```
pip install wordcloud
```

```
!pip install -U spacy
```

```
import re
import string
import numpy as np
import pandas as pd
import random
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.pipeline import Pipeline
from sklearn.base import TransformerMixin
from sklearn.metrics import accuracy_score, plot_confusion_matrix, classification_report, confusion_matrix
from wordcloud import WordCloud
import spacy
from spacy.lang.en.stop_words import STOP_WORDS
from spacy.lang.en import English
```

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

```
    Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
df = pd.read_csv('/content/drive/MyDrive/Sem 6/Mini Project/Material Project/fake_job_postings.csv')
```

```
df.head()
```

	job_id	title	location	department	salary_range	company_profile	description	requirements
0	1	Marketing Intern	US, NY, New York	Marketing	NaN	We're Food52, and we've created a groundbreaki...	Food52, a fast-growing, James Beard Award-winn...	Experience with content management systems a m...
1	2	Customer Service - Cloud Video Production	NZ, , Auckland	Success	NaN	90 Seconds, the worlds Cloud Video Production ...	Organised - Focused - Vibrant - Awesome!Do you...	What we expect from you:Your key responsibilit...
2	3	Commissioning Machinery Assistant (CMA)	US, IA, Wever	NaN	NaN	Valor Services provides Workforce Solutions th...	Our client, located in Houston, is actively se...	Implement pre-commissioning and commissioning ...
3	4	Account Executive - Washington DC	US, DC, Washington	Sales	NaN	Our passion for improving quality of life thro...	THE COMPANY: ESRI – Environmental Systems Rese...	EDUCATION: Bachelor's or Master's in GIS, busi...
4	5	Bill Review Manager	US, FL, Fort Worth	NaN	NaN	SpotSource Solutions LLC is a Global Human Cap...	JOB TITLE: Itemization Review ManagerLOCATION:...	QUALIFICATIONS:RN license in the State of Texa...

```
df.shape
```

```
    (17880, 18)
```

```
# c1 = 16884
# c2 = 756
# for ind in df.index:
#   if df['fraudulent'][ind] == 0:
#     df = df.drop(index = ind)
#     if(c1 > 0):
#       c1 -= 1
#     else:
```

```
#       break

# for ind in df.index:
#   if df['fraudulent'][ind] == 1:
#     df = df.drop(index = ind)
#     if(c2 > 0):
#       c2 -= 1
#     else:
#       break
```

```
columns = ['job_id', 'telecommuting', 'has_company_logo', 'has_questions', 'salary_range', 'employment_type']
for colu in columns:
  del df[colu]
```

```
df.isnull().sum()
```

```
title                      0
location                 346
department             11547
company_profile         3308
description                1
requirements            2695
benefits                7210
required_experience     7050
required_education      8105
industry                4903
function                6455
fraudulent                 0
dtype: int64
```

```
df.head()
```

	title	location	department	company_profile	description	requirements	benefits	required_expe
0	Marketing Intern	US, NY, New York	Marketing	We're Food52, and we've created a groundbreaki...	Food52, a fast-growing, James Beard Award-winn...	Experience with content management systems a m...	NaN	In
1	Customer Service - Cloud Video Production	NZ, , Auckland	Success	90 Seconds, the worlds Cloud Video Production ...	Organised - Focused - Vibrant - Awesome!Do you...	What we expect from you:Your key responsibilit...	What you will get from usThrough being part of...	Not Ap
2	Commissioning Machinery Assistant (CMA)	US, IA, Wever	NaN	Valor Services provides Workforce Solutions th...	Our client, located in Houston, is actively se...	Implement pre-commissioning and commissioning ...	NaN	
3	Account Executive - Washington DC	US, DC, Washington	Sales	Our passion for improving quality of life thro...	THE COMPANY: ESRI – Environmental Systems Rese...	EDUCATION: Bachelor's or Master's in GIS, busi...	Our culture is anything but corporate—we have ...	Mid-Ser
4	Bill Review Manager	US, FL, Fort Worth	NaN	SpotSource Solutions LLC is a Global Human Cap...	JOB TITLE: Itemization Review ManagerLOCATION:...	QUALIFICATIONS:RN license in the State of Texa...	Full Benefits Offered	Mid-Ser

```
df.fillna('', inplace=True)
```

```
plt.figure(figsize=(15,5))
sns.countplot(y='fraudulent', data=df)
plt.show()
```

0
fraudulent

```
#displays the count of real and fake jobs in fradulent column
df.groupby('fraudulent')['fraudulent'].count()
```

```
fraudulent
0    17014
1      866
Name: fraudulent, dtype: int64
```

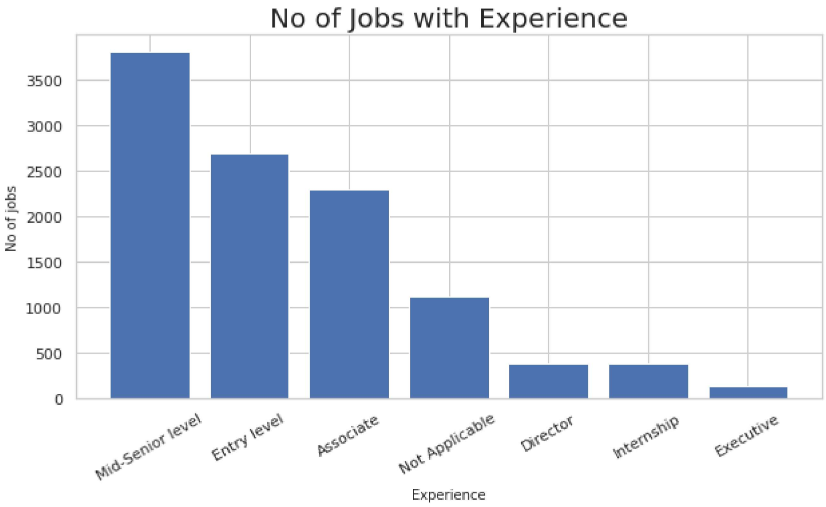
```
dff1 = df
```

```
exp = dict(df.required_experience.value_counts())
del exp['']
```

```
exp
```

```
{'Associate': 2297,
 'Director': 389,
 'Entry level': 2697,
 'Executive': 141,
 'Internship': 381,
 'Mid-Senior level': 3809,
 'Not Applicable': 1116}
```

```
plt.figure(figsize=(10, 5))
sns.set_theme(style="whitegrid")
plt.bar(exp.keys(), exp.values())
plt.title('No of Jobs with Experience', size = 20)
plt.xlabel('Experience', size = 10)
plt.ylabel('No of jobs', size = 10)
plt.xticks(rotation = 30)
plt.show()
```



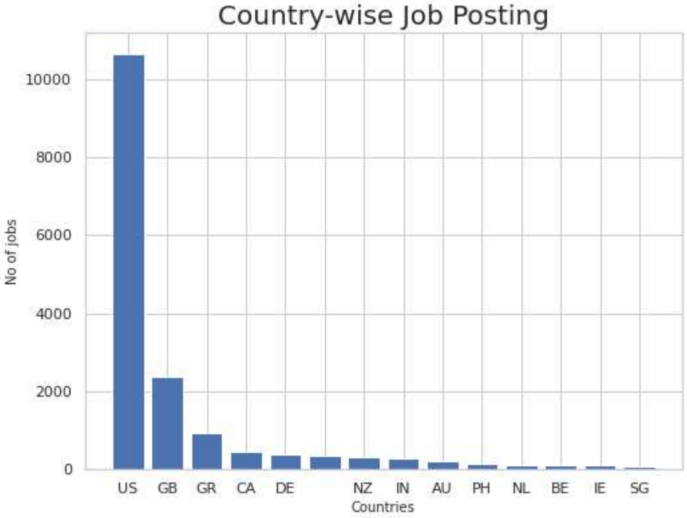
```
def split(location):
  l = location.split(',')
  return l[0]
df['country'] = df.location.apply(split)
```

```
country = dict(df.country.value_counts()[:14])
del country['']
country
```

```
     {'AU': 214,
      'BE': 117,
      'CA': 457,
      'DE': 383,
      'GB': 2384,
      'GR': 940,
      'IE': 114,
      'IN': 276,
      'NL': 127,
      'NZ': 333,
      'PH': 132,
      'SG': 80,
      'US': 10656}
```

```
plt.figure(figsize=(8, 6))
plt.title('Country-wise Job Posting', size = 20)
plt.bar(countr.keys(), countr.values())
plt.xlabel('Countries', size = 10)
plt.ylabel('No of jobs', size = 10)
```

```
     Text(0, 0.5, 'No of jobs')
```



```
edu = dict(df.required_education.value_counts()[:7])
del edu['']
edu
```

```
     {'Associate Degree': 274,
      "Bachelor's Degree": 5145,
      'Certification': 170,
      'High School or equivalent': 2080,
      "Master's Degree": 416,
      'Unspecified': 1397}
```

```
plt.figure(figsize=(15, 6))
plt.title('Job Posting Based on Education', size = 20)
plt.bar(edu.keys(), edu.values())
plt.xlabel('Education', size = 10)
plt.ylabel('No of jobs', size = 10)
```

```
Text(0, 0.5, 'No of jobs')
```

```
print(df[df.fraudulent == 0].title.value_counts()[:10])
```

```
English Teacher Abroad                                311
Customer Service Associate                            146
Graduates: English Teacher Abroad (Conversational)    144
English Teacher Abroad                                 95
Software Engineer                                      86
English Teacher Abroad (Conversational)                83
Customer Service Associate - Part Time                 76
Account Manager                                        73
Web Developer                                          66
Project Manager                                        62
Name: title, dtype: int64
```

```
print(df[df.fraudulent == 1].title.value_counts()[:10])
```

```
Data Entry Admin/Clerical Positions - Work From Home                21
Home Based Payroll Typist/Data Entry Clerks Positions Available     21
Cruise Staff Wanted *URGENT*                                        21
Customer Service Representative                                     17
Administrative Assistant                                            16
Home Based Payroll Data Entry Clerk Position - Earn $100-$200 Daily 12
Account Sales Managers $80-$130,000/yr                              10
Network Marketing                                                   10
Payroll Clerk                                                       10
Payroll Data Coordinator Positions - Earn $100-$200 Daily           10
Name: title, dtype: int64
```

```
df['text']=df['title'] + ' ' + df['company_profile'] + ' ' + df['description'] + ' ' + df['requirements'] + ' ' + df['benefits']
del df['title']
del df['location']
del df['department']
del df['company_profile']
del df['description']
del df['requirements']
del df['benefits']
del df['required_experience']
del df['required_education']
del df['industry']
del df['function']
del df['country']
```

```
df.head()
```

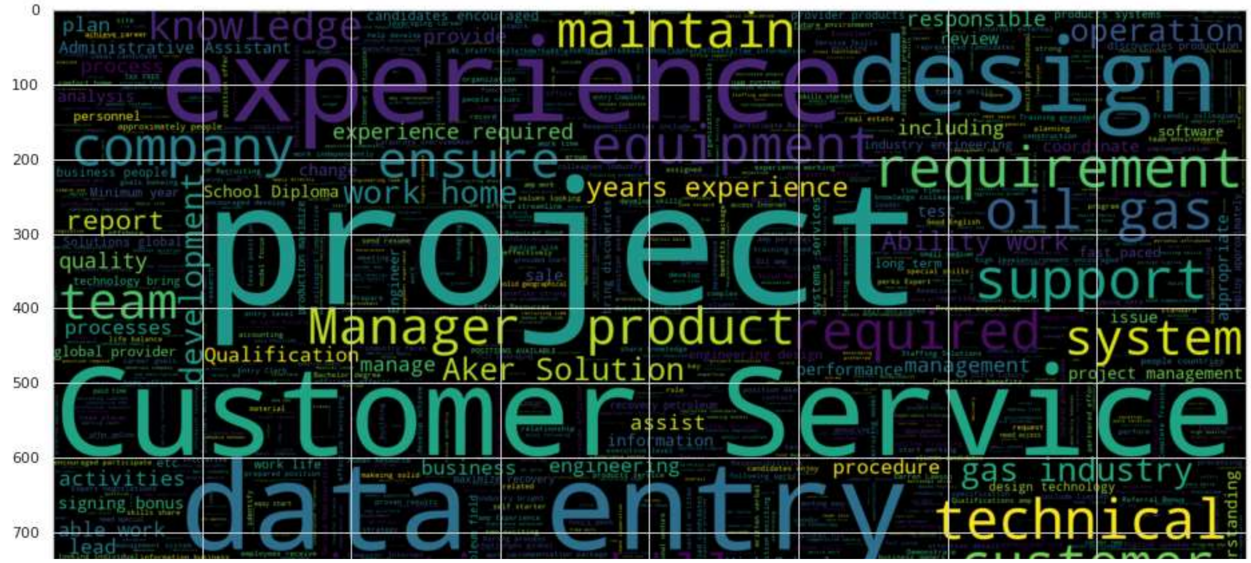
	fraudulent	text
0	0	Marketing Intern We're Food52, and we've creat...
1	0	Customer Service - Cloud Video Production 90 S...
2	0	Commissioning Machinery Assistant (CMA) Valor ...
3	0	Account Executive - Washington DC Our passion ...
4	0	Bill Review Manager SpotSource Solutions LLC i...

```
# dff1 = df
```

```
fraudjobs_text = df[df.fraudulent == 1].text
realjobs_text = df[df.fraudulent == 0].text
```

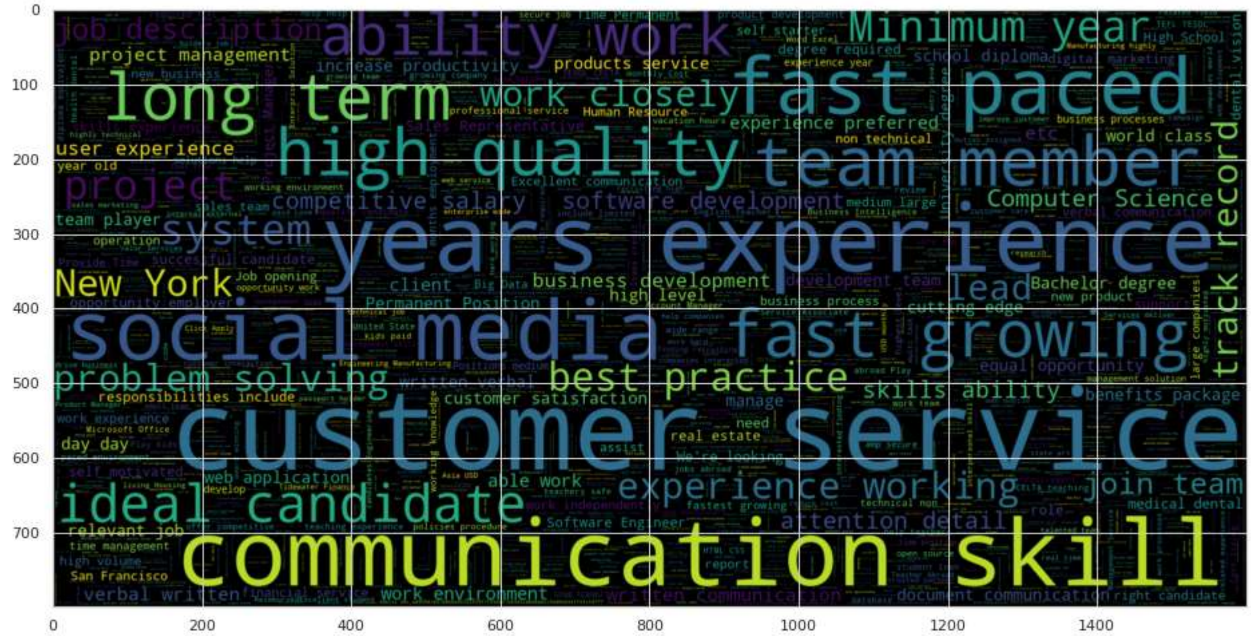
```
STOPWORDS = spacy.lang.en.stop_words.STOP_WORDS
plt.figure(figsize = (16,14))
wc = WordCloud(min_font_size = 3, max_words = 3000, width = 1600, height = 800, stopwords = STOPWORDS).generate(str(" ".join(fraudjobs_text))
plt.imshow(wc, interpolation = 'bilinear')
```

```
<matplotlib.image.AxesImage at 0x7fd1a59ed250>
```



```
STOPWORDS = spacy.lang.en.stop_words.STOP_WORDS
plt.figure(figsize = (16,14))
wc = WordCloud(min_font_size = 3, max_words = 3000, width = 1600, height = 800, stopwords = STOPWORDS).generate(str(" ".join(realjobs_text)))
plt.imshow(wc, interpolation = 'bilinear')
```

```
<matplotlib.image.AxesImage at 0x7fd1a581d090>
```



....

```
!pip install spacy && python -m spacy download en
```

```
# punctuation = string.punctuation

# nlp = spacy.load("en_core_web_sm")
# stop_words = spacy.lang.en.stop_words.STOP_WORDS

# parser = English()

# def spacy_tokenizer(sentence):

#   punctuation = string.punctuation

#   nlp = spacy.load("en_core_web_sm")
#   stop_words = spacy.lang.en.stop_words.STOP_WORDS
#   parser = English()
#   mytokens = parser(sentence)
#   # mytokens = [word.lemma_.lower().strip() if word.lemma_ != "-PRON-" else word.lower_ for word in mytokens]
#   mytokens = [word for word in mytokens if word not in stop_words and word not in punctuation]
#   return mytokens

# class predictors(TransformerMixin):
#   def tranform(self, X, **transform_params):
#     return [clean_text(text) for text in X]

#   def fit(self, X, y=None, **fit_params):
#     return self

#   def get_params(self, deep = True):
#     return {}

def clean_text(text):
  return text.strip().lower()

from nltk.corpus import stopwords
import nltk
nltk.download('punkt')
nltk.download('stopwords')
from nltk.tokenize import word_tokenize

def spacy_tokenizer(text):
  print(1)
  text_tokens = word_tokenize(text)
  tokens_without_sw = [word for word in text_tokens if not word in stopwords.words()]
  filtered_sentence = (" ").join(tokens_without_sw)
  return filtered_sentence
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

```
df['text'] = df['text'].apply(clean_text)
```

```
# dff2 = df
```

```
df['text'] = df['text'].apply(spacy_tokenizer)
```

```
# dff3 = df
```

```
# for ind in df.index:
#   if df['fraudulent'][ind] == 1:
#     print(df['text'][ind])
#     print()
```

```
cv = TfidfVectorizer(max_features = 100)
x = cv.fit_transform(df['text'])
df1 = pd.DataFrame(x.toarray(), columns = cv.get_feature_names())

# df1

df.drop(["text"], axis = 1, inplace = True)

# df

main_df = pd.concat([df1, df], axis = 1)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function get_feature_names is deprecated; get_fea
  warnings.warn(msg, category=FutureWarning)
```

```
# l2 = []
# for ind in df.index:
#   l2.append(df['fraudulent'][ind])
# main_df = df1.assign(fraudulent = l2)
# dff4 = main_df
```

```
Y = main_df.iloc[:,-1]
X = main_df.iloc[:,:-1]

X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.2)

print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)
```

```
     (14304, 100)
     (14304,)
     (3576, 100)
     (3576,)
```

```
X_test
```

```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(n_jobs = 3, oob_score = True, n_estimators = 100, criterion = "entropy")
model = rfc.fit(X_train, y_train)
```

```
pred = rfc.predict(X_test)
score = accuracy_score(y_test, pred)
score

     0.9703579418344519
```

```
      47  0.158051  0.152808  0.156264  0.000000  0.654673  0.000000  0.000000     0.000000     0.103562  0.151967     0.164645  0.
```

```
print("Classification Report\n")
print(classification_report(y_test, pred))
print("Confusion Matrix\n")
print(confusion_matrix(y_test, pred))

     Classification Report

                   precision    recall  f1-score   support

                0       0.97      1.00      0.98      3403
                1       0.99      0.39      0.56       173

         accuracy                           0.97      3576
        macro avg       0.98      0.70      0.77      3576
     weighted avg       0.97      0.97      0.96      3576

     Confusion Matrix

     [[3402    1]
      [ 105   68]]
```

Oversampling

```
      108  0.000000  0.000000  0.000000  0.434832  0.000000  0.000000  0.000000     0.000000     0.332515  0.000000     0.000000  0.
```

```
Y = main_df.iloc[:,-1]
X = main_df.iloc[:,:-1]

df_train, df_test = train_test_split(main_df, test_size = 0.2)
```

```
df_train.groupby('fraudulent')['fraudulent'].count()

     fraudulent
     0    13601
     1      703
     Name: fraudulent, dtype: int64
```

```
      187  0.000000  0.000000  0.000000  0.601064  0.000000  0.000000  0.000000     0.548020     0.229804  0.168608     0.000000  0.
```

```
df_test.groupby('fraudulent')['fraudulent'].count()

     fraudulent
     0    3413
     1     163
     Name: fraudulent, dtype: int64
```

```
from sklearn.utils import resample

df_majority = df_train[(df_train['fraudulent']==0)]
df_minority = df_train[(df_train['fraudulent']==1)]

df_minority_upsampled = resample(df_minority,
                                 replace=True,
                                 n_samples= 13601,
                                 random_state=42)

df_upsampled = pd.concat([df_minority_upsampled, df_majority])
```

```
      104  0.574381  0.000000  0.283944  0.246098  0.000000  0.000000  0.000000     0.000000     0.000000  0.352273     0.000000  0.
```

```
df_upsampled.groupby('fraudulent')['fraudulent'].count()

     fraudulent
     0    13601
     1    13601
     Name: fraudulent, dtype: int64
```

```
Y = df_upsampled.iloc[:,-1]
X = df_upsampled.iloc[:,:-1]

X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.2)

print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)

     (21761, 100)
     (21761,)
     (5441, 100)
     (5441,)
```

```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(n_jobs = 3, oob_score = True, n_estimators = 100, criterion = "entropy")
model = rfc.fit(X_train, y_train)
```

```
pred = rfc.predict(X_test)
score = accuracy_score(y_test, pred)
score

     0.9983458922992097
```

```
print("Classification Report\n")
print(classification_report(y_test, pred))
+print("Confusion Matrix\n")
print(confusion_matrix(y_test, pred))

     Classification Report

                   precision    recall  f1-score   support

                0       1.00      1.00      1.00      2685
                1       1.00      1.00      1.00      2756

         accuracy                           1.00      5441
        macro avg       1.00      1.00      1.00      5441
     weighted avg       1.00      1.00      1.00      5441

     Confusion Matrix

     [[2676    9]
      [   0 2756]]
```

....

```
# txt = "marketing intern we're food52, and we've created a groundbreaking and award-winning cooking site. we support, connect, and celebrate

txt = "Facilities Development Engineer Aker Solutions is a global provider of products, systems and services to the oil and gas industry. Our

# txt = "Customer service/ Data Entry Customer Service SpecialistWe are currently looking for talented and creative individuals to continue g

# txt = "Process Engineer JOB DESCRIPTION: PROCESS ENGINEER Process EngineerProvide process engineering support to unit operations. Troublesh

txt = clean_text(txt)
# txt1 = clean_text(txt1)
```

```
list = [txt]
inp = pd.DataFrame()
inp["text"] = list

inp
```

	text
0	facilities development engineer aker solutions...

```
cv = TfidfVectorizer(max_features = 20)
x1 = cv.fit_transform(inp['text'])
inp1 = pd.DataFrame(x1.toarray(), columns = cv.get_feature_names())
# inp.drop(["text"], axis = 1, inplace = True)
# main_inp = pd.concat([inp1, inp], axis = 1)

     /usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function get_feature_names is deprecated; get_fea
       warnings.warn(msg, category=FutureWarning)
```

```
# main_inp
inp1
```

	aker	all	and	concept	development	engineering	facilities	field
0	0.056176	0.056176	0.853879	0.078647	0.168529	0.089882	0.089882	0.089882

```
inp_pred = rfc.predict(inp1)
```

```
inp_pred[0]
```

```
0
```

```
# import pickle
# filename = 'model_2.pkl'
# pickle.dump(rfc, open(filename, 'wb'))
```

```
# Customer Service - Cloud Video Production
# 90 Seconds, the worlds Cloud Video Production Service.90 Seconds is the worlds Cloud Video Production Service enabling brands and agencies
# Organised - Focused - Vibrant - Awesome!Do you have a passion for customer service? Slick typing skills? Maybe Account Management? ...And t
# What we expect from you:Your key responsibility will be to communicate with the client, 90 Seconds team and freelance community throughout
# What you will get from usThrough being part of the 90 Seconds team you will gain:experience working on projects located around the world wi
```

```
# import pickle

# with open('model_2.pkl' , 'rb') as f:
#     rfc = pickle.load(f)
```