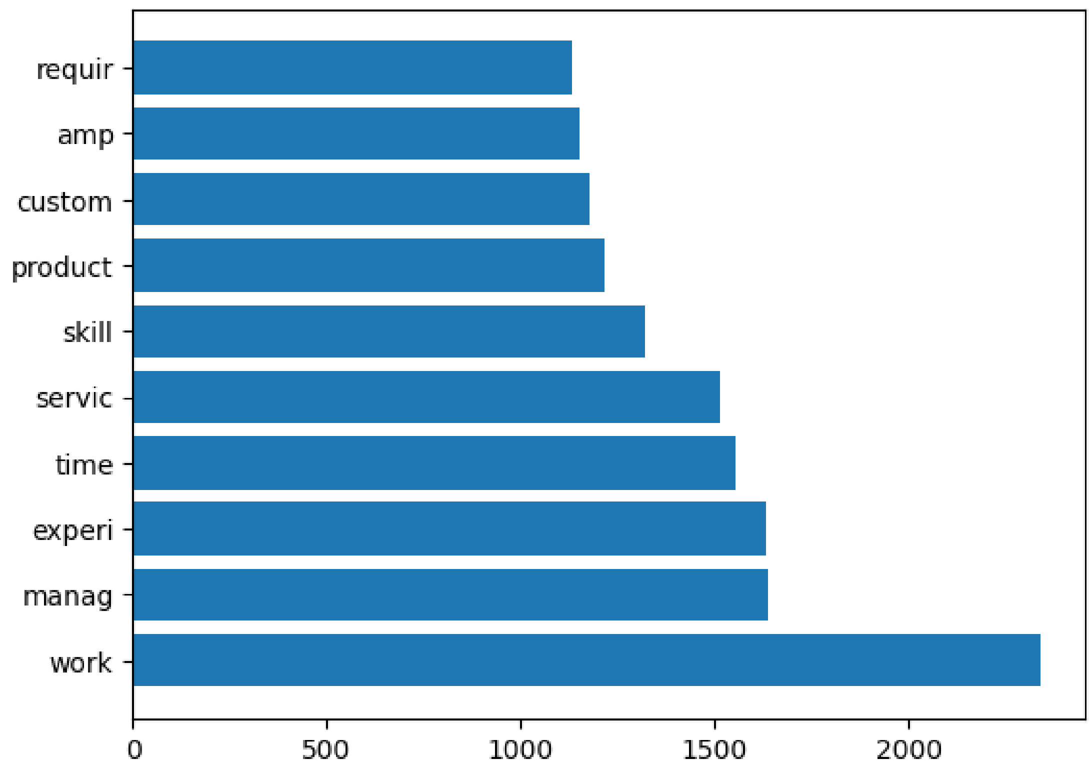
[61]:
```
pos_2=defaultdict(int)

for text in data[data["fraudulent"]==0]["text"]:
    for words in generate(text,2):
        pos_2[words]+=1

pos=pd.DataFrame(sorted(pos_2.items(),key=lambda x: x[1],reverse=True))
plt.barh(pos[0][:10],pos[1][:10])
plt.show()
```



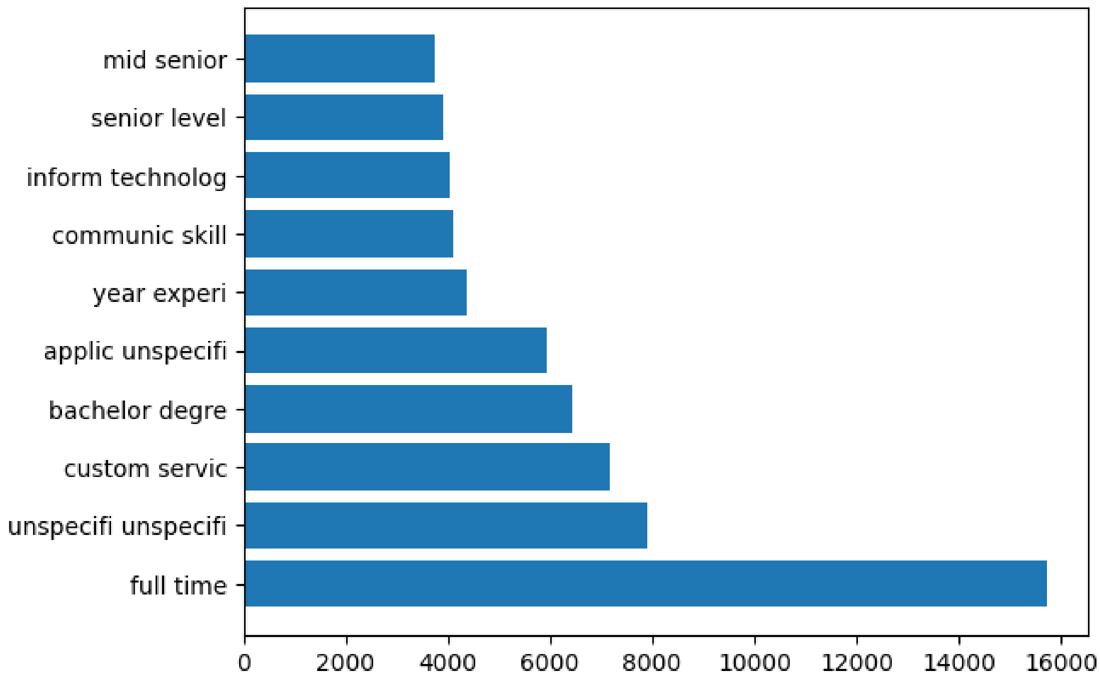
[62]:
```
fake_job=data[data["fraudulent"]==1]["text"].values
```

[63]:
```
wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = STOPWORDS).generate(str(fake_job))
plt.imshow(wordcloud, interpolation = 'bilinear')
plt.axis('off');
```

[64]:
```python
neg_1=defaultdict(int)

for text in data[data["fraudulent"]==1]["text"].values:
    for words in generate(text,1):
        neg_1[words]+=1

neg=pd.DataFrame(sorted(neg_1.items(),key=lambda x: x[1],reverse=True))
plt.barh(neg[0][:10],neg[1][:10])
plt.show()
```



[65]:
```
neg_2=defaultdict(int)

for text in data[data["fraudulent"]==0]["text"].values:
    for words in generate(text,2):
        neg_2[words]+=1

neg=pd.DataFrame(sorted(neg_2.items(),key=lambda x: x[1],reverse=True))
plt.barh(neg[0][:10],neg[1][:10])
plt.show()
```



1.5 Data Modeling

[19]:
```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
```

[20]:
```
vectorizer=TfidfVectorizer(strip_accents='unicode',
                           analyzer='word',
                           ngram_range=(1, 2),
                           max_features=15000,
                           smooth_idf=True,
                           sublinear_tf=True)

vectorizer.fit(data["text"])
X = vectorizer.transform(data["text"])
```

[21]:
```
X.shape
```

[21]: (17880, 15000)

[22]:
```
y=data["fraudulent"]
X = X.toarray()

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20,␣
 ↪random_state=42)
```

[23]:
```
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import␣
 ↪accuracy_score,classification_report,confusion_matrix
```

[ ]:
```
```

1.6 Conclusion

[83]:
```
```

1.7 References

- https://www.fbi.gov/contact-us/field-offices/elpaso/news/press-releases/fbi-warns-cyber-criminals-are-using-fake-job-listings-to-target-applicants-personally-identifiable-information
- http://emscad.samos.aegean.gr/
- https://www.kaggle.com/datasets/shivamb/real-or-fake-fake-jobposting-prediction
-