NLP

October 17, 2023

[1]:
```python
import pandas as pd

#unlabeled dataset
jobs = pd.read_csv("/Users/poojagrewal/Downloads/remote_jobs - remote_jobs.csv")
print(jobs.head())
print(jobs.shape)
```

```
                                                 url     company_name  \
0  https://nodesk.co/remote-jobs/activecampaign-e…  ActiveCampaign
1  https://nodesk.co/remote-jobs/betterup-marketi…        BetterUp
2  https://nodesk.co/remote-jobs/siege-media-digi…     Siege Media
3  https://nodesk.co/remote-jobs/gusto-head-of-pr…           Gusto
4  https://nodesk.co/remote-jobs/myfbaprep-partne…       MyFBAPrep

                                         job_title  \
0  Event Marketing and Brand Activation Manager
1    Marketing Manager, Conversion Optimization
2                           Digital PR Specialist
3           Head of Product Marketing & Revenue
4                      Partner Marketing Manager

                                      company_info       country  \
0  Grow your business with customer experience au…            US
1                          We're reworking how you work           US
2  Brands trust us to deliver best-in-class conte…            US
3  The all-in-one people platform for payroll, be…            US
4          Make Logistics Your Competitive Advantage  Remote-First

     job_type industries       salary  \
0  Internship  Marketing            0
1   Full-Time  Marketing            0
2   Full-Time  Marketing  $60K - $69K
3   Full-Time  Marketing            0
4    Contract  Marketing            0

                                          skills  \
0                        Event Marketing, Non-Tech
1                      Marketing Manager, Non-Tech
```

```
2                          Non-Tech, Public Relations
3  Marketing Manager, Non-Tech, Product, Product …
4  Co-Marketing, Marketing Manager, Non-Tech, Par…

                                                  points
0                                                    NaN
1                                                    NaN
2                                                    NaN
3                                                    NaN
4  Plan, coordinate, and execute marketing activi…
(400, 10)
```

[2]:
```
#labeled dataset
SSOC = pd.read_csv("/Users/poojagrewal/Downloads/SSOC2020 Alphabetical Index  -␣
 ↪SSOC 2020 Alpha Index.csv")
SSOC = SSOC.rename(columns={'SSOC 2020 Alphabetical Index Description':␣
 ↪'job_title'})
SSOC = SSOC[['SSOC 2020', 'job_title']]
print(SSOC.head())
```

```
  SSOC 2020                job_title
0     11110  Legislator (government)
1     11110      Member of parliament
2     11110    Minister (government)
3     11110           Parliamentarian
4     11110   President (government)
```

[14]:
```
import pandas as pd
import re
import nltk
from nltk.corpus import stopwords
nltk.download('punkt')
nltk.download('wordnet')
from nltk.tokenize import word_tokenize


# Clean and preprocess job_title column of labeled dataset
SSOC['job_title'] = SSOC['job_title'].apply(lambda x: re.sub(r'[^a-zA-Z\s]',␣
 ↪'', str(x))) # remove special characters
SSOC['job_title'] = SSOC['job_title'].apply(lambda x: ' '.join([word.lower()␣
 ↪for word in x.split() if word.lower() not in stopwords.words('english')])) #␣
 ↪remove stopwords and convert to lowercase

# Clean and preprocess job_title column of non-labeled dataset
jobs['job_title'] = jobs['job_title'].apply(lambda x: re.sub(r'[^a-zA-Z\s]',␣
 ↪'', str(x))) # remove special characters
```

```
jobs['job_title'] = jobs['job_title'].apply(lambda x: ' '.join([word.lower()␣
↪for word in x.split() if word.lower() not in stopwords.words('english')])) #␣
↪remove stopwords and convert to lowercase

# Tokenize and preprocess points column of non-labeled dataset
jobs['points'] = jobs['points'].fillna('')
jobs['points'] = jobs['points'].apply(lambda x: ' '.join([word.lower() for word␣
↪in word_tokenize(x) if word.lower() not in stopwords.words('english')])) #␣
↪remove stopwords and convert to lowercase

# Clean and preprocess skills column of non-labeled dataset
jobs['skills'] = jobs['skills'].apply(lambda x: re.sub(r'[^a-zA-Z\s]', '',␣
↪str(x))) # remove special characters
jobs['skills'] = jobs['skills'].apply(lambda x: ' '.join([word.lower() for word␣
↪in x.split() if word.lower() not in stopwords.words('english')])) # remove␣
↪stopwords and convert to lowercase

# Clean and preprocess company_info column of non-labeled dataset
jobs['company_info'] = jobs['company_info'].apply(lambda x: re.
↪sub(r'[^a-zA-Z\s]', '', str(x))) # remove special characters
jobs['company_info'] = jobs['company_info'].apply(lambda x: ' '.join([word.
↪lower() for word in x.split() if word.lower() not in stopwords.
↪words('english')])) # remove stopwords and convert to lowercase

# Stemming and Lemmatization
from nltk.stem import WordNetLemmatizer, PorterStemmer
stemmer = PorterStemmer()
lemmatizer = WordNetLemmatizer()

jobs['points'] = jobs['points'].apply(lambda x: ' '.join([stemmer.stem(word)␣
↪for word in word_tokenize(x)])) # stemming
jobs['points'] = jobs['points'].apply(lambda x: ' '.join([lemmatizer.
↪lemmatize(word) for word in word_tokenize(x)])) # lemmatization

# Define the set of stop words to use
en_sw = set(stopwords.words('english'))
additional_stopwords = ["'d", "'ll", "'re", "'s", "'ve", 'could', 'might',␣
↪'must', "n't", 'need', 'sha', 'wo', 'would']
en_sw.update(additional_stopwords)

# Remove additional stopwords from points column of non-labeled dataset
jobs['points'] = jobs['points'].apply(lambda x: ' '.join([word for word in␣
↪word_tokenize(x) if word not in en_sw]))
```

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

```
[nltk_data] Downloading package wordnet to
[nltk_data]     /Users/poojagrewal/nltk_data…
[nltk_data]   Package wordnet is already up-to-date!
```

[15]:
```
cleaned_jobs = pd.read_csv("/Users/poojagrewal/Downloads/cleaned_jobs_dataset.
 ↪csv").fillna({'points': ''})
cleaned_ssoc = pd.read_csv("/Users/poojagrewal/Downloads/cleaned_ssoc_dataset.
 ↪csv")

print(cleaned_jobs.head())
```

```
                                                  url    company_name  \
0  https://nodesk.co/remote-jobs/activecampaign-e…  ActiveCampaign
1  https://nodesk.co/remote-jobs/betterup-marketi…        BetterUp
2  https://nodesk.co/remote-jobs/siege-media-digi…     Siege Media
3  https://nodesk.co/remote-jobs/gusto-head-of-pr…           Gusto
4  https://nodesk.co/remote-jobs/myfbaprep-partne…       MyFBAPrep

                                job_title  \
0   event marketing brand activation manager
1  marketing manager conversion optimization
2                     digital pr specialist
3              head product marketing revenue
4                   partner marketing manager

                                     company_info       country  \
0      grow business customer experience automation            US
1                                     reworking work            US
2  brands trust us deliver bestinclass content tr…            US
3      allinone people platform payroll benefits hr            US
4                make logistics competitive advantage  Remote-First

     job_type industries       salary  \
0  Internship  Marketing            0
1   Full-Time  Marketing            0
2   Full-Time  Marketing  $60K - $69K
3   Full-Time  Marketing            0
4    Contract  Marketing            0

                                            skills  \
0                        event marketing nontech
1                      marketing manager nontech
2                       nontech public relations
3  marketing manager nontech product product mark…
4  comarketing marketing manager nontech partners…

                                            points
0
```

```
1
2
3
4  plan coordinate execute marketing activities p…
```

[19]:
```
#create bag of words COUNT with customized sklearn
'''{r}
bag of words (BoW) count is created using the CountVectorizer class
from the sklearn.feature_extraction.text module. The stop_words parameter
is set to the variable en_sw, which is a set of English stopwords from
the NLTK library. The tokenizer parameter is set to the word_tokenize
function from the nltk.tokenize module, which tokenizes the input text
into individual words.

An analyzer is built from the vectorizer using the build_analyzer
method of the CountVectorizer class.
A stem_analyzer function is defined which takes a document as input,
tokenizes it, and applies stemming to each token using the PorterStemmer
from the nltk.stem module. The function returns a list of stemmed tokens.

A new CountVectorizer object is created with the analyzer parameter set
to the stem_analyzer function. This vectorizer object is then fit
to the "points" column of the "jobs" dataset using the fit_transform
method, which creates a document-term matrix of word counts.
The toarray() method is then called on this matrix to convert it into a
2D NumPy array, and the feature names (i.e., the words) are extracted
using the get_feature_names() method of the vectorizer object.

The resulting BoW count matrix is stored in the variable freq_skl,
and the feature names are stored in feature_name.
'''

from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer(stop_words=en_sw, tokenizer=word_tokenize)
analyzer = vectorizer.build_analyzer()

def stem_analyzer(doc):
    return [stemmer.stem(w) for w in analyzer(doc)]

stem_vectorizer = CountVectorizer(analyzer=stem_analyzer)
bow_skl = stem_vectorizer.fit_transform(cleaned_jobs["points"])
freq_skl = bow_skl.toarray()
feature_name = stem_vectorizer.get_feature_names_out()
```

[20]:
```
# Explore the top words by counts
import numpy as np
word_name = feature_name
```

```
word_count = np.sum(freq_skl, axis=0)
index_sort = np.argsort(word_count)
top_index = index_sort[-50:]
top_word = [word_name[ind] for ind in top_index[::-1]]
print(top_word)
```

```
['work', 'experi', 'team', 'und', 'product', 'develop', 'commun', 'year',
'remot', 'skill', 'hour', 'abil', 'flexibl', 'project', 'understand', 'du',
'mit', 'support', 'onlin', 'design', 'applic', 'manag', 'base', 'environ',
'engin', 'custom', 'busi', 'time', 'media', 'compani', 'knowledg', 'familiar',
'english', 'new', 'code', 'use', 'job', 'schedul', 'process', 'strong',
'cultur', 'content', 'interview', 'tool', 'partner', 'build', 'task',
'profession', 'well', 'zu']
```

[22]:
```
# Create bag of words TFIDF with customized sklearn
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf_vectorizer = TfidfVectorizer(analyzer=stem_analyzer)
tfidf_skl = tfidf_vectorizer.fit_transform(cleaned_jobs['points'])
tokens = tfidf_vectorizer.get_feature_names_out()
```

[23]:
```
# Explore the top words by tfidf
import numpy as np
word_name = tokens
word_count = np.sum(tfidf_skl.toarray(), axis=0)
index_sort = np.argsort(word_count)
top_index = index_sort[-50:]
top_word = [word_name[ind] for ind in top_index[::-1]]
print(top_word)
```

```
['work', 'experi', 'team', 'commun', 'onlin', 'year', 'remot', 'skill',
'develop', 'product', 'base', 'hour', 'flexibl', 'media', 'abil', 'und',
'applic', 'cultur', 'project', 'news', 'familiar', 'support', 'task', 'map',
'english', 'environ', 'design', 'schedul', 'busi', 'understand', 'k', 'use',
'custom', 'du', 'salari', 'engin', 'social', 'profession', 'knowledg',
'countri', 'full', 'practic', 'initi', 'mit', 'earn', 'fit', 'time', 'week',
'independ', 'partner']
```

[ ]:
```
#pip install spacy
#pip install --upgrade pydantic
```

[24]:
```
import re

# Define lists of keywords for responsibilities, qualifications, and benefits
responsibilities_keywords = ["manage", "develop", "coordinate", "implement",␣
 ↪"communicate", "analyze", "create", "design", "monitor", "improve"]
qualifications_keywords = ["degree", "experience", "knowledge", "skill"]
```

```
benefits_keywords = ["health", "insurance", "vacation", "paid", "time", "off",␣
↪"flexible", "remote", "work", "life", "balance", "401k"]

# Define empty lists to store the extracted values
responsibilities_list = []
qualifications_list = []
benefits_list = []

points = cleaned_jobs['points']
for point in points:
    if not isinstance(point, str):
        responsibilities_list.append([])
        continue

    # Extract job responsibilities
    matches = re.findall(r"(?:" + "|".join(responsibilities_keywords) +␣
↪r")\s+(\w+[\s\w]*)", point, re.IGNORECASE)
    responsibilities = [match.strip() for match in matches]
    responsibilities_list.append(responsibilities)

    # Extract job qualifications
    matches = re.findall(r"(?:" + "|".join(qualifications_keywords) +␣
↪r")\s+(\w+[\s\w]*)", point, re.IGNORECASE)
    qualifications = [match.strip() for match in matches]
    qualifications_list.append(qualifications)

    # Extract job benefits
    matches = re.findall(r"(?:" + "|".join(benefits_keywords) +␣
↪r")\s+(\w+[\s\w]*)", point, re.IGNORECASE)
    benefits = [match.strip() for match in matches]
    benefits_list.append(benefits)

# Add the new columns to the dataframe
cleaned_jobs["responsibilities"] = responsibilities_list
cleaned_jobs["qualifications"] = qualifications_list
cleaned_jobs["benefits"] = benefits_list

# Print the first few rows of the updated dataframe
#print(cleaned_jobs['benefits'].head(30))
#print(cleaned_jobs['responsibilities'].head(30))
#print(cleaned_jobs['qualifications'].head(30))

cleaned_jobs.head(30)
```

[24]:
```
                                                 url      company_name  \
0   https://nodesk.co/remote-jobs/activecampaign-e…    ActiveCampaign
1   https://nodesk.co/remote-jobs/betterup-marketi…          BetterUp
```

```
2   https://nodesk.co/remote-jobs/siege-media-digi…      Siege Media
3   https://nodesk.co/remote-jobs/gusto-head-of-pr…            Gusto
4   https://nodesk.co/remote-jobs/myfbaprep-partne…        MyFBAPrep
5   https://nodesk.co/remote-jobs/grafana-labs-mar…     Grafana Labs
6   https://nodesk.co/remote-jobs/help-scout-seo-s…       Help Scout
7   https://nodesk.co/remote-jobs/brex-content-des…             Brex
8   https://nodesk.co/remote-jobs/stripe-customer-…           Stripe
9   https://nodesk.co/remote-jobs/duckduckgo-senio…       DuckDuckGo
10  https://nodesk.co/remote-jobs/eyeo-brand-strat…             Eyeo
11  https://nodesk.co/remote-jobs/10up-senior-ux-d…             10up
12  https://nodesk.co/remote-jobs/brex-manager-pro…             Brex
13  https://nodesk.co/remote-jobs/kinsta-full-stac…           Kinsta
14  https://nodesk.co/remote-jobs/dropbox-internat…          Dropbox
15  https://nodesk.co/remote-jobs/general-assembly…  General Assembly
16  https://nodesk.co/remote-jobs/cloudflare-suppo…       Cloudflare
17  https://nodesk.co/remote-jobs/khan-academy-sen…     Khan Academy
18  https://nodesk.co/remote-jobs/boulevard-data-a…        Boulevard
19  https://nodesk.co/remote-jobs/shopify-data-sci…          Shopify
20  https://nodesk.co/remote-jobs/cb-insights-data…      CB Insights
21  https://nodesk.co/remote-jobs/octopus-deploy-a…   Octopus Deploy
22  https://nodesk.co/remote-jobs/nannyml-senior-d…          NannyML
23  https://nodesk.co/remote-jobs/graphcms-ecosyst…         GraphCMS
24  https://nodesk.co/remote-jobs/angellist-ventur…        AngelList
25  https://nodesk.co/remote-jobs/semaphore-releas…        Semaphore
26  https://nodesk.co/remote-jobs/okta-sr-quality-…             Okta
27  https://nodesk.co/remote-jobs/monzo-ios-engineer/            Monzo
28  https://nodesk.co/remote-jobs/grafana-labs-pla…     Grafana Labs
29  https://nodesk.co/remote-jobs/dropbox-software…          Dropbox

                                           job_title  \
0           event marketing brand activation manager
1          marketing manager conversion optimization
2                              digital pr specialist
3                    head product marketing revenue
4                          partner marketing manager
5       marketing analytics demand generation intern
6                                     seo specialist
7                                     content design
8                         customer marketing manager
9          senior designer ad creative art direction
10                               brand strategy lead
11                                senior ux designer
12                           manager product design
13                           fullstack web designer
14     international internal communications manager
15                              uxdi instructor lead
16              support project manager intern summer
```

```
17          senior data scientist analyst marketing
18                                     data analyst
19                       data science manager emea
20                         data research associate
21                       account director bilingual
22                       senior data science writer
23             ecosystem technology partner manager
24                                venture associate
25                                 release engineer
26   sr quality assurance engineer account services
27                                     ios engineer
28              platform engineering intern program
29  software development engineer intern test summer

                                       company_info       country  \
0       grow business customer experience automation            US
1                                      reworking work            US
2   brands trust us deliver bestinclass content tr…           US
3       allinone people platform payroll benefits hr            US
4               make logistics competitive advantage  Remote-First
5      composable open source observability platform            US
6         simple customer service software education            US
7             financial os next generation business            US
8      online payment processing internet businesses            US
9                     smarter search without tracking  Remote-First
10   develops open source software makers adblockplus  Remote-First
11             finely crafted websites content tools  Remote-First
12             financial os next generation business            US
13  application hosting database hosting managed w…  Remote-First
14          keep life organised work moving one place       Ireland
15  leading source training staffing career transi…           US
16                  web performance security company            US
17              learn anything free everyone forever        Canada
18                                  software selfcare            US
19                        best ecommerce platform made          EMEA
20          build software predicts technology trends            US
21       automated deployment release management tool       Germany
22  monitor decisions taken ai ensuring always add…  Remote-First
23  join graphcms building cuttingedge content man…  Remote-First
24                              world meets startups  Remote-First
25  best cicd solution highperformance engineering…          EMEA
26                     identity company stands trust            US
27  bank lives phone mission make money work everyone            US
28      composable open source observability platform          EMEA
29          keep life organised work moving one place            US

      job_type   industries        salary  \
0      Internship    Marketing            0
1       Full-Time    Marketing            0
2       Full-Time    Marketing   $60K - $69K
3       Full-Time    Marketing            0
4        Contract    Marketing            0
5      Internship    Marketing            0
6       Full-Time    Marketing  $99K - $107K
7       Full-Time       Design            0
8       Full-Time    Marketing            0
9       Full-Time       Design            0
10      Full-Time    Marketing            0
11      Full-Time       Design            0
12      Full-Time       Design            0
13      Full-Time       Design            0
14  Not specified    Marketing            0
15      Full-Time        Other            0
16     Internship   Operations            0
17             US        Other            0
18      Full-Time        Other            0
19      Full-Time        Other            0
20      Full-Time        Other            0
21  Not specified        Sales            0
22      Full-Time        Other            0
23      Full-Time        Other            0
24      Full-Time        Other            0
25  Latin America  Engineering            0
26      Full-Time  Engineering            0
27      Full-Time  Engineering            0
28     Internship  Engineering            0
29     Internship  Engineering            0

                                                skills  \
0                             event marketing nontech
1                           marketing manager nontech
2                            nontech public relations
3   marketing manager nontech product product mark…
4   comarketing marketing manager nontech partners…
5                                  entrylevel nontech
6                                         nontech seo
7                                    content designer
8                           marketing manager nontech
9                                            designer
10                            brand marketing nontech
11                                        ux designer
12                            product product designer
13                                       web designer
14                              communications nontech
```

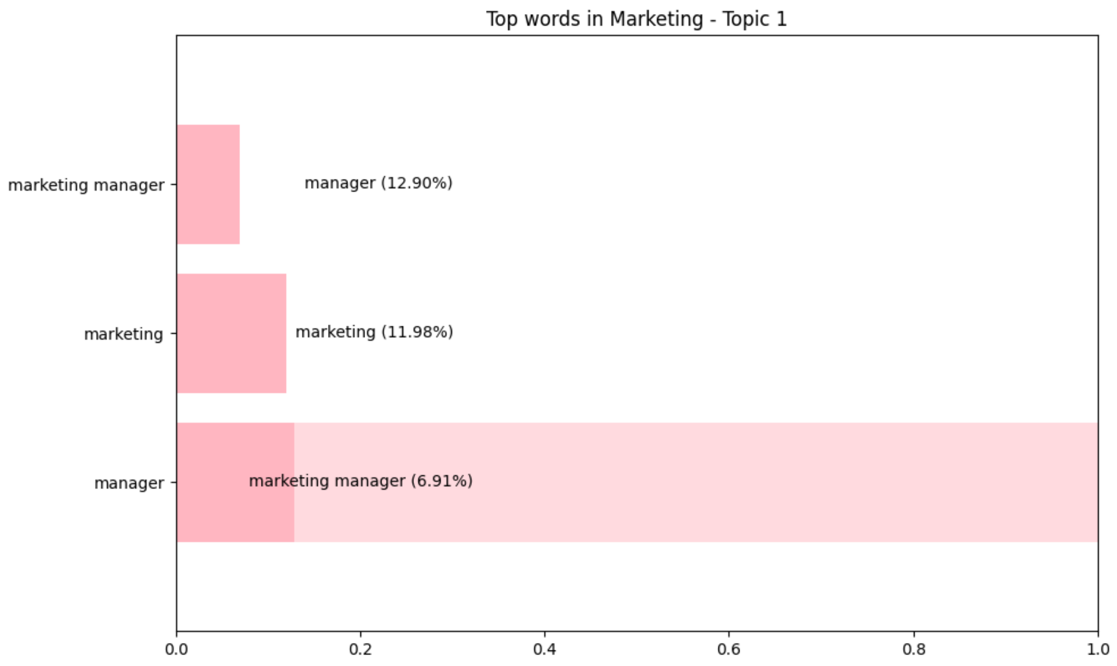
```
15                              instructor ux designer
16                          entrylevel project manager
17                       data data scientist nonprofit
18                                                data
19                                      data scientist
20                               data finance research
21                             account manager nontech
22  content writer data data scientist technical w…
23  account manager business development partner m…
24                        finance nontech venture capital
25   aws cloud engineer google cloud kubernetes linux
26                                         engineer qa
27                  engineer ios mobile developer swift
28                 backend engineer engineer entrylevel
29                   engineer entrylevel test engineer

                                              points  \
0
1
2
3
4   plan coordinate execute marketing activities p…
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22  work closely founders research product teams s…
23  customercentric goaldriven believe innovation …
24
25  charge testing release process collaborate clo…
26
27
28
29
```

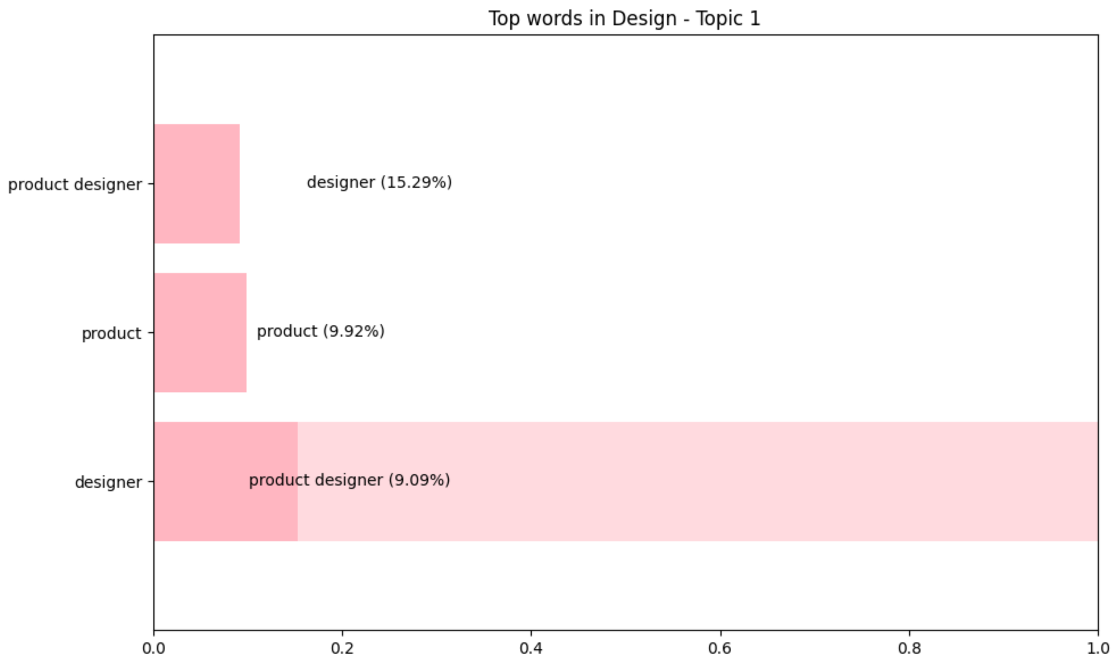
```
                                     responsibilities  \
0                                                  []
1                                                  []
2                                                  []
3                                                  []
4   [execute marketing activities partners manage …
5                                                  []
6                                                  []
7                                                  []
8                                                  []
9                                                  []
10                                                 []
11                                                 []
12                                                 []
13                                                 []
14                                                 []
15                                                 []
16                                                 []
17                                                 []
18                                                 []
19                                                 []
20                                                 []
21                                                 []
22  [interactive visualizations code ensure nannym…
23  [recommend products solutions establishing mut…
24                                                 []
25  [support hosting solutions maintain release do…
26                                                 []
27                                                 []
28                                                 []
29                                                 []

                                       qualifications  \
0                                                  []
1                                                  []
2                                                  []
3                                                  []
4                                                  []
5                                                  []
6                                                  []
7                                                  []
8                                                  []
9                                                  []
10                                                 []
11                                                 []
12                                                 []
```

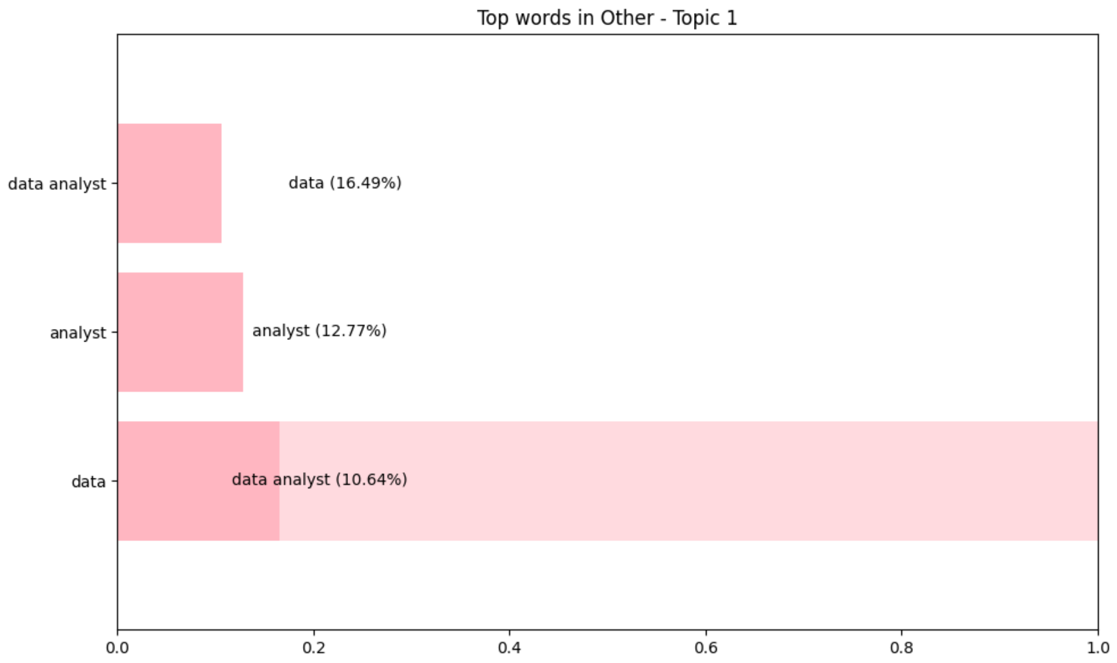
```
13                                                []
14                                                []
15                                                []
16                                                []
17                                                []
18                                                []
19                                                []
20                                                []
21                                                []
22  [developing productionready ml algorithms pyth…
23  [building relationships partner customer teams…
24                                                []
25  [experience shipping onpremises enterprise sof…
26                                                []
27                                                []
28                                                []
29                                                []

                                            benefits
0                                                 []
1                                                 []
2                                                 []
3                                                 []
4                                        [timezones]
5                                                 []
6                                                 []
7                                                 []
8                                                 []
9                                                 []
10                                                []
11                                                []
12                                                []
13                                                []
14                                                []
15                                                []
16                                                []
17                                                []
18                                                []
19                                                []
20                                                []
21                                                []
22  [closely founders research product teams shape…
23  [together responsibility accountability owners…
24                                                []
25  [people love healthy hour work week friendly s…
26                                                []
27                                                []
```

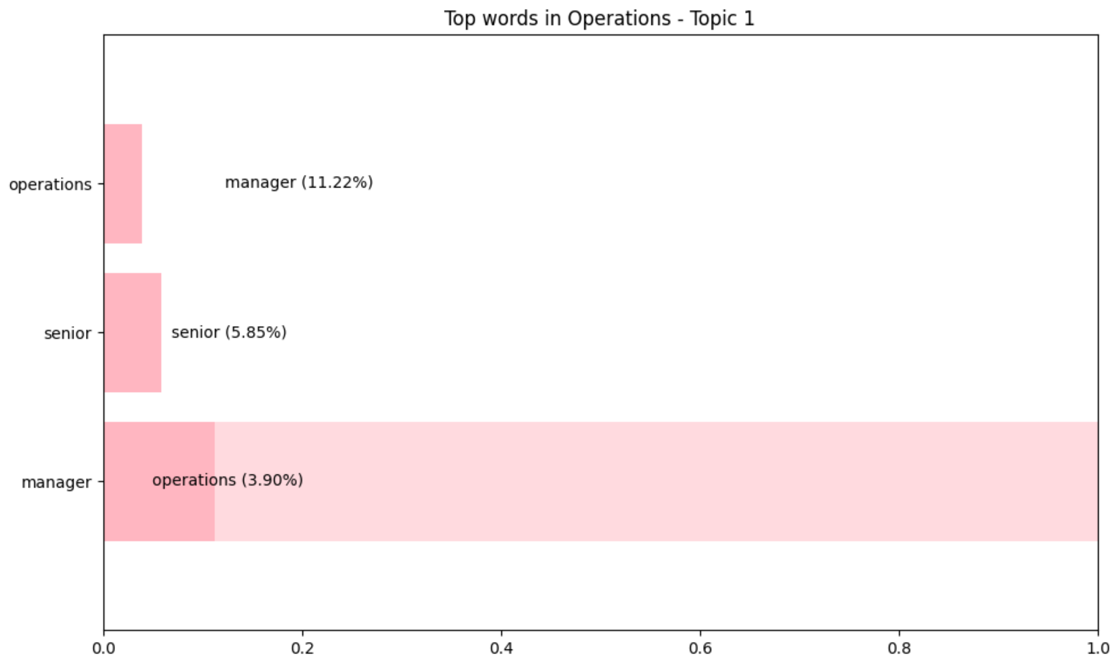
```
28                                                   []
29                                                   []
```

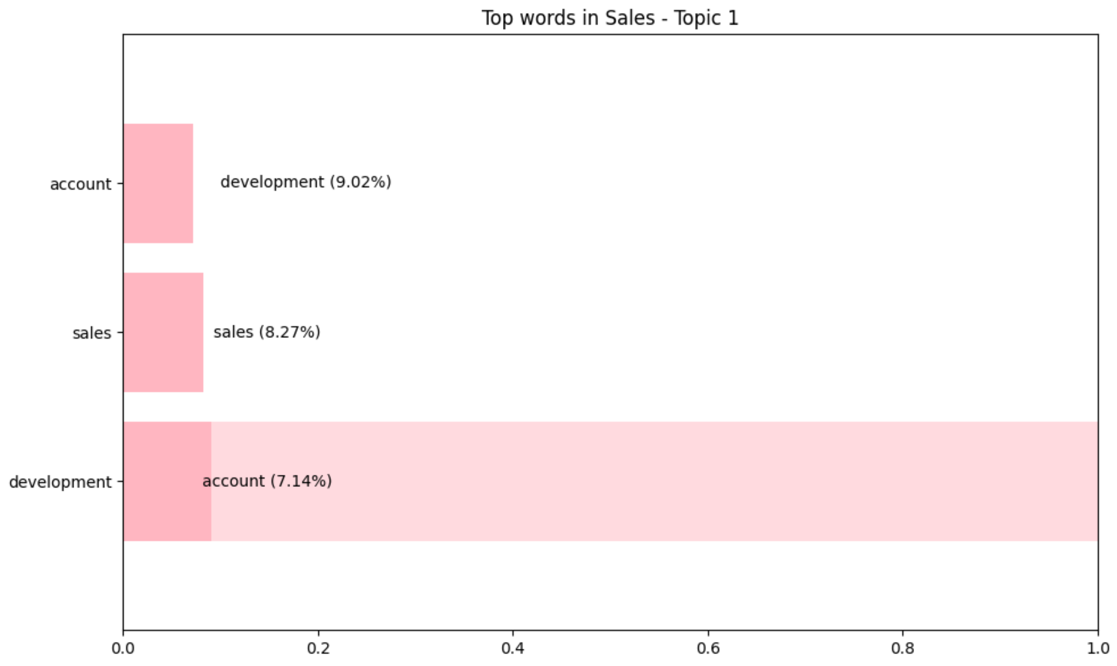
[29]:
```
'''{r}
- plot the top words by industry and topic
'''
import matplotlib.pyplot as plt
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.decomposition import LatentDirichletAllocation

# Define CountVectorizer with desired parameters
cv = CountVectorizer(stop_words='english', max_df=0.95, min_df=2,␣
 ↪ngram_range=(1,2))

# Set the color for the plot
color = "#FFB6C1"

# Loop through each industry and find the top topics in their job titles
for industry in cleaned_jobs['industries'].unique():
    # Filter the jobs dataframe for the current industry
    industry_jobs = cleaned_jobs[cleaned_jobs['industries'] == industry]
    # Fit the vectorizer to the 'job_title' column of the industry jobs
    cv.fit(industry_jobs['job_title'])

    # Transform the 'job_title' column into a document-term matrix
    dtm = cv.transform(industry_jobs['job_title'])

    # Create an LDA object with the desired number of topics
    num_topics = 1
    lda_model = LatentDirichletAllocation(n_components=num_topics,␣
 ↪random_state=42)

    # Fit the model to the document-term matrix
    lda_model.fit(dtm)

    # Define a function to display the top words from each topic
    def display_topics(model, feature_names, num_top_words):
        top_words_list = []
        for topic_idx, topic in enumerate(model.components_):
            top_keywords = [(feature_names[i], topic[i]/np.sum(topic)) for i in␣
 ↪topic.argsort()[:-num_top_words - 1:-1]]
            top_words_list.append(top_keywords)
        return top_words_list

    # Get the top topics and their associated words for the industry
    top_words = display_topics(lda_model, cv.get_feature_names_out(), 3)
    # Create a bar chart for each topic
    plt.figure(figsize=(10,6))
    for topic_idx, topic_words in enumerate(top_words):
        plt.barh([f"Topic {topic_idx+1}"], [1], color=color, alpha=0.5)
        y_pos = np.arange(len(topic_words))[::-1]
        x_pos = [word_prob[1] for word_prob in topic_words][::-1]
        plt.barh(y_pos, x_pos, color=color)
        for i, (word, prob) in enumerate(topic_words):
            plt.text(prob+0.01, y_pos[i], f"{word} ({prob:.2%})", va="center")
        plt.xlim([0, 1])
        plt.ylim([-1, len(topic_words)])
        plt.yticks(range(len(topic_words)), [word for word, prob in␣
↪topic_words])
        plt.title(f"Top words in {industry} - Topic {topic_idx+1}")
        plt.tight_layout()
        plt.show()
```

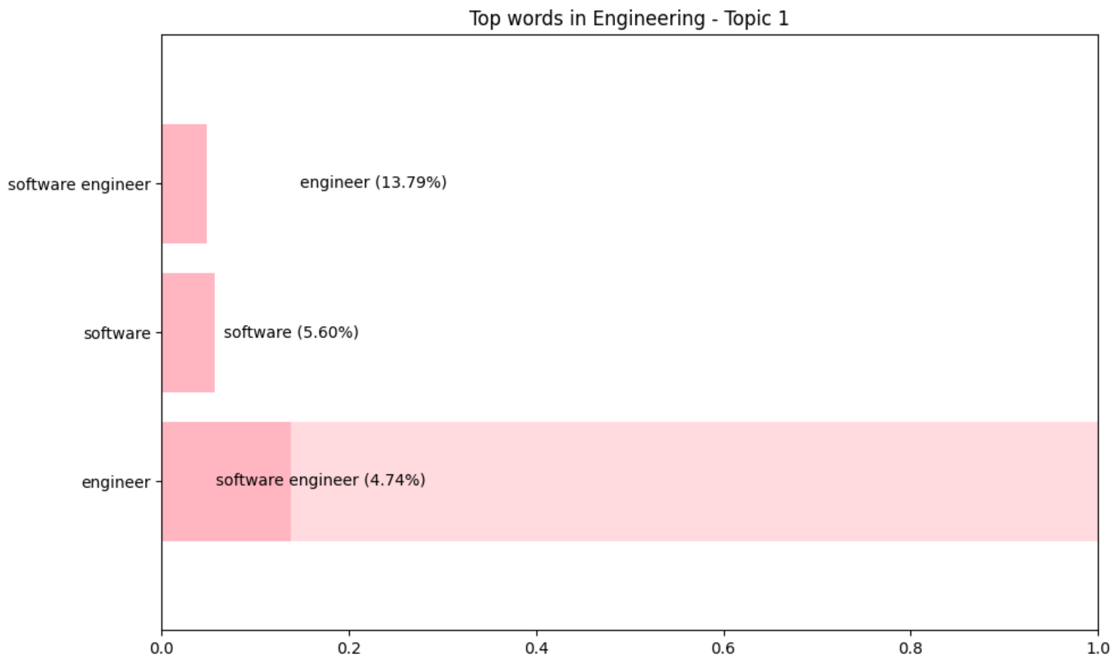


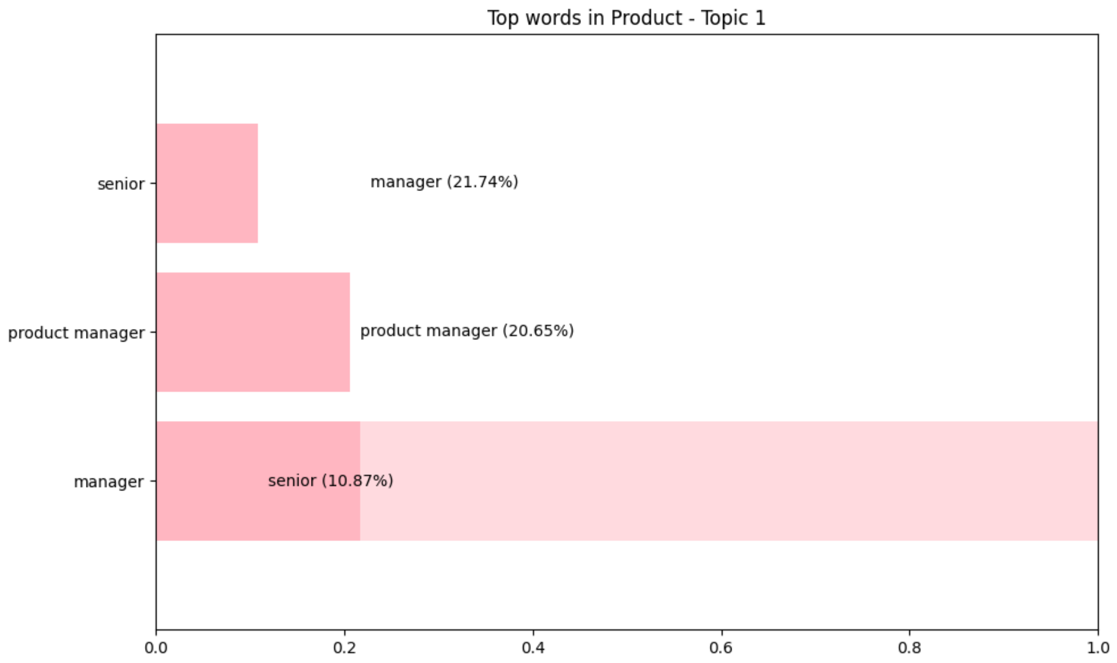

Top words in Design - Topic 1
product designer
product
designer
designer (15.29%)
product (9.92%)
product designer (9.09%)
0.0
0.2
0.4
0.6
0.8
1.0

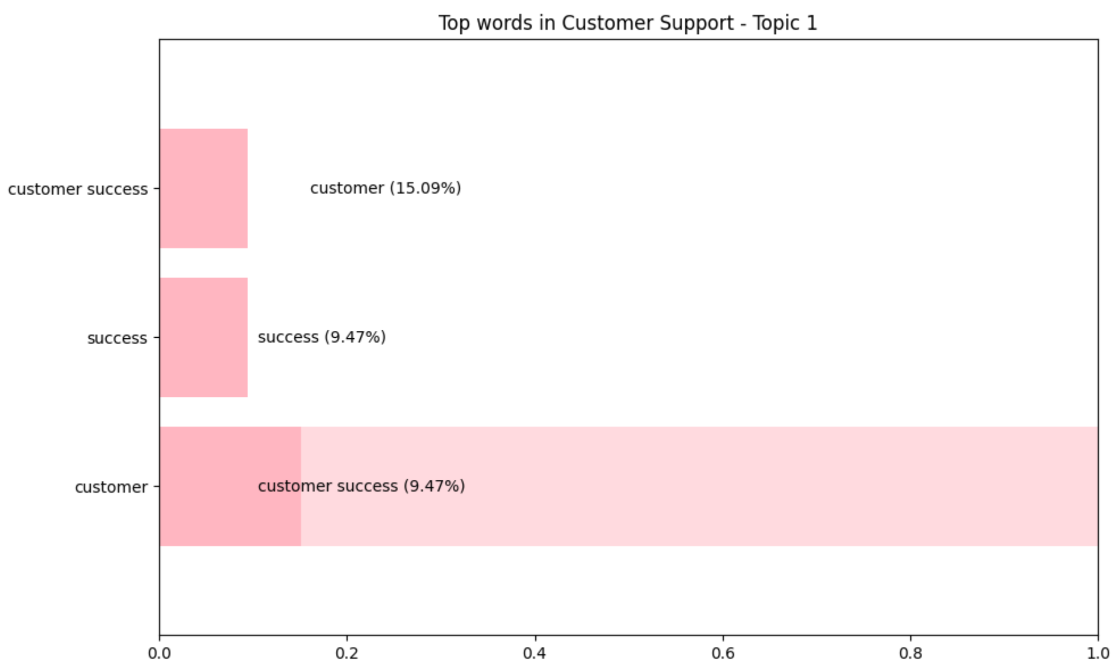

Top words in Other - Topic 1
data analyst
analyst
data
data (16.49%)
analyst (12.77%)
data analyst (10.64%)
0.0
0.2
0.4
0.6
0.8
1.0


Top words in Operations - Topic 1
operations
senior
manager
manager (11.22%)
senior (5.85%)
operations (3.90%)
0.0
0.2
0.4
0.6
0.8
1.0


Top words in Sales - Topic 1
account
sales
development
development (9.02%)
sales (8.27%)
account (7.14%)
0.0
0.2
0.4
0.6
0.8
1.0


Top words in Engineering - Topic 1
software engineer
software
engineer
engineer (13.79%)
software (5.60%)
software engineer (4.74%)
0.0
0.2
0.4
0.6
0.8
1.0


Top words in Product - Topic 1
senior
product manager
manager
manager (21.74%)
product manager (20.65%)
senior (10.87%)
0.0
0.2
0.4
0.6
0.8
1.0



```
[32]: from datasets import load_dataset
      from sklearn.model_selection import train_test_split
      import pandas as pd
      import numpy as np
      from sklearn.model_selection import train_test_split


      # Split the labeled data into train and test sets
      train_data, test_data = train_test_split(cleaned_ssoc, test_size=0.2,␣
       ↪random_state=42)
      print(train_data)
      print(test_data)

      # Save the training set and testing set to separate CSV files
      train_data.to_csv('train_dataset.csv', index=False)
      test_data.to_csv('test_dataset.csv', index=False)
```

```
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/tqdm/auto.py:21: TqdmWarning: IProgress not found. Please update
jupyter and ipywidgets. See
https://ipywidgets.readthedocs.io/en/stable/user_install.html
  from .autonotebook import tqdm as notebook_tqdm

     SSOC 2020                                          job_title
664      13420                           health services manager
4367     33313  freight inspectorincoming quality inspector fr…
```

```
4161    32400                                veterinary assistant
208     12193  operations directorgeneral manager sfwenvironm…
5318    51112                                 air hostess aircrew
…       …                                                       …
5734    54130                                       prison warden
5191    43141                                        coding clerk
5390    51311                                        fb executive
860     14121                       food services store manager
7270    81420                                  plastics laminator

[6705 rows x 2 columns]
     SSOC 2020                                   job_title
2865    26120                                 adjudicator
8173    93332                                truck loader
5933    71120                      bricklayer construction
4857    36100                       nursery school teacher
6857    75390                             fabrics repairer
…       …                                               …
1129    21311  biomedical nanotechnology research scientist
7160    81251                          sheet metal spinner
6395    73130                                 jewel setter
5720    54121                             police constable
3139    26521                                      flutist

[1677 rows x 2 columns]
```

[35]:
```
#trying for fun
from gensim.models import Word2Vec, KeyedVectors

# Tokenize the job_title column
data = [row.split(' ') for row in cleaned_ssoc['job_title']]

# Train a Word2Vec model on the tokenized job_title column
model = Word2Vec(data, min_count=3, vector_size=100, workers=3, window=5, sg=1,␣
 ↪epochs=100)

# Save the trained word vectors to a file
model.wv.save_word2vec_format('vectors.kv', binary=True)

# Load the trained word vectors from the file
word_vectors = KeyedVectors.load_word2vec_format('vectors.kv', binary=True)

# Get the list of words in the vocabulary
words = list(word_vectors.index_to_key)

# Print the list of words
#print(words)
```

```
# Explore relationships between words
word1 = 'analyst'
word2 = 'machine'
print(f"Similarity between '{word1}' and '{word2}': {word_vectors.
  ↪similarity(word1, word2)}")

# Map new words to the vector space
new_word = 'programming'
print(f"Vector representation of '{new_word}': {word_vectors[new_word]}")
```

```
Similarity between 'analyst' and 'machine': 0.17255747318267822
Vector representation of 'programming': [-0.61112374 -0.05657014  0.22469619
0.28364    -0.44403413 -0.289874
 -0.34934992  0.5734066  -0.06208234 -0.43059972  0.06194444 -0.5441969
 -0.17961998 -0.20258233  0.0128432   0.03245772  0.02500661 -0.54600245
  0.3759271  -0.36747542  0.73641825 -0.318087    0.22887765  0.03946764
 -0.4210757   0.68849945 -0.63713306  0.02472589  0.18615648  0.26614985
 -0.12027439  0.19654961  0.45121232 -0.813081   -0.04206379  0.61168796
  0.10973629 -0.05091398 -0.07452362 -0.22194774  0.14084585 -0.32045656
  0.3522506   0.160962   -0.00569785 -0.09806851 -0.27507904  0.02199258
  0.19263768  0.1491626  -0.5785498   0.15970841  0.69931954  0.5721956
  0.26628387 -0.1486726  -0.166907    0.47890863 -0.21747401  0.22871628
  0.00463984  0.26408875  0.5623532   0.39671814 -0.5362544   0.13415706
  0.18390684  0.3632501   0.17562474  0.39739686 -0.3902734  -0.0096298
 -0.07587623 -0.00961866  0.15085302  0.09576426  0.15380363  0.23961432
  0.03866796 -0.29133603 -0.33881772 -0.04829717 -0.02509066  0.31236476
 -0.3090384  -0.04044955 -0.12887569 -0.356607    0.52258706 -0.18679208
  0.28844693 -0.1504067  -0.138986    0.27622804  0.52003896 -0.12552471
 -0.451733   -0.49155277 -0.52762836 -0.29303637]
```

[36]:
```
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import LinearSVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score,␣
  ↪f1_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import StratifiedKFold


# Split the labeled data into train and test sets
train_data, test_data = train_test_split(cleaned_ssoc, test_size=0.2,␣
  ↪random_state=42)

# Define the models to use
```

```
models = [
('Naive Bayes with CountVectorizer', MultinomialNB(), CountVectorizer(),␣
 ↪{'alpha': [0.1, 0.5, 1.0]}),
('Naive Bayes with TfidfVectorizer', MultinomialNB(), TfidfVectorizer(),␣
 ↪{'alpha': [0.1, 0.5, 1.0]}),
('Decision Tree with CountVectorizer', DecisionTreeClassifier(),␣
 ↪CountVectorizer(), {'max_depth': [None, 10, 50, 100]}),
('Decision Tree with TfidfVectorizer', DecisionTreeClassifier(),␣
 ↪TfidfVectorizer(), {'max_depth': [None, 10, 50, 100]}),
('Random Forest with CountVectorizer', RandomForestClassifier(),␣
 ↪CountVectorizer(), {'n_estimators': [50, 100, 200], 'max_depth': [None, 10,␣
 ↪50, 100]}),
('Random Forest with TfidfVectorizer', RandomForestClassifier(),␣
 ↪TfidfVectorizer(), {'n_estimators': [50, 100, 200], 'max_depth': [None, 10,␣
 ↪50, 100]}),
('SVM with CountVectorizer', LinearSVC(), CountVectorizer(), {'C': [0.1, 1,␣
 ↪10]}),
('SVM with TfidfVectorizer', LinearSVC(), TfidfVectorizer(), {'C': [0.1, 1,␣
 ↪10]})
]

# Define the cross-validation strategy
cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)

# Loop through each model, fit the data and evaluate the performance using␣
 ↪cross-validation
for name, model, vectorizer, params in models:
    # Fit the vectorizer to the training data
    train_vectors = vectorizer.fit_transform(train_data['job_title'])

    # Perform grid search cross-validation to find the best hyperparameters
    clf = GridSearchCV(model, params, cv=cv)
    clf.fit(train_vectors, train_data['SSOC 2020'])

    # Make predictions on the test data using the best model found
    test_vectors = vectorizer.transform(test_data['job_title'])
    y_pred = clf.predict(test_vectors)

    # Evaluate the model performance
    accuracy = accuracy_score(test_data['SSOC 2020'], y_pred)
    precision = precision_score(test_data['SSOC 2020'], y_pred,␣
 ↪average='weighted', zero_division=1)
    recall = recall_score(test_data['SSOC 2020'], y_pred, average='weighted',␣
 ↪zero_division=1)
    f1 = f1_score(test_data['SSOC 2020'], y_pred, average='weighted')
```

```python
    # Print the performance metrics for the current model
    print(f"Model: {name}")
    print(f"Best Parameters: {clf.best_params_}")
    print(f"Accuracy: {accuracy}")
    print(f"Precision: {precision}")
    print(f"Recall: {recall}")
    print(f"F1 Score: {f1}")
    print("\n")
```

```
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/model_selection/_split.py:737: UserWarning: The least populated
class in y has only 1 members, which is less than n_splits=5.
  warnings.warn(
```

```
Model: Naive Bayes with CountVectorizer
Best Parameters: {'alpha': 0.1}
Accuracy: 0.43410852713178294
Precision: 0.7458299924363061
Recall: 0.43410852713178294
F1 Score: 0.39082156629150555
```

```
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/model_selection/_split.py:737: UserWarning: The least populated
class in y has only 1 members, which is less than n_splits=5.
  warnings.warn(
```

```
Model: Naive Bayes with TfidfVectorizer
Best Parameters: {'alpha': 0.1}
Accuracy: 0.40429338103756707
Precision: 0.7502641279334091
Recall: 0.40429338103756707
F1 Score: 0.3533154614893246
```

```
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/model_selection/_split.py:737: UserWarning: The least populated
class in y has only 1 members, which is less than n_splits=5.
  warnings.warn(
```

```
Model: Decision Tree with CountVectorizer
Best Parameters: {'max_depth': None}
Accuracy: 0.4108527131782946
Precision: 0.7238777214882429
Recall: 0.4108527131782946
F1 Score: 0.40946753481195153
```

```
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/model_selection/_split.py:737: UserWarning: The least populated
class in y has only 1 members, which is less than n_splits=5.
  warnings.warn(
```

```
Model: Decision Tree with TfidfVectorizer
Best Parameters: {'max_depth': None}
Accuracy: 0.358974358974359
Precision: 0.6566175579164613
Recall: 0.358974358974359
F1 Score: 0.36547730386508254
```

```
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/model_selection/_split.py:737: UserWarning: The least populated
class in y has only 1 members, which is less than n_splits=5.
  warnings.warn(
```

```
Model: Random Forest with CountVectorizer
Best Parameters: {'max_depth': None, 'n_estimators': 200}
Accuracy: 0.5002981514609421
Precision: 0.7438874422013969
Recall: 0.5002981514609421
F1 Score: 0.49364141871061884
```

```
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/model_selection/_split.py:737: UserWarning: The least populated
class in y has only 1 members, which is less than n_splits=5.
  warnings.warn(
```

```
Model: Random Forest with TfidfVectorizer
Best Parameters: {'max_depth': None, 'n_estimators': 100}
Accuracy: 0.456768038163387
Precision: 0.6948528592137229
Recall: 0.456768038163387
F1 Score: 0.4496128619510709
```

```
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/model_selection/_split.py:737: UserWarning: The least populated
class in y has only 1 members, which is less than n_splits=5.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
```

```
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_base.py:1250: ConvergenceWarning: Liblinear failed to
converge, increase the number of iterations.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_base.py:1250: ConvergenceWarning: Liblinear failed to
converge, increase the number of iterations.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_base.py:1250: ConvergenceWarning: Liblinear failed to
converge, increase the number of iterations.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_base.py:1250: ConvergenceWarning: Liblinear failed to
converge, increase the number of iterations.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_base.py:1250: ConvergenceWarning: Liblinear failed to
converge, increase the number of iterations.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
Model: SVM with CountVectorizer
Best Parameters: {'C': 1}
Accuracy: 0.5497912939773405
Precision: 0.7539879728699048
Recall: 0.5497912939773405
F1 Score: 0.5407857802791008
```

```
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/model_selection/_split.py:737: UserWarning: The least populated
class in y has only 1 members, which is less than n_splits=5.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
```

```
Model: SVM with TfidfVectorizer
Best Parameters: {'C': 10}
```

```
Accuracy: 0.5414430530709601
Precision: 0.7408800177494274
Recall: 0.5414430530709601
F1 Score: 0.53372106338198
```

[38]:

```python
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import LinearSVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score,␣
 ↪f1_score
from sklearn.ensemble import RandomForestClassifier

# Split the labeled data into train and test sets
train_data, test_data = train_test_split(cleaned_ssoc, test_size=0.2,␣
 ↪random_state=42)

# Define the models to use
models = [
('Naive Bayes with CountVectorizer', MultinomialNB(), CountVectorizer()),
('Naive Bayes with TfidfVectorizer', MultinomialNB(), TfidfVectorizer()),
('Decision Tree with CountVectorizer', DecisionTreeClassifier(),␣
 ↪CountVectorizer()),
('Decision Tree with TfidfVectorizer', DecisionTreeClassifier(),␣
 ↪TfidfVectorizer()),
('Random Forest with CountVectorizer', RandomForestClassifier(),␣
 ↪CountVectorizer()),
('Random Forest with TfidfVectorizer', RandomForestClassifier(),␣
 ↪TfidfVectorizer()),
('SVM with CountVectorizer', LinearSVC(), CountVectorizer()),
('SVM with TfidfVectorizer', LinearSVC(), TfidfVectorizer())
]

# Loop through each model, fit the data and evaluate the performance
for name, model, vectorizer in models:
    # Fit the vectorizer to the training data
    train_vectors = vectorizer.fit_transform(train_data['job_title'])

    # Transform the test data using the same vectorizer
    test_vectors = vectorizer.transform(test_data['job_title'])

    # Fit the model to the training data
    model.fit(train_vectors, train_data['SSOC 2020'])

    # Make predictions on the test data
    y_pred = model.predict(test_vectors)

    # Evaluate the model performance
    accuracy = accuracy_score(test_data['SSOC 2020'], y_pred)
    precision = precision_score(test_data['SSOC 2020'], y_pred,␣
↪average='weighted', zero_division=1)
    recall = recall_score(test_data['SSOC 2020'], y_pred, average='weighted',␣
↪zero_division=1)
    f1 = f1_score(test_data['SSOC 2020'], y_pred, average='weighted')


    # Print the performance metrics for the current model
    print(f"Model: {name}")
    print(f"Accuracy: {accuracy}")
    print(f"Precision: {precision}")
    print(f"Recall: {recall}")
    print(f"F1 Score: {f1}")
    print("\n")
```

```
Model: Naive Bayes with CountVectorizer
Accuracy: 0.28920691711389385
Precision: 0.7843289167817011
Recall: 0.28920691711389385
F1 Score: 0.24138009699461668


Model: Naive Bayes with TfidfVectorizer
Accuracy: 0.18604651162790697
Precision: 0.8663963113370651
Recall: 0.18604651162790697
F1 Score: 0.15322777820328284


Model: Decision Tree with CountVectorizer
Accuracy: 0.41741204531902204
Precision: 0.729854590179532
Recall: 0.41741204531902204
F1 Score: 0.41909465757128905


Model: Decision Tree with TfidfVectorizer
Accuracy: 0.3595706618962433
Precision: 0.674915795630009
Recall: 0.3595706618962433
F1 Score: 0.3674359338626936
```

```
Model: Random Forest with CountVectorizer
Accuracy: 0.48300536672629696
Precision: 0.7317805657735863
Recall: 0.48300536672629696
F1 Score: 0.4775837294056962


Model: Random Forest with TfidfVectorizer
Accuracy: 0.4537865235539654
Precision: 0.6898007657344256
Recall: 0.4537865235539654
F1 Score: 0.4514382096781464
```

```
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
```

```
Model: SVM with CountVectorizer
Accuracy: 0.5497912939773405
Precision: 0.7539879728699048
Recall: 0.5497912939773405
F1 Score: 0.5407857802791008
```

```
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
```

```
Model: SVM with TfidfVectorizer
Accuracy: 0.5235539654144306
Precision: 0.7296693620046573
Recall: 0.5235539654144306
F1 Score: 0.50894700187864
```

[39]:
```
#tried hyperparameter tuning for svm model but only slight diff to SVM with␣
 ↪TfidfVectorizer
from sklearn.pipeline import Pipeline
from sklearn.model_selection import GridSearchCV

# Split the labeled data into train and test sets
```

```python
train_data, test_data = train_test_split(cleaned_ssoc, test_size=0.2,␣
 ↪random_state=42)

# Define the models to use
models = [
    ('Naive Bayes with CountVectorizer', MultinomialNB(), CountVectorizer(),␣
 ↪{}),
    ('Naive Bayes with TfidfVectorizer', MultinomialNB(), TfidfVectorizer(),␣
 ↪{}),
    ('SVM with CountVectorizer', LinearSVC(), CountVectorizer(), {'model__C':␣
 ↪[0.1, 1, 10]}),
    ('SVM with TfidfVectorizer', LinearSVC(), TfidfVectorizer(), {'model__C':␣
 ↪[0.1, 1, 10]})
]

# Loop through each model, fit the data and evaluate the performance
for name, model, vectorizer, param_grid in models:
    # Create a pipeline with the vectorizer and model
    pipeline = Pipeline([
        ('vectorizer', vectorizer),
        ('model', model)
    ])

    # Perform grid search with cross-validation to find the best hyperparameters
    grid_search = GridSearchCV(pipeline, param_grid=param_grid, cv=5)
    grid_search.fit(train_data['job_title'], train_data['SSOC 2020'])

    # Make predictions on the test data
    y_pred = grid_search.predict(test_data['job_title'])

    # Evaluate the model performance
    accuracy = accuracy_score(test_data['SSOC 2020'], y_pred)
    precision = precision_score(test_data['SSOC 2020'], y_pred,␣
 ↪average='weighted', zero_division=1)
    recall = recall_score(test_data['SSOC 2020'], y_pred, average='weighted',␣
 ↪zero_division=1)
    f1 = f1_score(test_data['SSOC 2020'], y_pred, average='weighted')

    # Print the performance metrics for the current model
    print(f"Model: {name}")
    print(f"Best hyperparameters: {grid_search.best_params_}")
    print(f"Accuracy: {accuracy}")
    print(f"Precision: {precision}")
    print(f"Recall: {recall}")
    print(f"F1 Score: {f1}")
    print("\n")
```

```
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/model_selection/_split.py:737: UserWarning: The least populated
class in y has only 1 members, which is less than n_splits=5.
  warnings.warn(
```

```
Model: Naive Bayes with CountVectorizer
Best hyperparameters: {}
Accuracy: 0.28920691711389385
Precision: 0.7843289167817011
Recall: 0.28920691711389385
F1 Score: 0.24138009699461668
```

```
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/model_selection/_split.py:737: UserWarning: The least populated
class in y has only 1 members, which is less than n_splits=5.
  warnings.warn(
```

```
Model: Naive Bayes with TfidfVectorizer
Best hyperparameters: {}
Accuracy: 0.18604651162790697
Precision: 0.8663963113370651
Recall: 0.18604651162790697
F1 Score: 0.15322777820328284
```

```
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/model_selection/_split.py:737: UserWarning: The least populated
class in y has only 1 members, which is less than n_splits=5.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
```

```
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_base.py:1250: ConvergenceWarning: Liblinear failed to
converge, increase the number of iterations.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_base.py:1250: ConvergenceWarning: Liblinear failed to
```

```
converge, increase the number of iterations.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_base.py:1250: ConvergenceWarning: Liblinear failed to
converge, increase the number of iterations.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_base.py:1250: ConvergenceWarning: Liblinear failed to
converge, increase the number of iterations.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_base.py:1250: ConvergenceWarning: Liblinear failed to
converge, increase the number of iterations.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
```

```
Model: SVM with CountVectorizer
Best hyperparameters: {'model__C': 1}
Accuracy: 0.5497912939773405
Precision: 0.7539879728699048
Recall: 0.5497912939773405
F1 Score: 0.5407857802791008
```

```
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/model_selection/_split.py:737: UserWarning: The least populated
class in y has only 1 members, which is less than n_splits=5.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
```

```
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
```

```
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/svm/_classes.py:32: FutureWarning: The default value of `dual`
will change from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly
to suppress the warning.
  warnings.warn(

Model: SVM with TfidfVectorizer
Best hyperparameters: {'model__C': 10}
Accuracy: 0.5414430530709601
Precision: 0.7408800177494274
Recall: 0.5414430530709601
F1 Score: 0.53372106338198
```

```
[ ]:
```