```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix,accuracy_score
from sklearn import svm
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier

#membaca dan menampilkan file
df = pd.read_csv('mldata.csv')
df.head()
```

```
   Logical quotient rating  hackathons  coding skills rating  \
0                        5           0                     6   
1                        7           6                     4   
2                        2           3                     9   
3                        2           6                     3   
4                        2           0                     3   

   public speaking points self-learning capability? Extra-courses did
\
0                       2                       yes                no

1                       3                        no               yes

2                       1                        no               yes

3                       5                        no               yes

4                       4                       yes                no


         certifications           workshops reading and writing skills
\
0  information security             testing                     poor

1     shell programming             testing                excellent

2  information security             testing                excellent

3         r programming   database security                excellent

4         distro making    game development                excellent


  memory capability score   Interested subjects   interested career
area   \
0                    poor           programming
```

```
testing
1                  medium            Management          system
developer
2                    poor      data engineering  Business process
analyst
3                    poor              networks
testing
4                  medium  Software Engineering          system
developer

  Type of company want to settle in? Taken inputs from seniors or
elders  \
0                                BPA
no
1                     Cloud Services
yes
2                product development
yes
3  Testing and Maintainance Services
yes
4                                BPA
no

  Interested Type of Books Management or Technical hard/smart
worker  \
0                   Series              Management      smart worker

1          Autobiographies               Technical       hard worker

2                   Travel               Technical      smart worker

3                    Guide              Management      smart worker

4                   Health               Technical       hard worker


  worked in teams ever? Introvert      Suggested Job Role
0                   yes        no  Applications Developer
1                    no       yes  Applications Developer
2                    no        no  Applications Developer
3                   yes       yes  Applications Developer
4                   yes        no  Applications Developer
```

```
#mencetak jumlah sampel (data poin) dan jumlah fitur yang terdapat
dalam dataset
print('The shape of our training set: %s professionals and %s
features'%(df.shape[0],df.shape[1]))
```

```
The shape of our training set: 6901 professionals and 20 features
```

```
#menampilkan kolom di dataset
print("Columns in our dataset: " , df.columns)
```

```
Columns in our dataset:  Index(['Logical quotient rating',
'hackathons', 'coding skills rating',
       'public speaking points', 'self-learning capability?',
       'Extra-courses did', 'certifications', 'workshops',
       'reading and writing skills', 'memory capability score',
       'Interested subjects', 'interested career area ',
       'Type of company want to settle in?',
       'Taken inputs from seniors or elders', 'Interested Type of
Books',
       'Management or Technical', 'hard/smart worker', 'worked in
teams ever?',
       'Introvert', 'Suggested Job Role'],
      dtype='object')
```

```
#menampilkan kolom bertipe numerikal dan kategorikal
print("List of Numerical features: \n" ,
df.select_dtypes(include=np.number).columns.tolist())
print("\n\nList of Categorical features: \n" ,
df.select_dtypes(include=['object']).columns.tolist())
```

```
List of Numerical features:
 ['Logical quotient rating', 'hackathons', 'coding skills rating',
'public speaking points']


List of Categorical features:
 ['self-learning capability?', 'Extra-courses did', 'certifications',
'workshops', 'reading and writing skills', 'memory capability score',
'Interested subjects', 'interested career area ', 'Type of company
want to settle in?', 'Taken inputs from seniors or elders',
'Interested Type of Books', 'Management or Technical', 'hard/smart
worker', 'worked in teams ever?', 'Introvert', 'Suggested Job Role']
```

```
#cek missing values
df.isnull().sum(axis=0)
```

```
Logical quotient rating                0
hackathons                             0
coding skills rating                   0
public speaking points                 0
self-learning capability?              0
Extra-courses did                      0
certifications                         0
workshops                              0
reading and writing skills             0
memory capability score                0
Interested subjects                    0
interested career area                 0
```

```
Type of company want to settle in?      0
Taken inputs from seniors or elders     0
Interested Type of Books                0
Management or Technical                 0
hard/smart worker                       0
worked in teams ever?                   0
Introvert                               0
Suggested Job Role                      0
dtype: int64
```

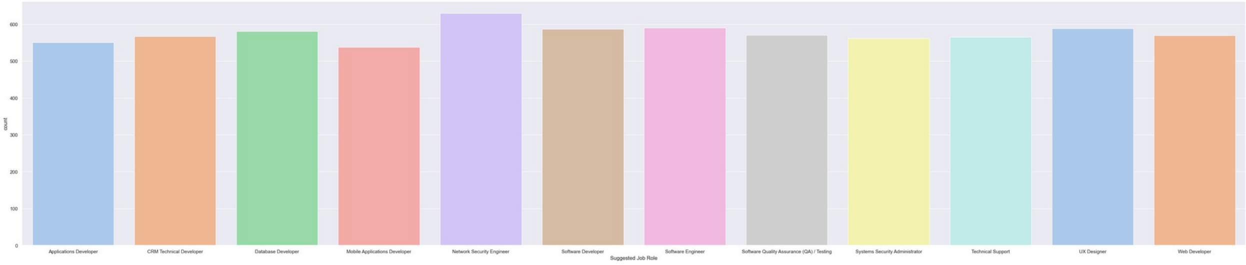
```
#menganalisa distribusi nilai dari kolom kategorikal dalam dataset
categorical_col = df[['self-learning capability?', 'Extra-courses
did','reading and writing skills', 'memory capability score',
                      'Taken inputs from seniors or elders',
'Management or Technical', 'hard/smart worker', 'worked in teams
ever?',
                      'Introvert', 'interested career area ']]
for i in categorical_col:
    print(df[i].value_counts(), end="\n\n")
```

```
self-learning capability?
yes    3496
no     3405
Name: count, dtype: int64

Extra-courses did
no     3529
yes    3372
Name: count, dtype: int64

reading and writing skills
excellent    2328
medium       2315
poor         2258
Name: count, dtype: int64

memory capability score
medium       2317
excellent    2303
poor         2281
Name: count, dtype: int64

Taken inputs from seniors or elders
yes    3501
no     3400
Name: count, dtype: int64

Management or Technical
Management    3461
Technical     3440
```

```
Name: count, dtype: int64

hard/smart worker
smart worker    3523
hard worker     3378
Name: count, dtype: int64

worked in teams ever?
no     3470
yes    3431
Name: count, dtype: int64

Introvert
yes    3544
no     3357
Name: count, dtype: int64

interested career area
system developer            1178
security                    1177
Business process analyst    1154
developer                   1145
testing                     1128
cloud computing             1119
Name: count, dtype: int64
```

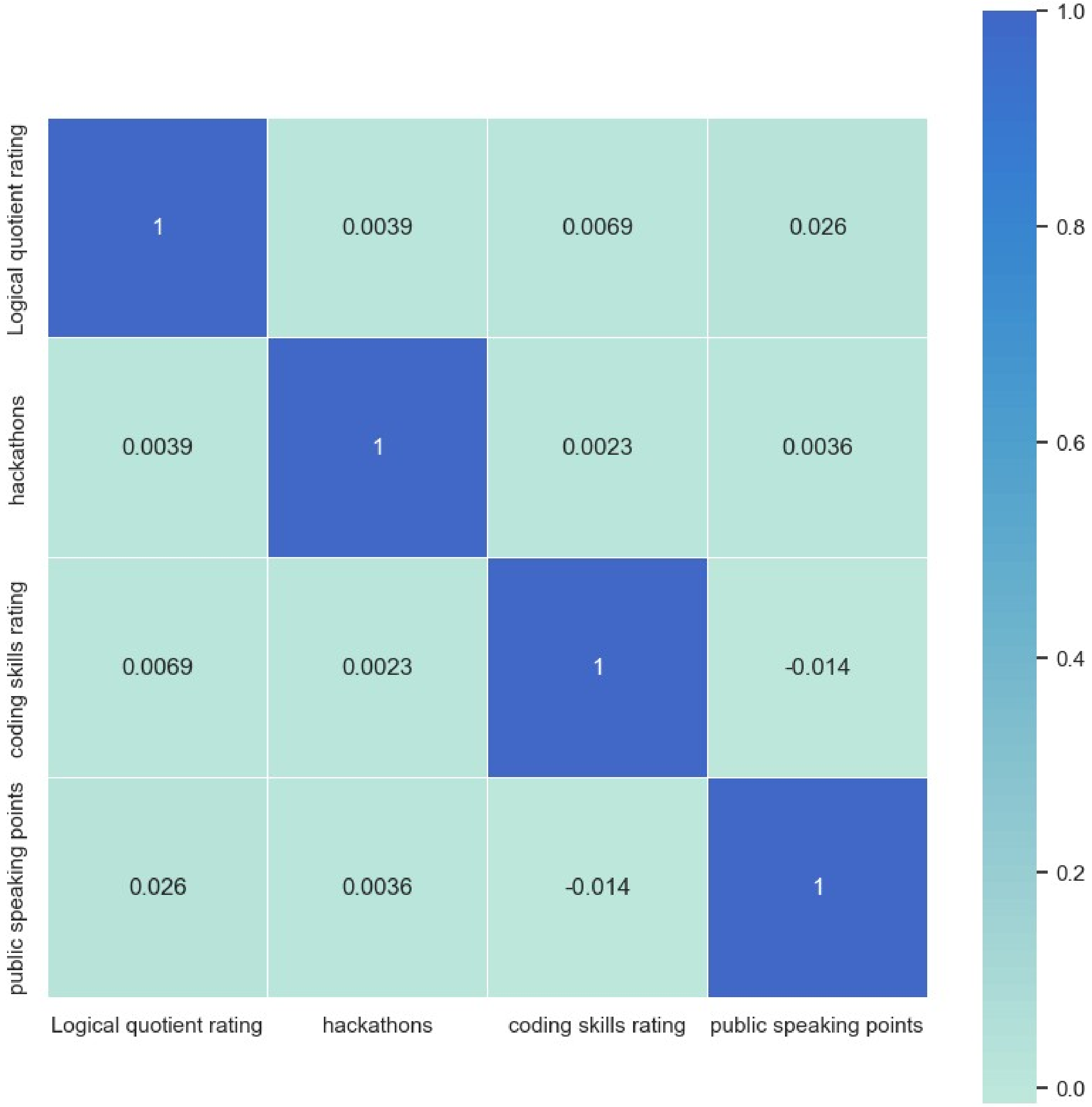
```
#visualisasi jumlah (count) setiap kategori di kolom "Suggested Job Role"
sns.set(rc={'figure.figsize':(50,10)})
sns.countplot(x=df["Suggested Job Role"], hue=df["Suggested Job Role"], palette='pastel', legend=False)
```

```
<Axes: xlabel='Suggested Job Role', ylabel='count'>
```



```
#heatmap kolom numerik
corr = df[['Logical quotient rating', 'hackathons',
           'coding skills rating', 'public speaking points']].corr()
f,axes = plt.subplots(1,1,figsize = (10,10))
sns.heatmap(corr,square=True,annot = True,linewidth = .4,center = 2,ax
= axes)
```

```
<Axes: >
```

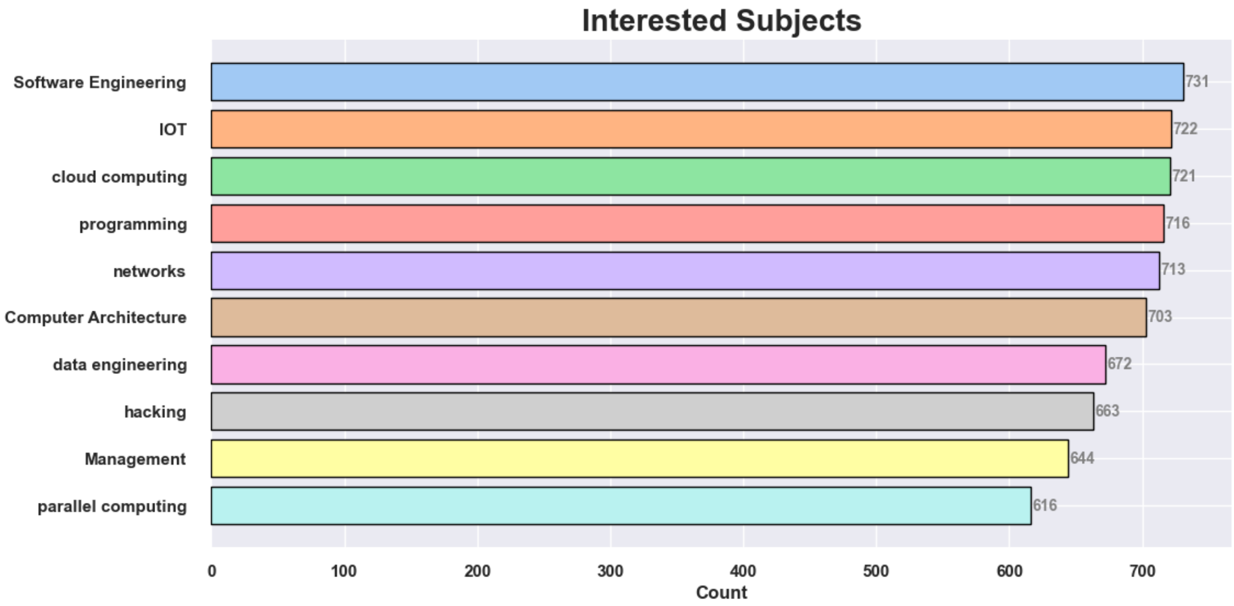


	Logical quotient rating	hackathons	coding skills rating	public speaking points
Logical quotient rating	1	0.0039	0.0069	0.026
hackathons	0.0039	1	0.0023	0.0036
coding skills rating	0.0069	0.0023	1	-0.014
public speaking points	0.026	0.0036	-0.014	1

```
#menghitung dan menampilkan jumlah frekuensi masing-masing nilai unik
dalam kolom "Interested subjects"
print(df["Interested subjects"].value_counts())

Interested subjects
Software Engineering    731
IOT                     722
cloud computing         721
programming             716
```

```
networks                    713
Computer Architecture       703
data engineering            672
hacking                     663
Management                  644
parallel computing          616
Name: count, dtype: int64
```

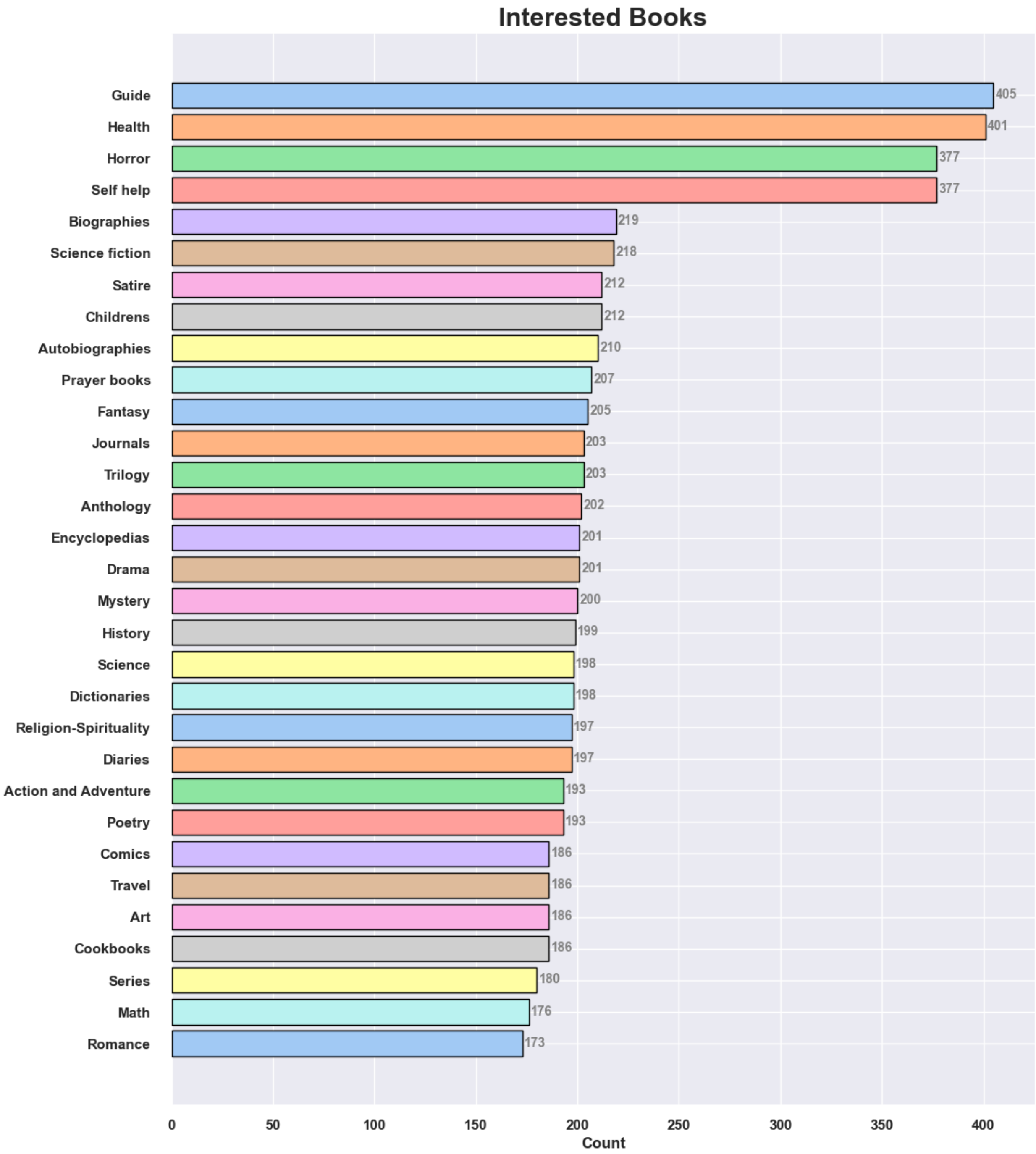
```python
# Figure Size
fig, ax = plt.subplots(figsize=(12,6))

# Horizontal Bar Plot
title_cnt=df["Interested
subjects"].value_counts().sort_values(ascending=False).reset_index()
mn= ax.barh(title_cnt.iloc[:,0],
title_cnt.iloc[:,1],edgecolor='black',
color=sns.color_palette('pastel',len(title_cnt)))

# Remove axes splines
for s in ['top','bottom','left','right']:
    ax.spines[s].set_visible(False)

# Remove x,y Ticks
ax.xaxis.set_ticks_position('none')
ax.yaxis.set_ticks_position('none')
# Add padding between axes and labels
ax.xaxis.set_tick_params(pad=5)
ax.yaxis.set_tick_params(pad=10)


# Show top values
ax.invert_yaxis()

# Add Plot Title
ax.set_title('Interested Subjects',weight='bold',fontsize=20)
ax.set_xlabel('Count', weight='bold')

# Add annotation to bars
for i in ax.patches:
    ax.text(i.get_width()+1, i.get_y()+0.5, str(round((i.get_width()),
2)),
             fontsize=10, fontweight='bold', color='grey')
plt.yticks(weight='bold')
plt.xticks(weight='bold')

# Show Plot
plt.show()
```



```
#menghitung dan mencetak jumlah frekuensi masing-masing tipe buku yang
ada dalam kolom "Interested Type of Books"
print(df["Interested Type of Books"].value_counts())

Interested Type of Books
Guide                       405
Health                      401
Horror                      377
Self help                   377
Biographies                 219
Science fiction             218
Childrens                   212
Satire                      212
Autobiographies             210
Prayer books                207
Fantasy                     205
Trilogy                     203
Journals                    203
Anthology                   202
Encyclopedias               201
Drama                       201
Mystery                     200
History                     199
Science                     198
Dictionaries                198
Diaries                     197
Religion-Spirituality       197
Action and Adventure        193
Poetry                      193
Cookbooks                   186
Art                         186
```

```
Comics                        186
Travel                        186
Series                        180
Math                          176
Romance                       173
Name: count, dtype: int64
```

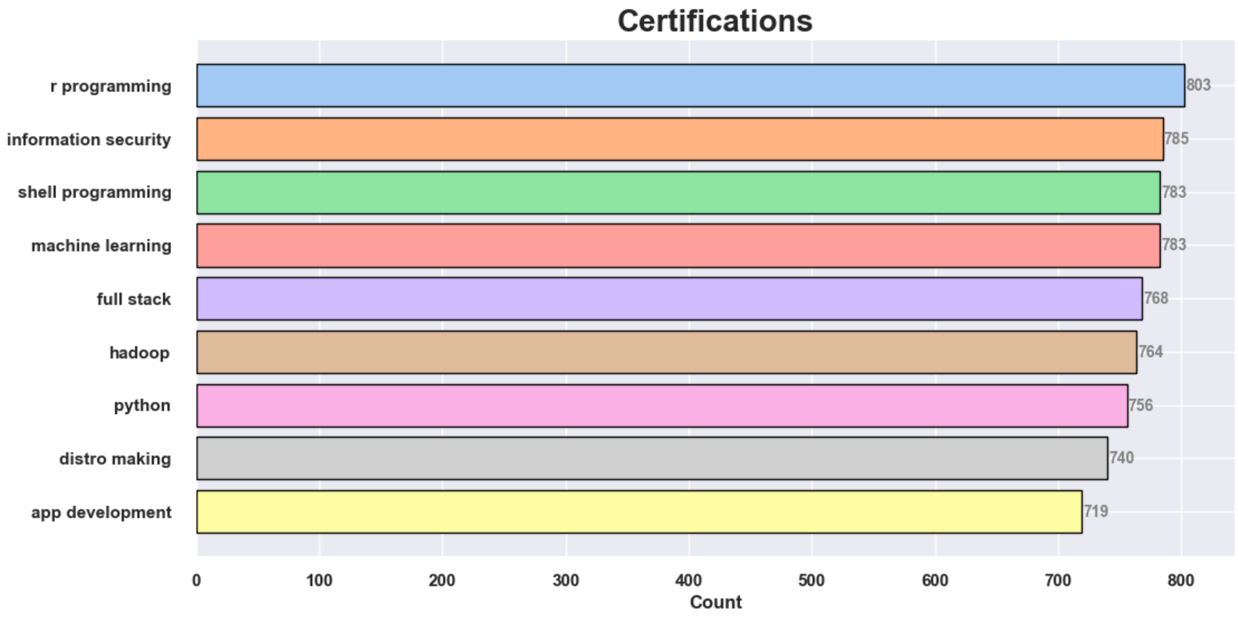
```python
# Figure Size
fig, ax = plt.subplots(figsize=(12,15))

# Horizontal Bar Plot
title_cnt=df["Interested Type of
Books"].value_counts().sort_values(ascending=False).reset_index()
mn= ax.barh(title_cnt.iloc[:,0],
title_cnt.iloc[:,1],edgecolor='black',
color=sns.color_palette('pastel',len(title_cnt)))



# Remove axes splines
for s in ['top','bottom','left','right']:
    ax.spines[s].set_visible(False)

# Remove x,y Ticks
ax.xaxis.set_ticks_position('none')
ax.yaxis.set_ticks_position('none')
# Add padding between axes and labels
ax.xaxis.set_tick_params(pad=5)
ax.yaxis.set_tick_params(pad=10)


# Show top values
ax.invert_yaxis()

# Add Plot Title
ax.set_title('Interested Books',weight='bold',fontsize=20)
ax.set_xlabel('Count', weight='bold')

# Add annotation to bars
for i in ax.patches:
    ax.text(i.get_width()+1, i.get_y()+0.5, str(round((i.get_width()),
2)),
             fontsize=10, fontweight='bold', color='grey')
plt.yticks(weight='bold')
plt.xticks(weight='bold')

# Show Plot
plt.show()
```



```
#menghitung dan mencetak frekuensi atau jumlah kemunculan nilai dalam
kolom "certifications"
print(df["certifications"].value_counts())

certifications
r programming           803
information security    785
shell programming       783
```

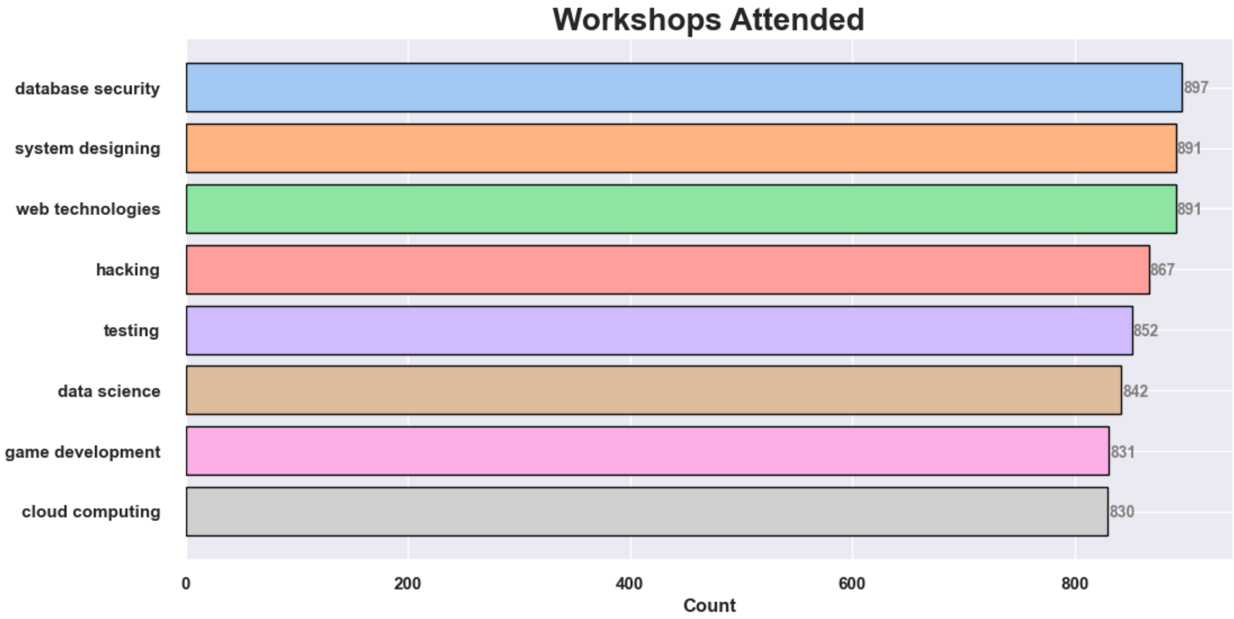
```
machine learning        783
full stack              768
hadoop                  764
python                  756
distro making           740
app development         719
Name: count, dtype: int64
```

```python
# Figure Size
fig, ax = plt.subplots(figsize=(12,6))

# Horizontal Bar Plot
title_cnt=df.certifications.value_counts().sort_values(ascending=False
).reset_index()
mn= ax.barh(title_cnt.iloc[:,0],
title_cnt.iloc[:,1],edgecolor='black',
color=sns.color_palette('pastel',len(title_cnt)))



# Remove axes splines
for s in ['top','bottom','left','right']:
    ax.spines[s].set_visible(False)

# Remove x,y Ticks
ax.xaxis.set_ticks_position('none')
ax.yaxis.set_ticks_position('none')
# Add padding between axes and labels
ax.xaxis.set_tick_params(pad=5)
ax.yaxis.set_tick_params(pad=10)


# Show top values
ax.invert_yaxis()

# Add Plot Title
ax.set_title('Certifications',weight='bold',fontsize=20)
ax.set_xlabel('Count', weight='bold')

# Add annotation to bars
for i in ax.patches:
    ax.text(i.get_width()+1, i.get_y()+0.5, str(round((i.get_width()),
2)),
             fontsize=10, fontweight='bold', color='grey')
plt.yticks(weight='bold')
plt.xticks(weight='bold')

# Show Plot
plt.show()
```



```
#mencetak jumlah kemunculan tiap nilai yang ada dalam kolom "workshops"
print(df["workshops"].value_counts())

workshops
database security    897
system designing     891
web technologies     891
hacking              867
testing              852
data science         842
game development     831
cloud computing      830
Name: count, dtype: int64

# Figure Size
fig, ax = plt.subplots(figsize=(12,6))

# Horizontal Bar Plot
title_cnt=df.workshops.value_counts().sort_values(ascending=False).reset_index()
mn= ax.barh(title_cnt.iloc[:,0],
title_cnt.iloc[:,1],edgecolor='black',
color=sns.color_palette('pastel',len(title_cnt)))



# Remove axes splines
for s in ['top','bottom','left','right']:
    ax.spines[s].set_visible(False)
```

```
# Remove x,y Ticks
ax.xaxis.set_ticks_position('none')
ax.yaxis.set_ticks_position('none')
# Add padding between axes and labels
ax.xaxis.set_tick_params(pad=5)
ax.yaxis.set_tick_params(pad=10)


# Show top values
ax.invert_yaxis()

# Add Plot Title
ax.set_title('Workshops Attended',weight='bold',fontsize=20)
ax.set_xlabel('Count', weight='bold')

# Add annotation to bars
for i in ax.patches:
    ax.text(i.get_width()+1, i.get_y()+0.5, str(round((i.get_width()),
2)),
             fontsize=10, fontweight='bold', color='grey')
plt.yticks(weight='bold')
plt.xticks(weight='bold')

# Show Plot
plt.show()
```

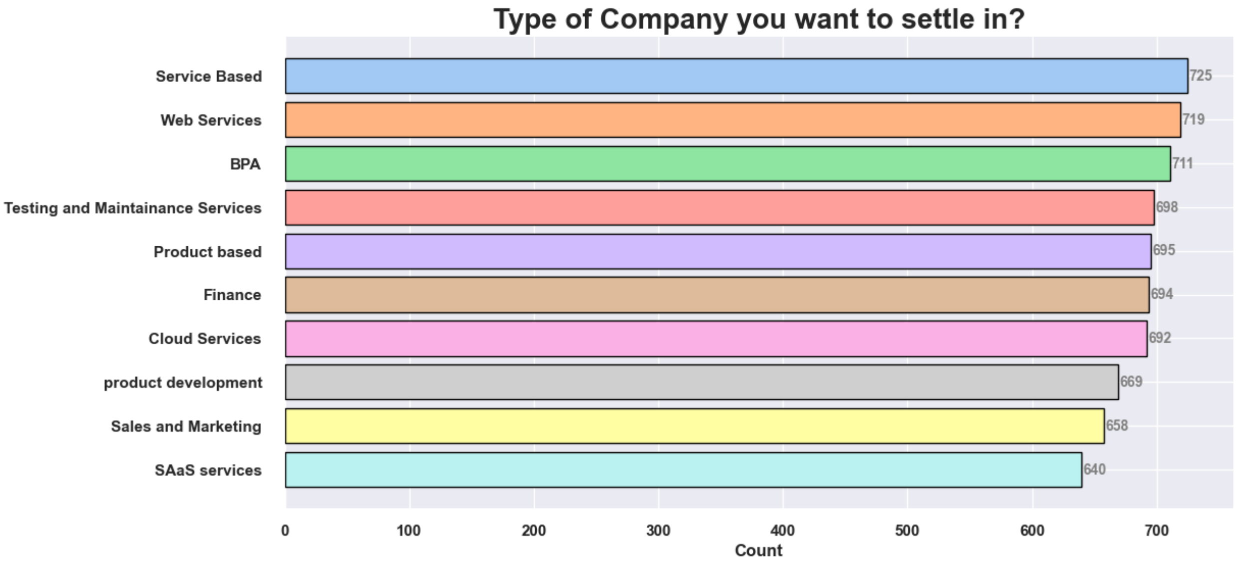


```
#mencetak jumlah kemunculan tiap nilai unik dalam kolom "Type of
company want to settle in?"
print(df["Type of company want to settle in?"].value_counts())
```

```
Type of company want to settle in?
Service Based                           725
Web Services                            719
BPA                                     711
Testing and Maintainance Services       698
Product based                           695
Finance                                 694
Cloud Services                          692
product development                     669
Sales and Marketing                     658
SAaS services                           640
Name: count, dtype: int64
```

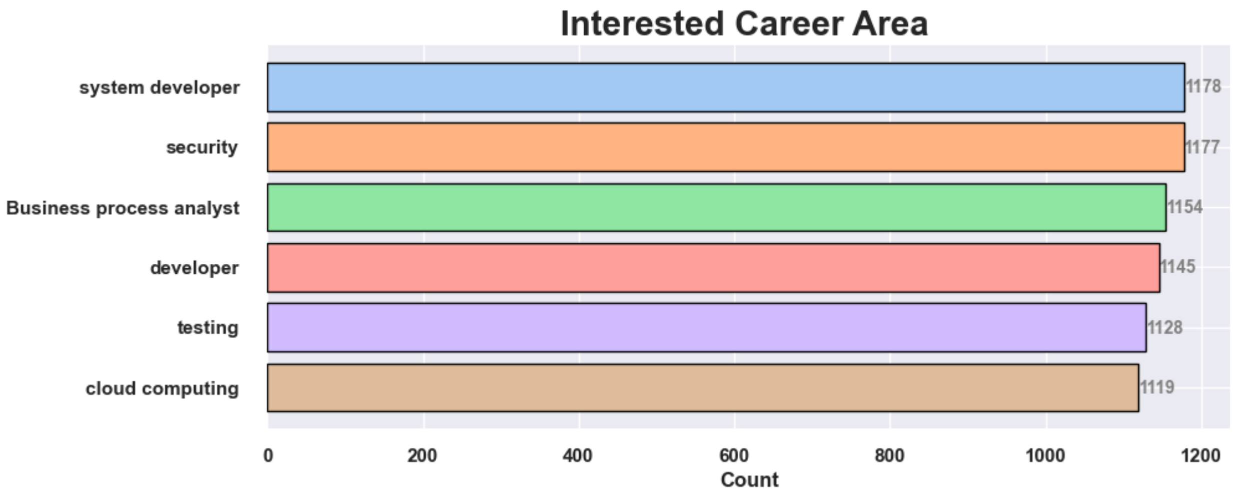
```python
# Figure Size
fig, ax = plt.subplots(figsize=(12,6))

# Horizontal Bar Plot
title_cnt=df["Type of company want to settle
in?"].value_counts().sort_values(ascending=False).reset_index()
mn= ax.barh(title_cnt.iloc[:,0],
title_cnt.iloc[:,1],edgecolor='black',
color=sns.color_palette('pastel',len(title_cnt)))



# Remove axes splines
for s in ['top','bottom','left','right']:
    ax.spines[s].set_visible(False)

# Remove x,y Ticks
ax.xaxis.set_ticks_position('none')
ax.yaxis.set_ticks_position('none')
# Add padding between axes and labels
ax.xaxis.set_tick_params(pad=5)
ax.yaxis.set_tick_params(pad=10)


# Show top values
ax.invert_yaxis()

# Add Plot Title
ax.set_title('Type of Company you want to settle
in?',weight='bold',fontsize=20)
ax.set_xlabel('Count', weight='bold')

# Add annotation to bars
for i in ax.patches:
    ax.text(i.get_width()+1, i.get_y()+0.5, str(round((i.get_width()),
2)),
             fontsize=10, fontweight='bold', color='grey')
```

```
plt.yticks(weight='bold')
plt.xticks(weight='bold')

# Show Plot
plt.show()
```



```
#mencetak jumlah kemunculan setiap nilai unik dalam kolom "interested
career area"
print(df["interested career area "].value_counts())

interested career area
system developer           1178
security                   1177
Business process analyst   1154
developer                  1145
testing                    1128
cloud computing            1119
Name: count, dtype: int64

# Figure Size
fig, ax = plt.subplots(figsize=(10,4)) #width,height

# Horizontal Bar Plot
title_cnt=df["interested career area
"].value_counts().sort_values(ascending=False).reset_index()
mn= ax.barh(title_cnt.iloc[:,0],
title_cnt.iloc[:,1],edgecolor='black',
color=sns.color_palette('pastel',len(title_cnt)))



# Remove axes splines
```

```
for s in ['top','bottom','left','right']:
    ax.spines[s].set_visible(False)

# Remove x,y Ticks
ax.xaxis.set_ticks_position('none')
ax.yaxis.set_ticks_position('none')
# Add padding between axes and labels
ax.xaxis.set_tick_params(pad=5)
ax.yaxis.set_tick_params(pad=10)


# Show top values
ax.invert_yaxis()

# Add Plot Title
ax.set_title('Interested Career Area ',weight='bold',fontsize=20)
ax.set_xlabel('Count', weight='bold')

# Add annotation to bars
for i in ax.patches:
    ax.text(i.get_width()+1, i.get_y()+0.5, str(round((i.get_width()),
2)),
             fontsize=10, fontweight='bold', color='grey')
plt.yticks(weight='bold')
plt.xticks(weight='bold')

# Show Plot
plt.show()
```



```
#mengganti nilai pada kolom-kolom tertentu dalam DataFrame df dari
bentuk teks ("yes" dan "no") menjadi bentuk numerik (1 dan 0)
cols = df[["self-learning capability?", "Extra-courses did", "Taken
inputs from seniors or elders", "worked in teams ever?", "Introvert"]]
```

```
# Looping untuk setiap kolom di cols dan mengganti nilai "yes" dan
"no" dengan 1 dan 0
for col in cols.columns:
    df[col] = df[col].replace({"yes": 1, "no": 0})


print("\n\nList of Categorical features: \n" ,
df.select_dtypes(include=['object']).columns.tolist())
```

```
List of Categorical features: 
 ['certifications', 'workshops', 'reading and writing skills', 'memory
capability score', 'Interested subjects', 'interested career area ',
'Type of company want to settle in?', 'Interested Type of Books',
'Management or Technical', 'hard/smart worker', 'Suggested Job Role']
```

```
# Mengganti nilai-nilai dalam kolom yang disebutkan dengan kode
numerik
mycol = ["reading and writing skills", "memory capability score"]
cleanup_nums = {
    "reading and writing skills": {"poor": 0, "medium": 1,
"excellent": 2},
    "memory capability score": {"poor": 0, "medium": 1, "excellent":
2}
}

# Menggunakan .replace() untuk mengganti nilai-nilai di DataFrame
df = df.replace(cleanup_nums)

# Mengubah kolom kategori menjadi tipe 'category' dan menambahkan
kolom kode kategori
category_cols = ['certifications', 'workshops', 'Interested subjects',
'interested career area ', 'Type of company want to settle in?',
                 'Interested Type of Books']
for col in category_cols:
    df[col] = df[col].astype('category')
    df[col + "_code"] = df[col].cat.codes

# Menampilkan daftar kolom kategori
print("\n\nList of Categorical features: \n",
df.select_dtypes(include=['object']).columns.tolist())
```

```
List of Categorical features: 
 ['Management or Technical', 'hard/smart worker', 'Suggested Job
Role']
```

```
#menampilkan nilai unik
print(df['Management or Technical'].unique())
print(df['hard/smart worker'].unique())
```

```
['Management' 'Technical']
['smart worker' 'hard worker']
```

```
#mengonversi kolom kategorikal menjadi format yang bisa diolah oleh
model machine learning
df = pd.get_dummies(df, columns=["Management or Technical",
"hard/smart worker"], prefix=["A", "B"])
df.head()
```

```
   Logical quotient rating  hackathons  coding skills rating  \
0                        5           0                     6
1                        7           6                     4
2                        2           3                     9
3                        2           6                     3
4                        2           0                     3

   public speaking points  self-learning capability?  Extra-courses
did  \
0                       2                          1
0
1                       3                          0
1
2                       1                          0
1
3                       5                          0
1
4                       4                          1
0

         certifications          workshops  reading and writing skills
\
0  information security            testing                           0

1     shell programming            testing                           2

2  information security            testing                           2

3         r programming  database security                           2

4         distro making   game development                           2


   memory capability score  ... certifications_code workshops_code  \
0                        0  ...                   4              6
1                        1  ...                   8              6
2                        0  ...                   4              6
3                        0  ...                   7              2
```

```
4                        1  ...                   1              3

  Interested subjects_code  interested career area _code  \
0                        9                             5
1                        2                             4
2                        5                             0
3                        7                             5
4                        3                             4

  Type of company want to settle in?_code  Interested Type of
Books_code  \
0                                       0
28
1                                       1
3
2                                       9
29
3                                       7
13
4                                       0
14

   A_Management A_Technical  B_hard worker  B_smart worker
0          True       False          False            True
1         False        True           True           False
2         False        True          False            True
3          True       False          False            True
4         False        True           True           False

[5 rows x 28 columns]
```

```
#menampilkan semua kolom numerik
print("List of Numerical features: \n" ,
df.select_dtypes(include=np.number).columns.tolist())
```

```
List of Numerical features:
 ['Logical quotient rating', 'hackathons', 'coding skills rating',
'public speaking points', 'self-learning capability?', 'Extra-courses
did', 'reading and writing skills', 'memory capability score', 'Taken
inputs from seniors or elders', 'worked in teams ever?', 'Introvert',
'certifications_code', 'workshops_code', 'Interested subjects_code',
'interested career area _code', 'Type of company want to settle in?
_code', 'Interested Type of Books_code']
```

```
feed = df[['Logical quotient rating', 'coding skills rating',
'hackathons', 'public speaking points', 'self-learning
capability?','Extra-courses did',
           'Taken inputs from seniors or elders', 'worked in teams
ever?', 'Introvert', 'reading and writing skills', 'memory capability
score',
           'B_hard worker', 'B_smart worker', 'A_Management',
'A_Technical', 'Interested subjects_code', 'Interested Type of
Books_code', 'certifications_code',
           'workshops_code', 'Type of company want to settle in?
_code',  'interested career area _code',
             'Suggested Job Role']]

# Taking all independent variable columns
df_train_x = feed.drop('Suggested Job Role',axis = 1)

# Target variable column
df_train_y = feed['Suggested Job Role']

x_train, x_test, y_train, y_test = train_test_split(df_train_x,
df_train_y, test_size=0.20, random_state=42)

userdata = [['7','6','6','8','3','5','4', '4', '7', '3', '3', '6','8',

                    '7','5','7','4','5','6','8','8']]
ynewclass = dtree.predict(userdata)
ynew = dtree.predict_proba(userdata)
print(ynewclass)
print("Probabilities of all classes: ", ynew)
print("Probability of Predicted class : ", np.max(ynew))
```

```
----------------------------------------------------------------------
-----
NameError                                 Traceback (most recent call
last)
Cell In[40], line 3
      1 userdata = [['7','6','6','8','3','5','4', '4', '7', '3', '3',
'6','8',
      2                    '7','5','7','4','5','6','8','8']]
----> 3 ynewclass = dtree.predict(userdata)
      4 ynew = dtree.predict_proba(userdata)
      5 print(ynewclass)

NameError: name 'dtree' is not defined
```

```
ynewclass = rf.predict(userdata)
ynew = rf.predict_proba(userdata)
print(ynewclass)
print("Probabilities of all classes: ", ynew)
print("Probability of Predicted class : ", np.max(ynew))
```

```
----------------------------------------------------------------------
-----
NameError                                 Traceback (most recent call
last)
Cell In[39], line 1
----> 1 ynewclass = rf.predict(userdata)
```

```
      2 ynew = rf.predict_proba(userdata)
      3 print(ynewclass)

NameError: name 'userdata' is not defined
```