main

February 15, 2025

[1]:
```
import pandas as pd
import numpy as np
!pip install xlrd
```

```
Requirement already satisfied: xlrd in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (2.0.1)


[notice] A new release of pip available: 22.3.1 -> 25.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

[2]:
```
df=pd.read_csv(r"C:\Users\GIM\Downloads\clean.csv")
```

[3]:
```
df.head()
```

[3]:

	female	age	educ	inc_q	emp_in	account	fin4	fin7	fin8	fin10	…	\
0	2	43.0	2	4	1.0	1	NaN	NaN	NaN	NaN	…	
1	2	55.0	1	3	1.0	0	NaN	NaN	NaN	NaN	…	
2	1	15.0	1	2	2.0	0	NaN	NaN	NaN	NaN	…	
3	2	23.0	1	4	1.0	0	NaN	NaN	NaN	NaN	…	
4	1	46.0	1	1	2.0	0	NaN	NaN	NaN	NaN	…	

	fin44a	fin44b	fin44c	fin44d	fin45	saved	borrowed	pay_utilities	\
0	1	2	3	4	1.0	0	1	1	
1	2	1	1	1	3.0	0	1	4	
2	2	1	1	1	4.0	0	1	4	
3	2	2	1	2	3.0	0	0	4	
4	1	2	4	3	1.0	0	1	4	

	remittances	year
0	5.0	2021
1	5.0	2021
2	3.0	2021
3	5.0	2021
4	5.0	2021

[5 rows x 26 columns]

[4]:
```
for col in ['fin4', 'fin7', 'fin8', 'fin10']:
    new_col_name = f'{col}'
    df[new_col_name]=np.where(df[col]==1, df[col], 0)

mean_fin24a= df['fin24a'].mean()
mean_fin24b= df['fin24b'].mean()
mean_fin45=df['fin45'].mean()
mean_r=df['remittances'].mean()
mean_age=df['age'].mean()

for col, mean_val in [('fin24a', mean_fin24a), ('fin24b', mean_fin24b),␣
 ↪('fin45', mean_fin45), ('remittances', mean_r), ('age', mean_age)]:
    df[col]=np.where(pd.isna(df[col]), mean_val, df[col])

mode_e=df['emp_in'].mode()[0]
df['emp_in'].fillna(mode_e, inplace=True)
```

```
C:\Users\GIM\AppData\Local\Temp\ipykernel_10320\3134576037.py:15: FutureWarning:
A value is trying to be set on a copy of a DataFrame or Series through chained
assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
instead, to perform the operation inplace on the original object.


  df['emp_in'].fillna(mode_e, inplace=True)
```

[5]:
```
df.head()
```

[5]:

	female	age	educ	inc_q	emp_in	account	fin4	fin7	fin8	fin10	…	\
0	2	43.0	2	4	1.0	1	0.0	0.0	0.0	0.0	…	
1	2	55.0	1	3	1.0	0	0.0	0.0	0.0	0.0	…	
2	1	15.0	1	2	2.0	0	0.0	0.0	0.0	0.0	…	
3	2	23.0	1	4	1.0	0	0.0	0.0	0.0	0.0	…	
4	1	46.0	1	1	2.0	0	0.0	0.0	0.0	0.0	…	

	fin44a	fin44b	fin44c	fin44d	fin45	saved	borrowed	pay_utilities	\
0	1	2	3	4	1.0	0	1	1	
1	2	1	1	1	3.0	0	1	4	
2	2	1	1	1	4.0	0	1	4	
3	2	2	1	2	3.0	0	0	4	
4	1	2	4	3	1.0	0	1	4	

	remittances	year

```
0          5.0  2021
1          5.0  2021
2          3.0  2021
3          5.0  2021
4          5.0  2021

[5 rows x 26 columns]
```

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

[6]:
```
female               0
age                  0
educ                 0
inc_q                0
emp_in               0
account              0
fin4                 0
fin7                 0
fin8                 0
fin10                0
fin24a               0
fin24b               0
fin43a          128582
fin43b          130846
fin43d          132062
fin43e          142569
fin44a               0
fin44b               0
fin44c               0
fin44d               0
fin45                0
saved                0
borrowed             0
pay_utilities        0
remittances          0
year                 0
dtype: int64
```

[7]:
```
df.columns.tolist()
```

[7]:
```
['female',
 'age',
 'educ',
 'inc_q',
 'emp_in',
 'account',
 'fin4',
```

```
 'fin7',
 'fin8',
 'fin10',
 'fin24a',
 'fin24b',
 'fin43a',
 'fin43b',
 'fin43d',
 'fin43e',
 'fin44a',
 'fin44b',
 'fin44c',
 'fin44d',
 'fin45',
 'saved',
 'borrowed',
 'pay_utilities',
 'remittances',
 'year']
```

[8]:
```
dff=df[['female',
 'age',
 'educ',
 'inc_q',
 'emp_in',
 'account',
 'fin4',
 'fin7',
 'fin8',
 'fin10',
 'fin24a',
 'fin24b',
 'fin44a',
 'fin44b',
 'fin44c',
 'fin44d',
 'fin45',
 'saved',
 'borrowed',
 'pay_utilities',
 'remittances',
 'year']]
```

[9]:
```
dff.isnull().sum()
```

[9]:
```
female          0
age             0
```

```
educ              0
inc_q             0
emp_in            0
account           0
fin4              0
fin7              0
fin8              0
fin10             0
fin24a            0
fin24b            0
fin44a            0
fin44b            0
fin44c            0
fin44d            0
fin45             0
saved             0
borrowed          0
pay_utilities     0
remittances       0
year              0
dtype: int64
```

[10]:
```
dff.to_csv('data.csv', index=False)
```

[11]:
```
dff.shape
```

[11]:
```
(143887, 22)
```

[12]:
```
!pip install matplotlib
import matplotlib.pyplot as plt
```

```
Requirement already satisfied: matplotlib in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (3.10.0)
Requirement already satisfied: contourpy>=1.0.1 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib) (1.3.1)
Requirement already satisfied: cycler>=0.10 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib) (4.56.0)
Requirement already satisfied: kiwisolver>=1.3.1 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib) (1.4.8)
Requirement already satisfied: numpy>=1.23 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib) (2.0.2)
```

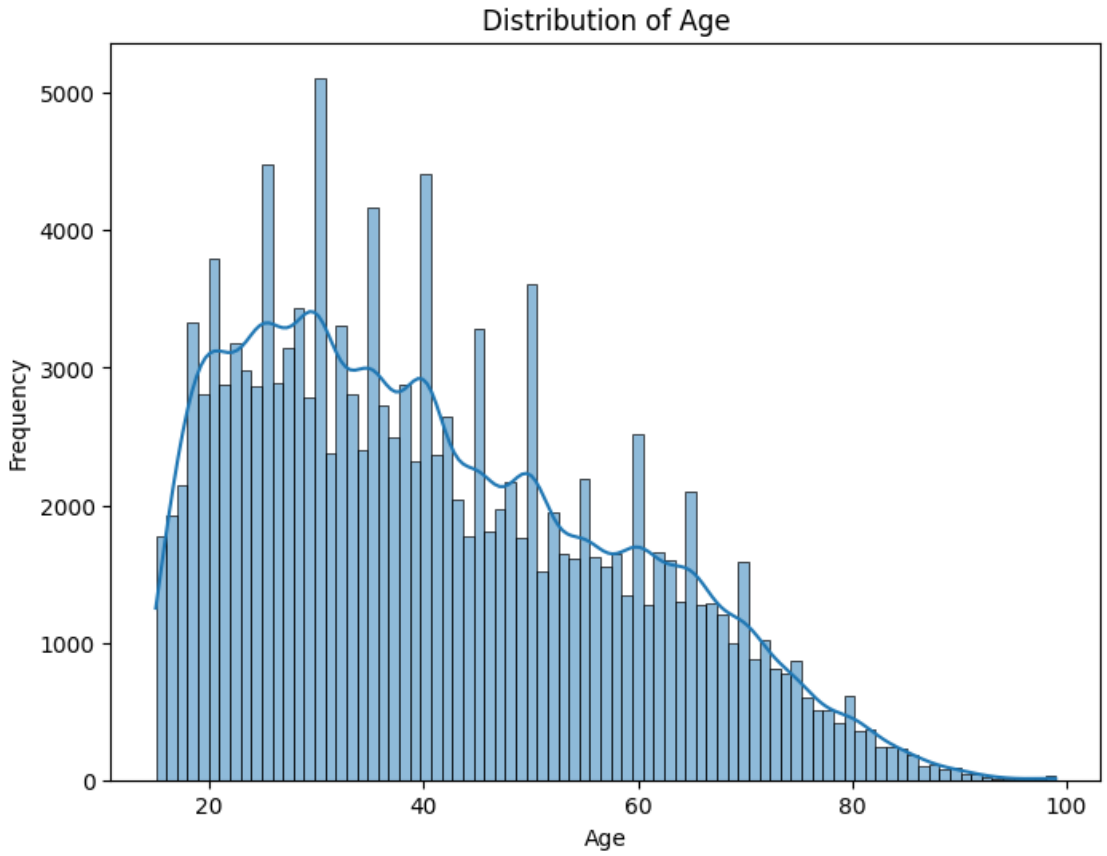
```
Requirement already satisfied: packaging>=20.0 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib) (24.2)
Requirement already satisfied: pillow>=8 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib) (11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib) (3.2.1)
Requirement already satisfied: python-dateutil>=2.7 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
python-dateutil>=2.7->matplotlib) (1.17.0)


[notice] A new release of pip available: 22.3.1 -> 25.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

[13]:
```
!pip install seaborn
import seaborn as sns
```

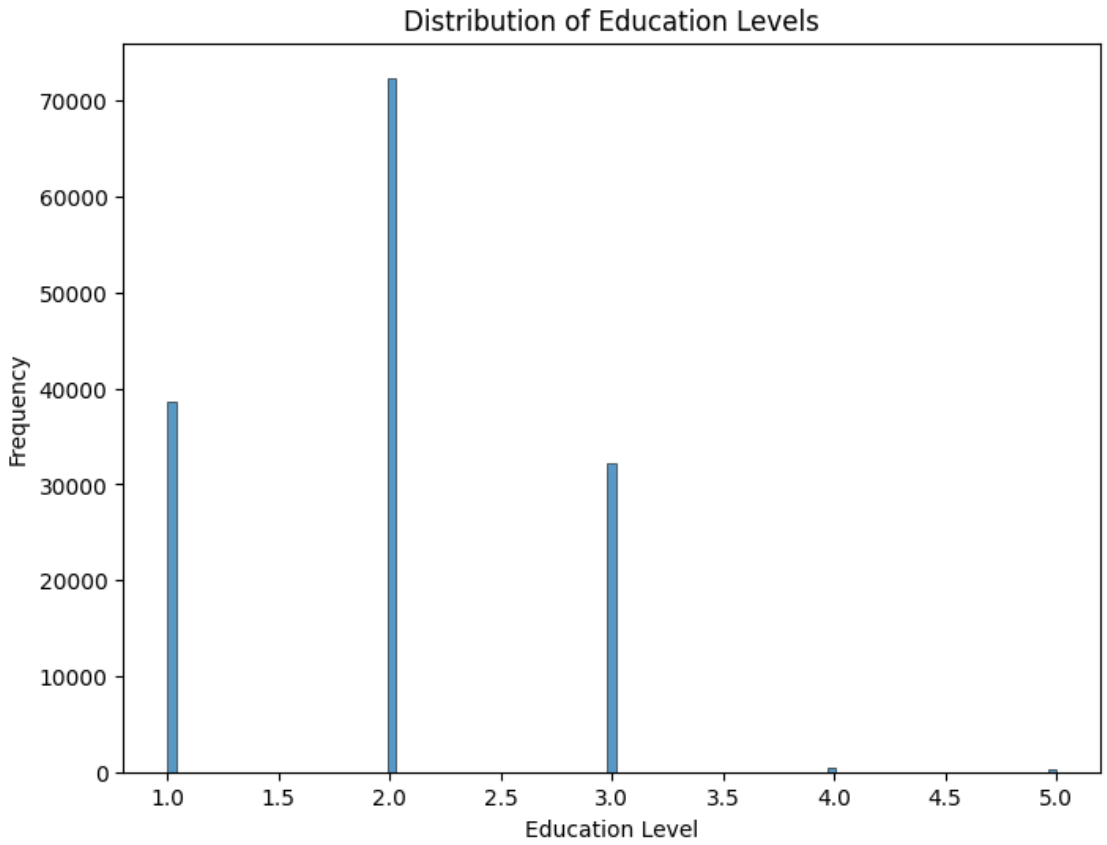
```
Requirement already satisfied: seaborn in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (0.13.2)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
seaborn) (2.0.2)
Requirement already satisfied: pandas>=1.2 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
seaborn) (2.2.3)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
seaborn) (3.10.0)
Requirement already satisfied: contourpy>=1.0.1 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (1.3.1)
Requirement already satisfied: cycler>=0.10 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (4.56.0)
Requirement already satisfied: kiwisolver>=1.3.1 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (1.4.8)
Requirement already satisfied: packaging>=20.0 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (24.2)
```

```
Requirement already satisfied: pillow>=8 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (3.2.1)
Requirement already satisfied: python-dateutil>=2.7 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
pandas>=1.2->seaborn) (2025.1)
Requirement already satisfied: tzdata>=2022.7 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
pandas>=1.2->seaborn) (2025.1)
Requirement already satisfied: six>=1.5 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
python-dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.17.0)


[notice] A new release of pip available: 22.3.1 -> 25.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

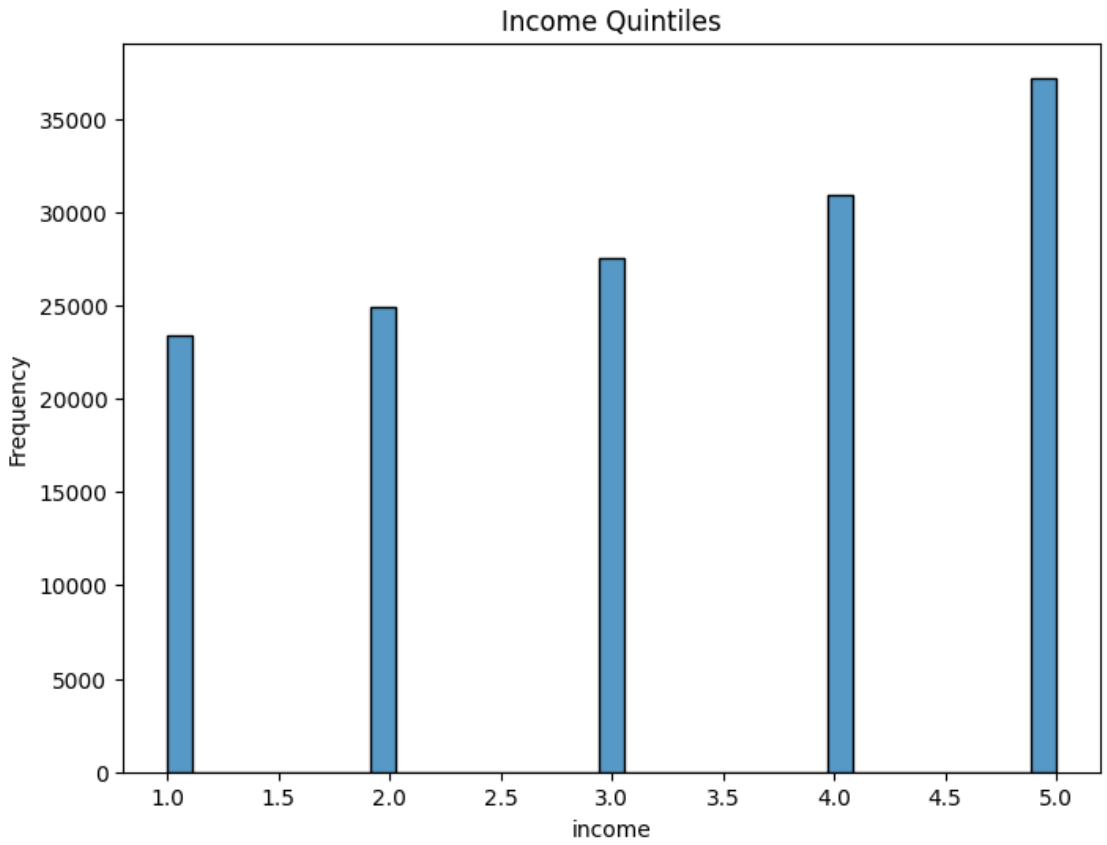
[14]:
```
plt.figure(figsize=(8, 6))
sns.histplot(dff['age'], kde=True)
plt.title('Distribution of Age')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```



[15]:
```
plt.figure(figsize=(8, 6))
sns.histplot(dff['educ'], kde=False) # Example: If 'educ' is categorical
plt.title('Distribution of Education Levels')
plt.xlabel('Education Level')
plt.ylabel('Frequency')
plt.show()
```

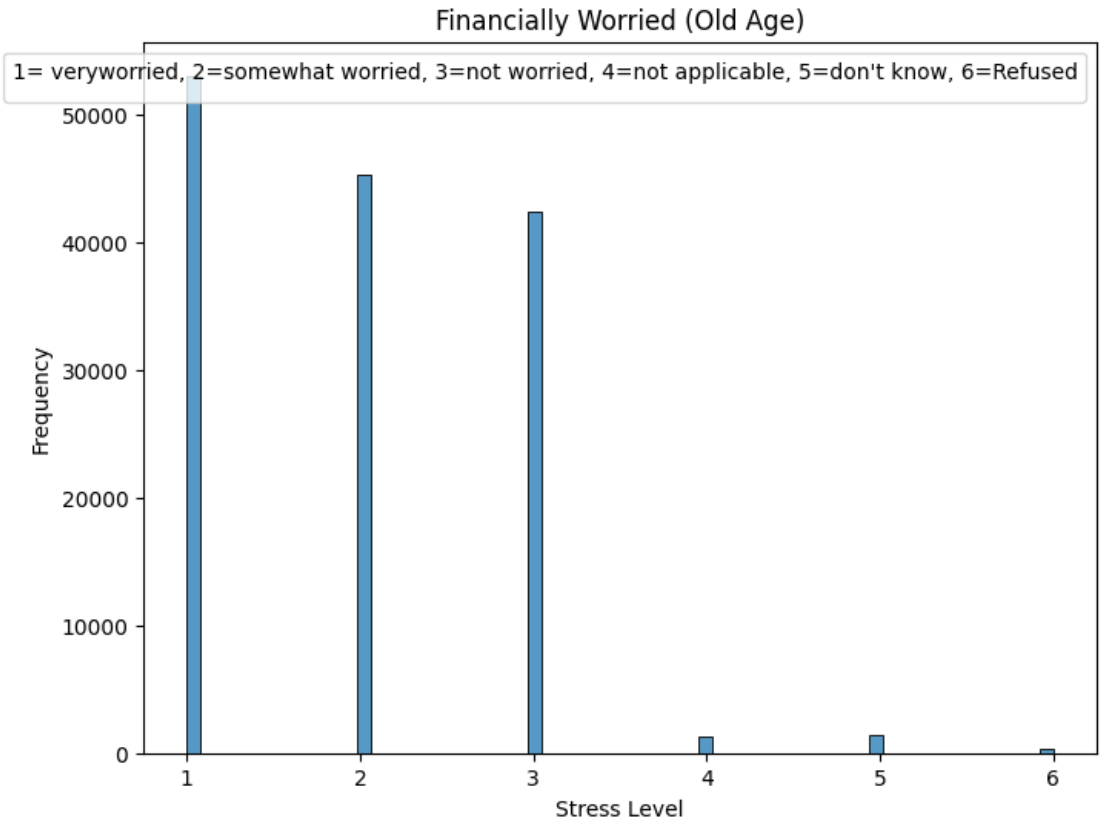


[16]:
```
plt.figure(figsize=(8, 6))
sns.histplot(dff['inc_q'], kde=False) # Example: If 'educ' is categorical
plt.title('Income Quintiles')
plt.xlabel('income')
plt.ylabel('Frequency')
plt.show()
```



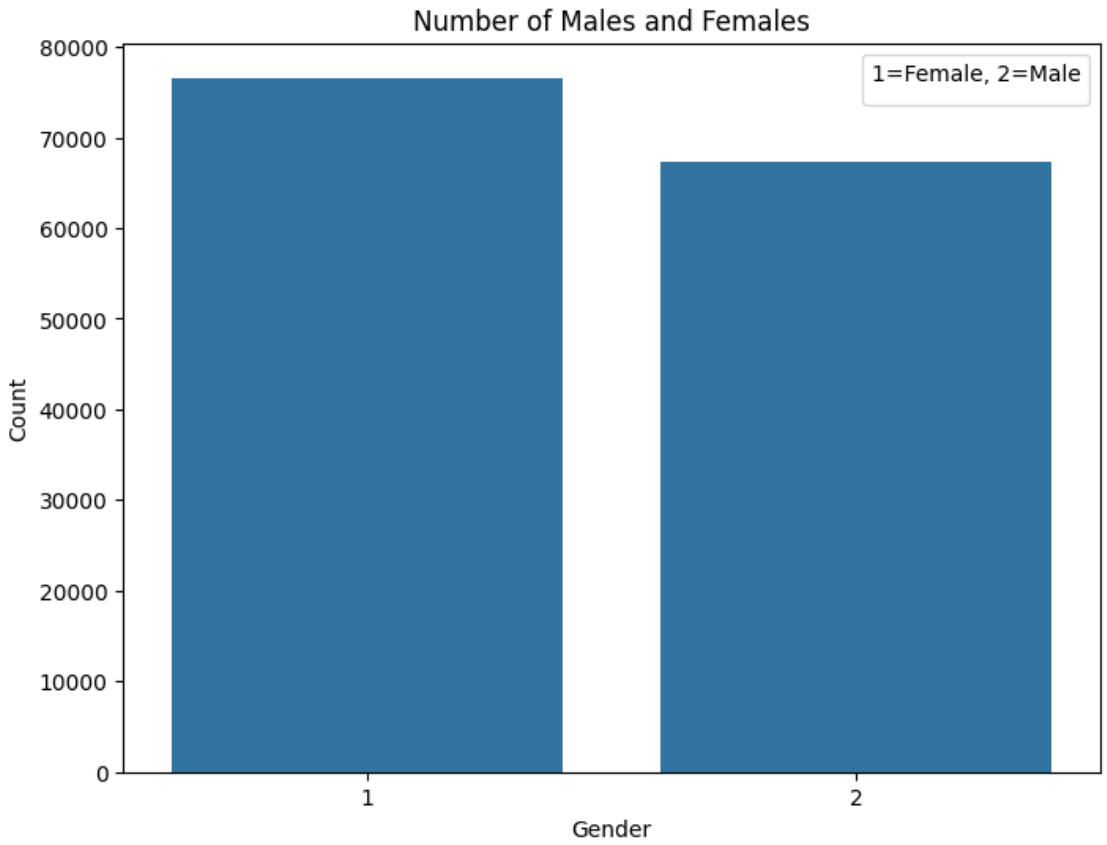
[17]:
```
plt.figure(figsize=(8, 6))
sns.histplot(dff['fin44a'], kde=False) # Example: If 'educ' is categorical
plt.title('Financially Worried (Old Age)')
plt.xlabel('Stress Level')
plt.ylabel('Frequency')
plt.legend(title="1= veryworried, 2=somewhat worried, 3=not worried, 4=not␣
 ↪applicable, 5=don't know, 6=Refused", loc="upper right")
plt.show()
```

```
C:\Users\GIM\AppData\Local\Temp\ipykernel_10320\3993712078.py:6: UserWarning: No
artists with labels found to put in legend.  Note that artists whose label start
with an underscore are ignored when legend() is called with no argument.
  plt.legend(title="1= veryworried, 2=somewhat worried, 3=not worried, 4=not
applicable, 5=don't know, 6=Refused", loc="upper right")
```

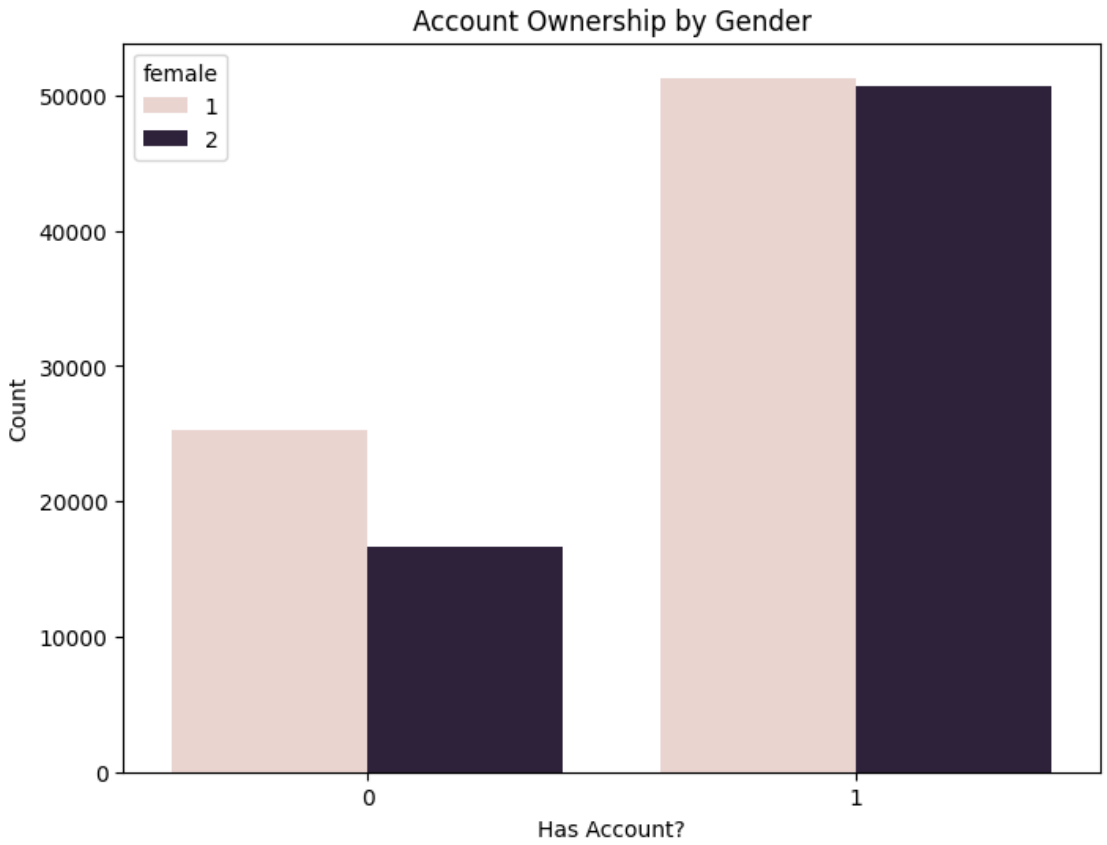


[18]:
```
plt.figure(figsize=(8, 6))
sns.countplot(x='female', data=dff)  # Easier for count of categories
plt.title('Number of Males and Females')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.legend(title="1=Female, 2=Male")
plt.show()
```

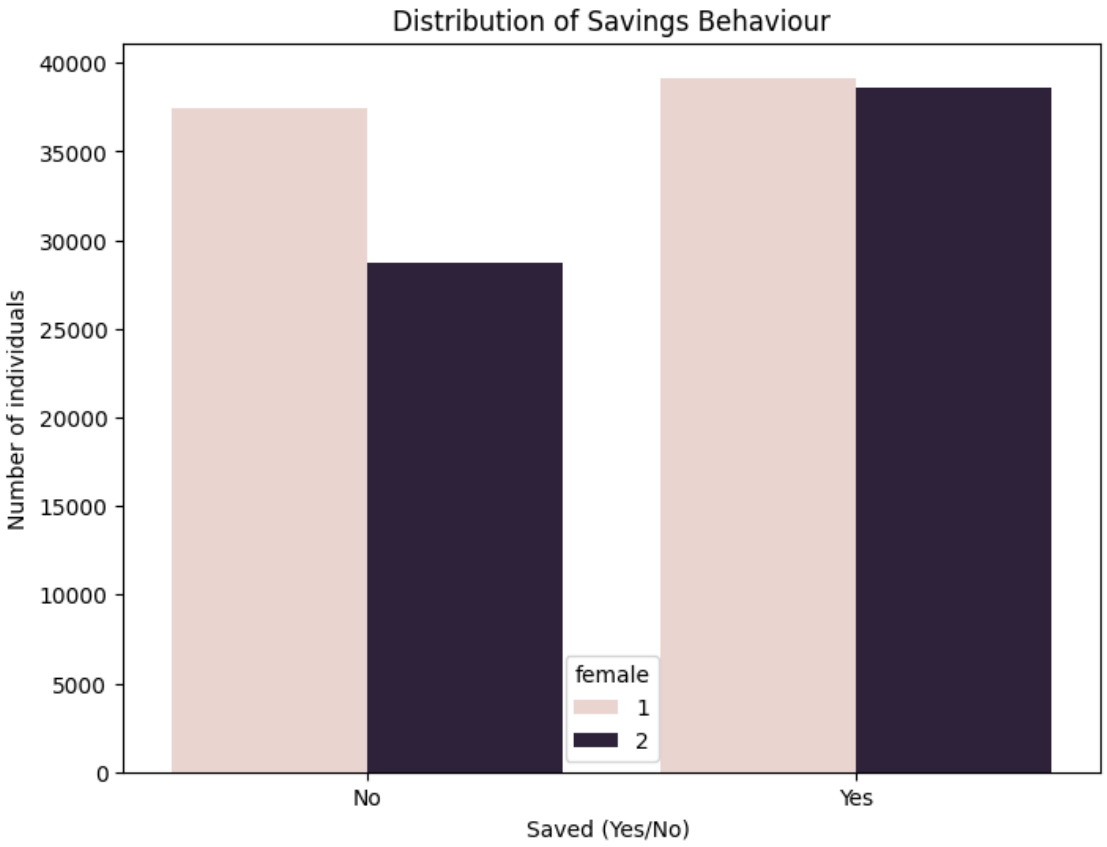
```
C:\Users\GIM\AppData\Local\Temp\ipykernel_10320\1875175662.py:6: UserWarning: No
artists with labels found to put in legend.  Note that artists whose label start
with an underscore are ignored when legend() is called with no argument.
  plt.legend(title="1=Female, 2=Male")
```



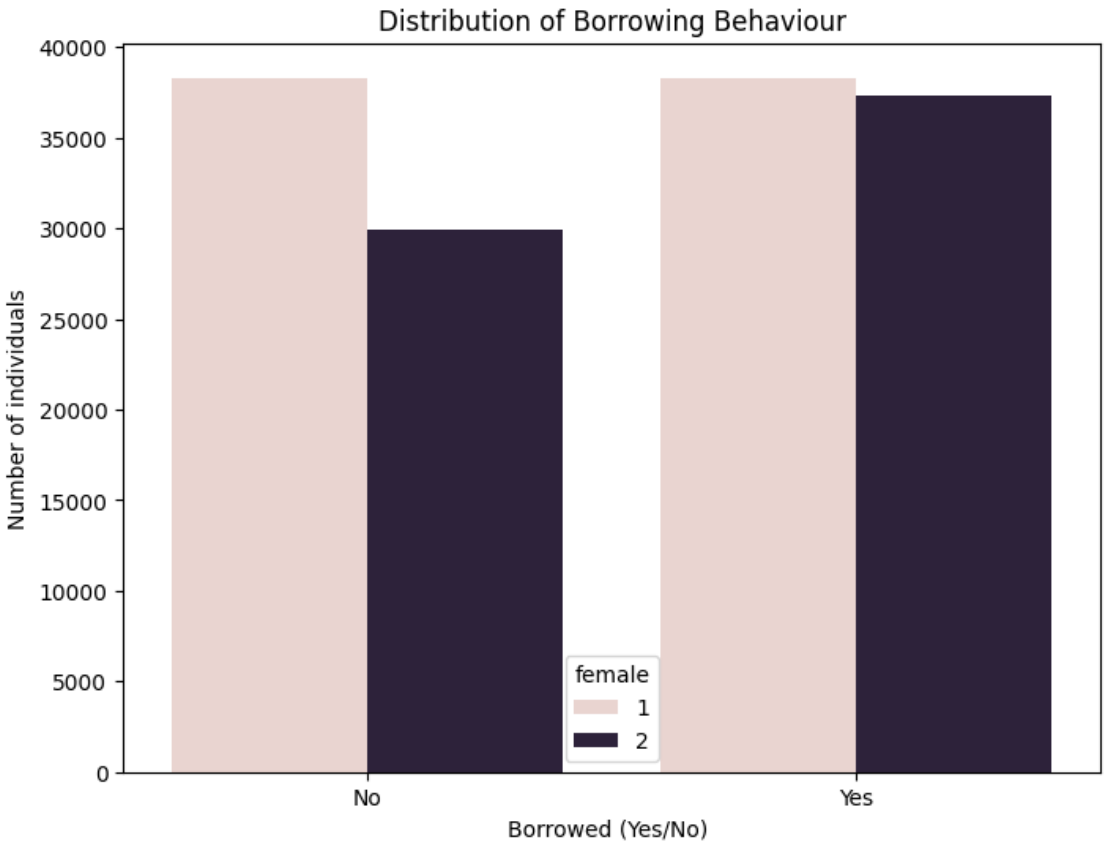
[19]:
```
plt.figure(figsize=(8, 6))
sns.countplot(x='account', data=dff, hue='female')
plt.title('Account Ownership by Gender')
plt.xlabel('Has Account?')
plt.ylabel('Count')
plt.show()
```



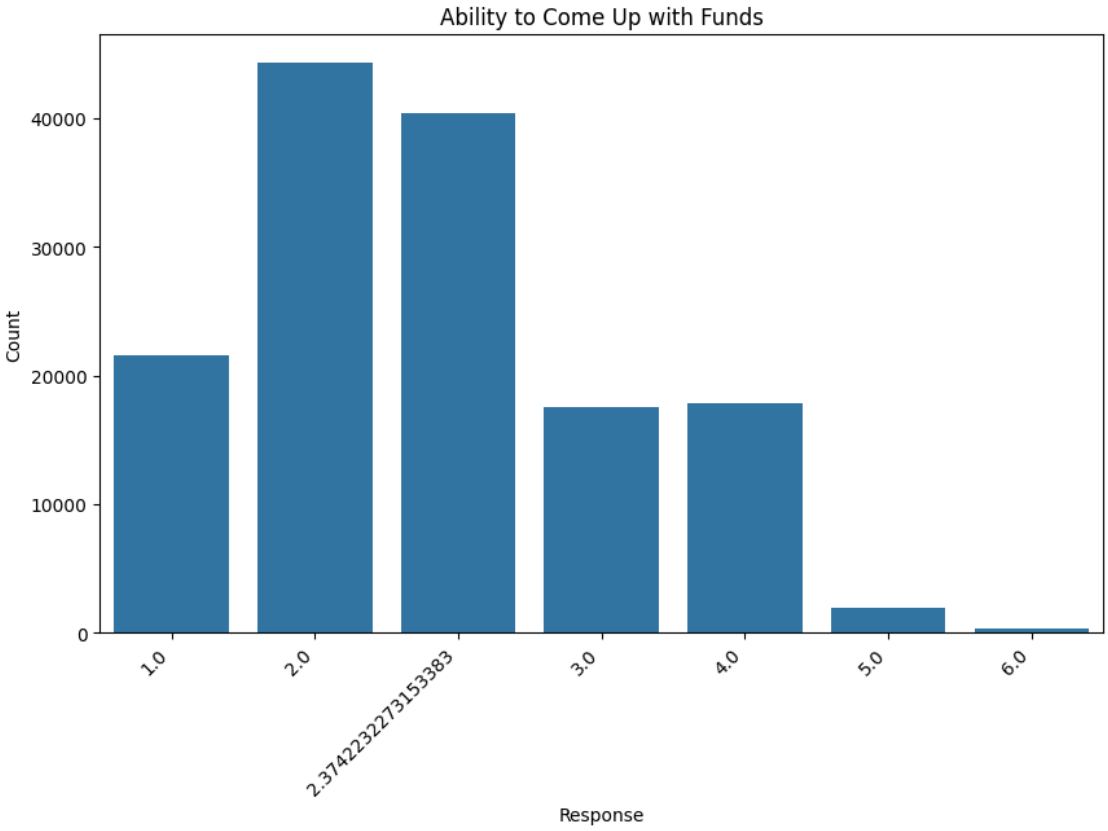
[20]:
```
plt.figure(figsize=(8, 6))
sns.countplot(x='saved', data=dff, hue='female')
plt.title('Distribution of Savings Behaviour')
plt.xlabel('Saved (Yes/No) ')
plt.ylabel('Number of individuals')
plt.xticks([0,1],['No', 'Yes'])
plt.show()
```



[21]:
```
plt.figure(figsize=(8, 6))
sns.countplot(x='borrowed', data=dff, hue='female')
plt.title('Distribution of Borrowing Behaviour')
plt.xlabel('Borrowed (Yes/No) ')
plt.ylabel('Number of individuals')
plt.xticks([0,1],['No', 'Yes'])
plt.show()
```



[22]:
```
plt.figure(figsize=(10, 6))
sns.countplot(x='fin45', data=dff)
plt.title('Ability to Come Up with Funds')
plt.xlabel('Response')
plt.ylabel('Count')
plt.xticks(rotation=45, ha='right')
plt.show()
```



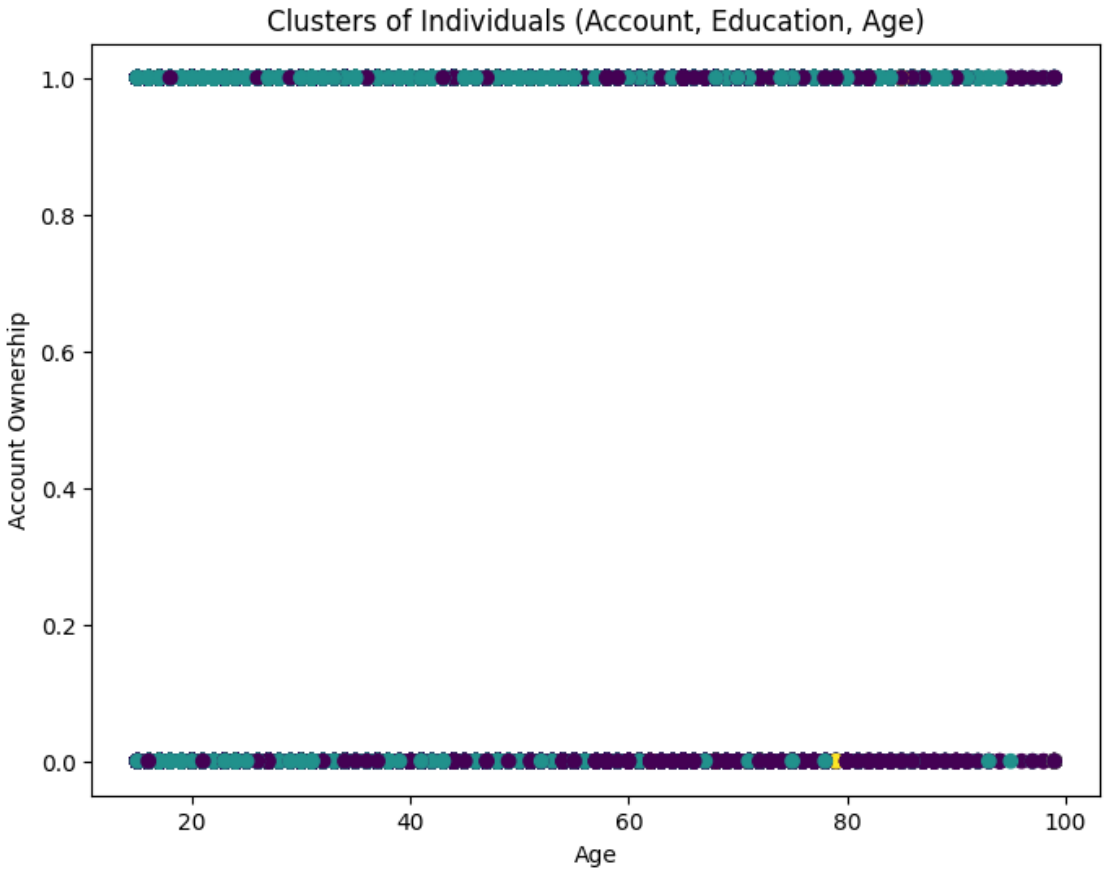
[23]:
```
!pip install scikit-learn

from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import OneHotEncoder
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
Requirement already satisfied: scikit-learn in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (1.6.1)
Requirement already satisfied: numpy>=1.19.5 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
scikit-learn) (2.0.2)
Requirement already satisfied: scipy>=1.6.0 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
scikit-learn) (1.15.1)
Requirement already satisfied: joblib>=1.2.0 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in
```

```
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
scikit-learn) (3.5.0)


[notice] A new release of pip available: 22.3.1 -> 25.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

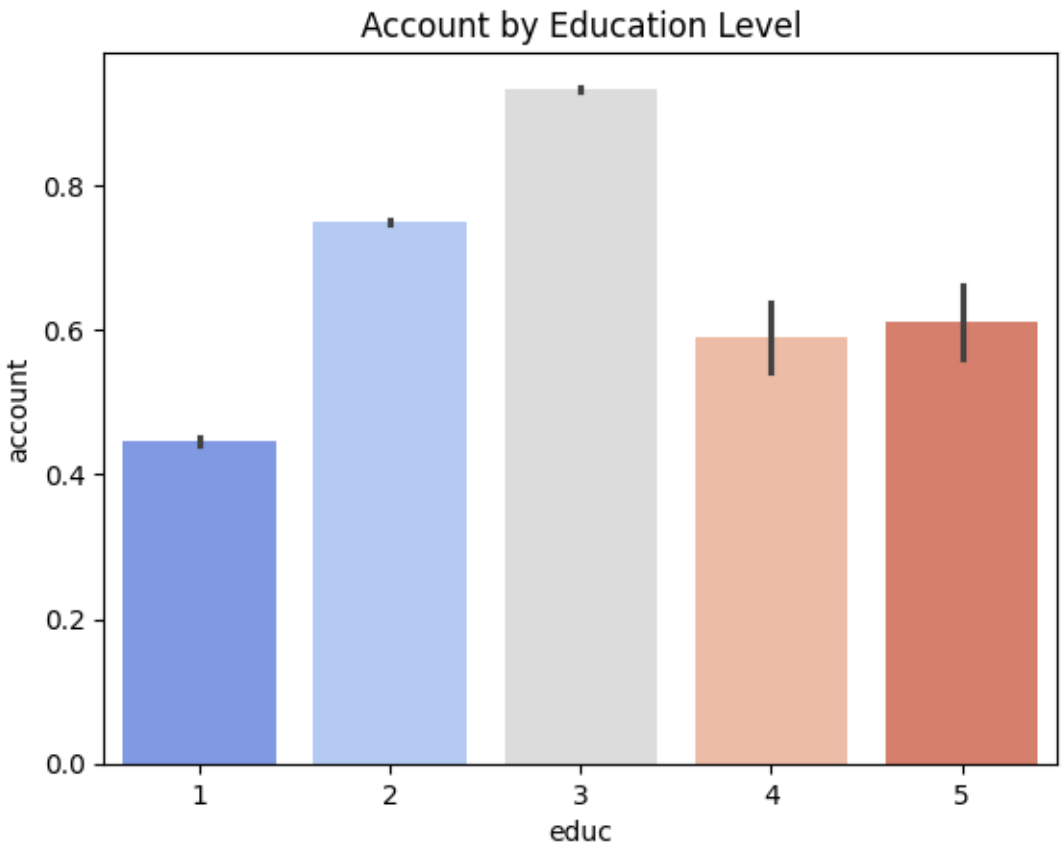
[24]:
```
data = dff
dff = pd.DataFrame(data)

df = dff.copy()

encoder = OneHotEncoder(handle_unknown='ignore', sparse_output=False)
educ_encoded = encoder.fit_transform(df[['educ']])
educ_df = pd.DataFrame(educ_encoded, columns=encoder.
 ↪get_feature_names_out(['educ']))
df = pd.concat([df, educ_df], axis=1)
df.drop('educ', axis=1, inplace=True)

X = df[['account'] + list(educ_df.columns)]


scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)


kmeans = KMeans(n_clusters=3, random_state=42, n_init="auto")
df['cluster'] = kmeans.fit_predict(X_scaled)


print(df.groupby('cluster')[['account'] + list(educ_df.columns)].mean())


plt.figure(figsize=(8, 6))
plt.scatter(df['age'], df['account'], c=df['cluster'], cmap='viridis')
plt.title('Clusters of Individuals (Account, Education, Age)')
plt.xlabel('Age')
plt.ylabel('Account Ownership')
plt.show()
```

```
          account  educ_1    educ_2    educ_3  educ_4    educ_5
cluster
0        0.446330     1.0  0.000000  0.000000     0.0  0.000000
1        0.805584     0.0  0.690164  0.306404     0.0  0.003432
2        0.589873     0.0  0.000000  0.000000     1.0  0.000000
```

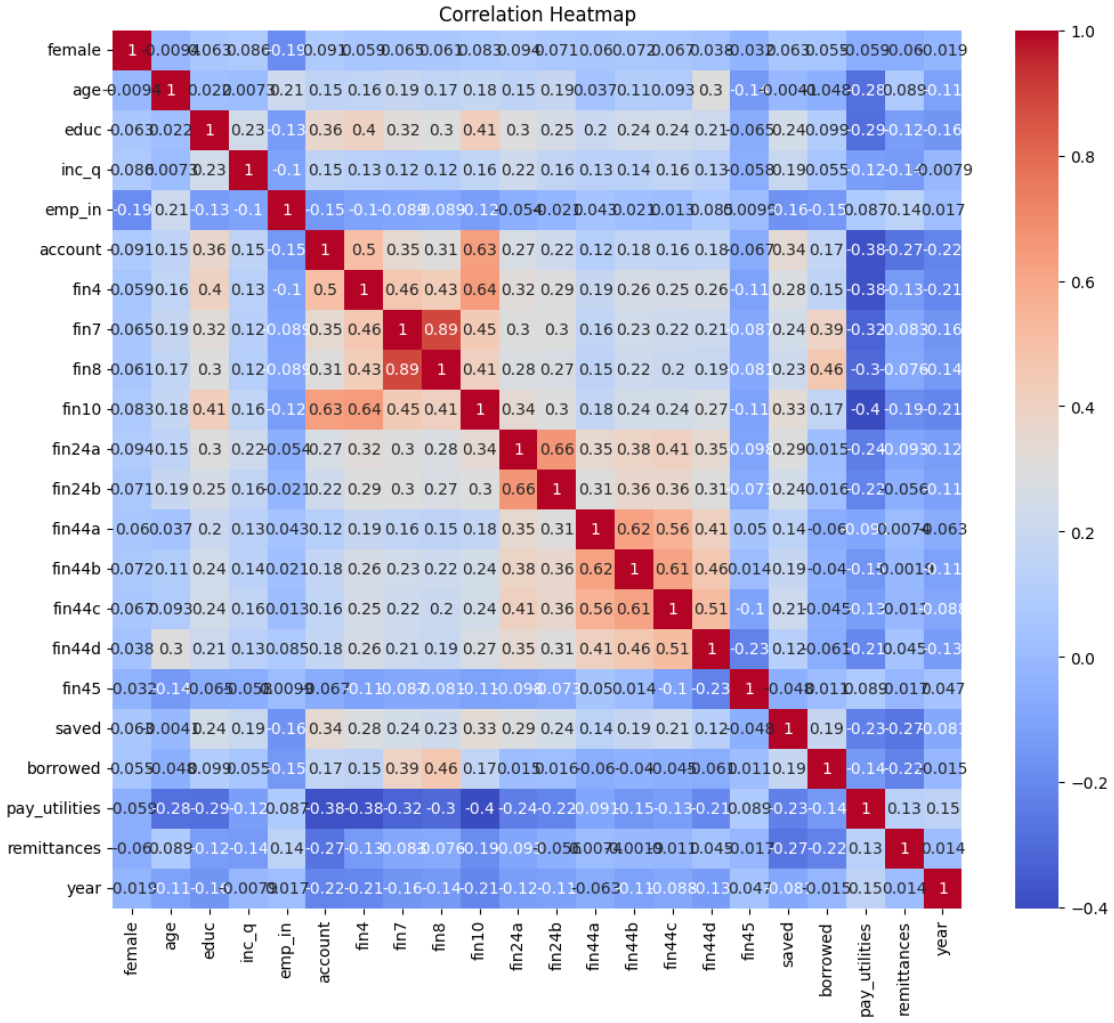


[25]:
```
# Additional visualization (Bar plot to show account by education level)
sns.barplot(x=dff['educ'], y=dff['account'], palette="coolwarm" )
plt.title('Account by Education Level')
plt.show()
```

```
C:\Users\GIM\AppData\Local\Temp\ipykernel_10320\234564392.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=dff['educ'], y=dff['account'], palette="coolwarm" )
```



[26]:
```
plt.figure(figsize=(12, 10))
corr_matrix = dff.corr(numeric_only=True)  # Calculate correlation matrix␣
↪(numeric columns only)
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```



[27]:
```
import pandas as pd
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, accuracy_score

# Step 1: Load the Data and Inspect
data = dff
# Step 2: Feature Engineering
# 2.1: Age
bins = [15, 25, 35, 45, 55, 65, 100]
labels = ['15-24', '25-34', '35-44', '45-54', '55-64', '65+']
data['age_group'] = pd.cut(data['age'], bins=bins, labels=labels, right=False)
data['age_group'] = data['age_group'].cat.codes

# 2.2: Female
```

```
dff['female'] = dff['female'].map({0: 0, 1: 1})

# 2.3: Education
education_map = {
    1: 'primary',
    2: 'secondary',
    3: 'higher'
}
data['educ_level'] = data['educ'].map(education_map)
data['educ_level'] = data['educ_level'].astype('category').cat.codes

# 2.4: Income Quintile
data['inc_q'] = data['inc_q'].astype('category').cat.codes

# 2.5: Binary Indicators
binary_cols = ['account', 'fin4', 'fin7', 'fin8', 'fin10', 'saved', 'borrowed',␣
 ↪'pay_utilities', 'remittances']
for col in binary_cols:
    data[col] = data[col].map({0: 0, 1: 1})

# 2.6: Level of Worry

data['fin44a'] = data['fin44a'] / data['fin44a'].max()

# 2.7: Ability to Come Up with Funds
data['fin45'] = data['fin45'].astype('category').cat.codes

# 2.8: Year
data['year'] = data['year'].astype('category').cat.codes

data['fin45'] = data['fin45'].astype('category').cat.codes

# Define financial distress score
data['financial_distress_score'] = (
    data['borrowed'].map({0: 0, 1: 1}).fillna(0) + # Fill NaN with 0
    data['saved'].map({0: 1, 1: 0}).fillna(0) + # Fill NaN with 0
    (data['fin45'] == 0).astype(int).fillna(0) # Fill NaN with 0
)

# Define financial distress based on the score
threshold = 2
data['financial_distress'] = (data['financial_distress_score'] >= threshold).
 ↪astype(int)

# Display value counts for the target variable
print(data['financial_distress'].value_counts())
```

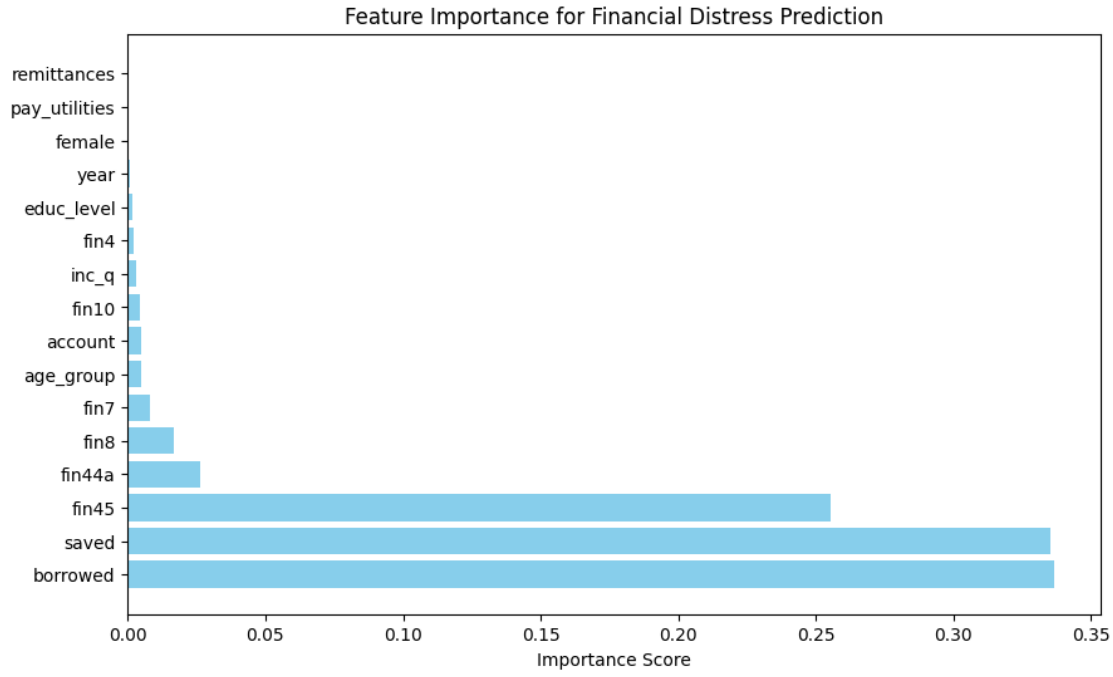
```
# Display the first few rows with the new target variable
print(data[['borrowed', 'saved', 'fin45', 'financial_distress_score',␣
 ↪'financial_distress']].head())

# Now, 'financial_distress' can be used as your target variable

# Step 3: Prepare the Data for Modeling
X = data[[
    'age_group', 'female', 'educ_level', 'inc_q', 'account',
    'fin4', 'fin7', 'fin8', 'fin10', 'saved', 'borrowed',
    'pay_utilities', 'remittances', 'fin44a', 'fin45', 'year'
]]
y = data['financial_distress'] # Replace 'financial_distress' with your target␣
 ↪column

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

# Step 4: Train a Machine Learning Model (Random Forest)
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# Step 5: Evaluate the Model
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))

# Step 6: Feature Importance
feature_importances = model.feature_importances_
feature_importance_df = pd.DataFrame({
    'Feature': X.columns,
    'Importance': feature_importances
})
feature_importance_df = feature_importance_df.sort_values(by='Importance',␣
 ↪ascending=False)
print("\nFeature Importances:\n", feature_importance_df)

# Step 7: Hyperparameter Tuning (Optional)
param_grid = {
    'n_estimators': [100, 200],
    'max_depth': [5, 10],
    'min_samples_split': [2, 5]
}
grid_search = GridSearchCV(estimator=RandomForestClassifier(random_state=42),
                           param_grid=param_grid,
                           cv=3,
                           scoring='accuracy')
```

```
grid_search.fit(X_train, y_train)

print("\nBest parameters:", grid_search.best_params_)
best_model = grid_search.best_estimator_
y_pred = best_model.predict(X_test)
print("Tuned Model Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
financial_distress
0    102905
1     40982
Name: count, dtype: int64
   borrowed  saved  fin45  financial_distress_score  financial_distress
0         1      0      0                         3                   1
1         1      0      3                         2                   1
2         1      0      4                         2                   1
3         0      0      3                         1                   0
4         1      0      0                         3                   1
Accuracy: 1.0
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     20538
           1       1.00      1.00      1.00      8240

    accuracy                           1.00     28778
   macro avg       1.00      1.00      1.00     28778
weighted avg       1.00      1.00      1.00     28778


Feature Importances:
           Feature  Importance
10       borrowed    0.336767
9           saved    0.335318
14          fin45    0.255473
13         fin44a    0.026294
7            fin8    0.016839
6            fin7    0.007888
0       age_group    0.004781
4         account    0.004710
8           fin10    0.004312
3           inc_q    0.003121
5            fin4    0.002101
2      educ_level    0.001858
15           year    0.000538
1          female    0.000000
11  pay_utilities    0.000000
12    remittances    0.000000
```

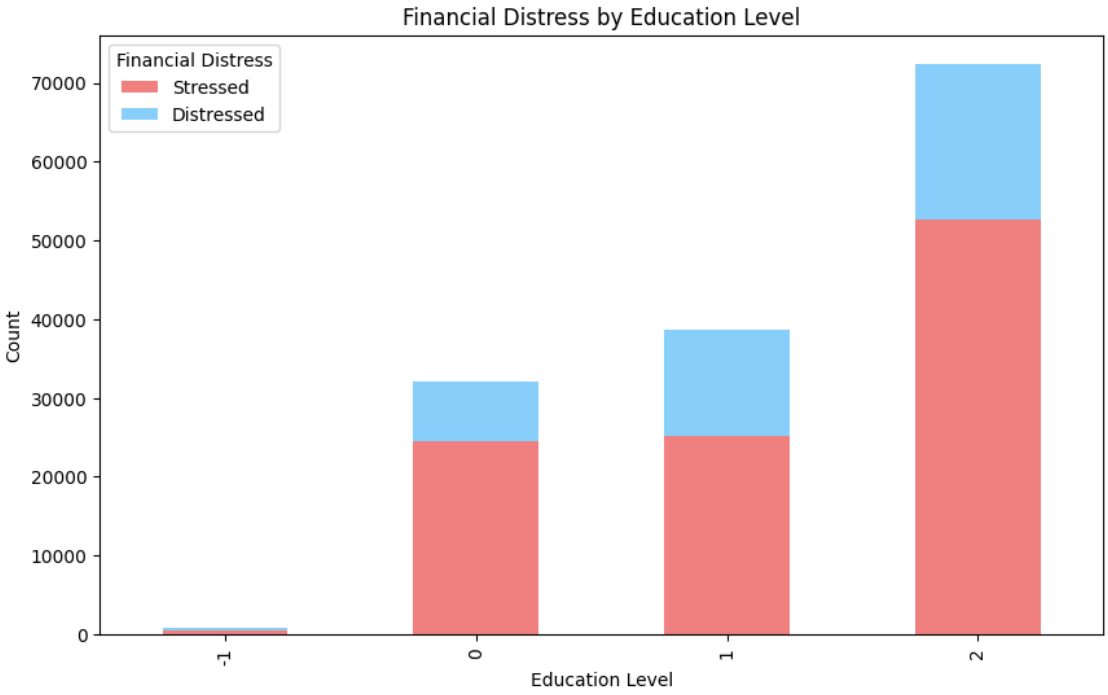
```
Best parameters: {'max_depth': 5, 'min_samples_split': 2, 'n_estimators': 100}
Tuned Model Accuracy: 1.0
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     20538
           1       1.00      1.00      1.00      8240

    accuracy                           1.00     28778
   macro avg       1.00      1.00      1.00     28778
weighted avg       1.00      1.00      1.00     28778
```

[28]:
```
importance_df = feature_importance_df


# Plotting
plt.figure(figsize=(10, 6))
plt.barh(importance_df['Feature'], importance_df['Importance'], color='skyblue')
plt.xlabel('Importance Score')
plt.title('Feature Importance for Financial Distress Prediction')
plt.show()
```

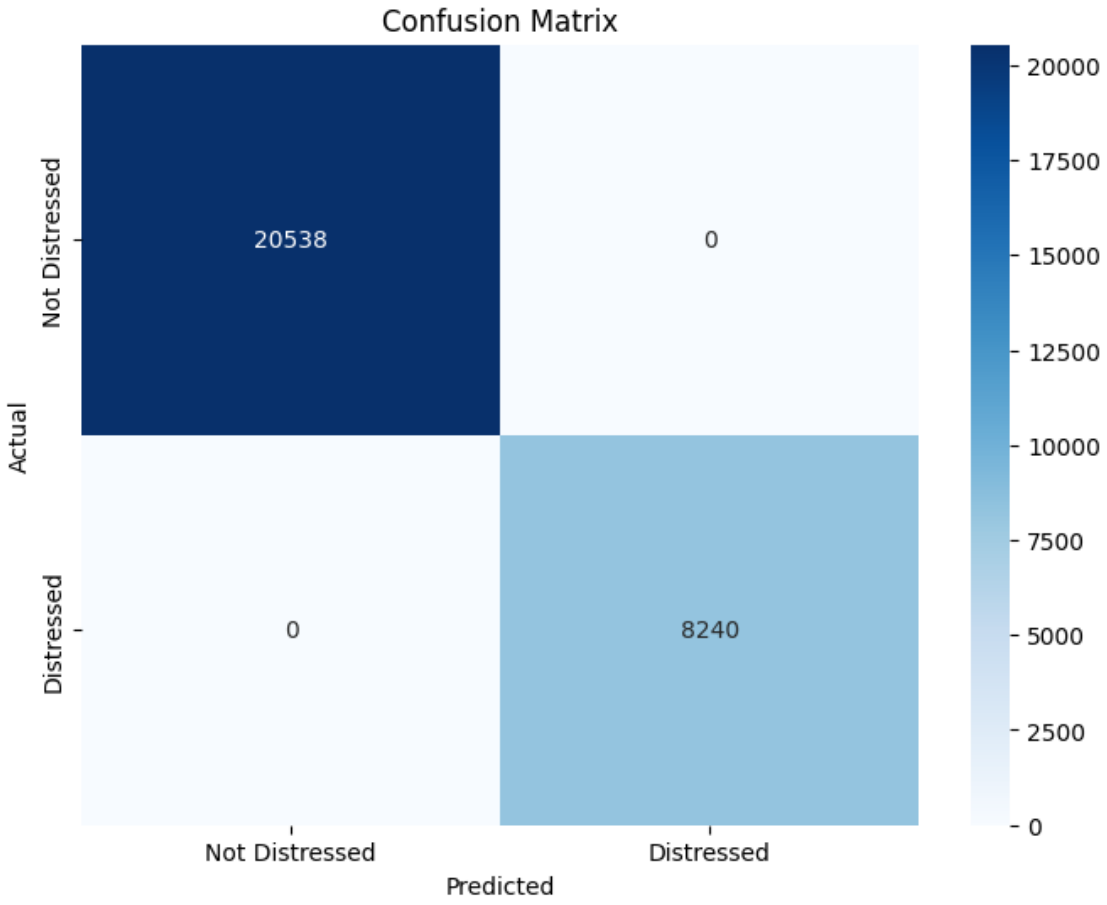


[29]:
```
#making another variable combining education level and distress
category_counts = data.groupby(['educ_level', 'financial_distress']).size().
 ↪unstack()
```

```
category_counts.plot(kind='bar', stacked=True, figsize=(10, 6),␣
 ↪color=['lightcoral', 'lightskyblue'])
plt.title('Financial Distress by Education Level')
plt.xlabel('Education Level')
plt.ylabel('Count')
plt.legend(title='Financial Distress', labels=['Stressed', 'Distressed'])
plt.show()
```



[30]:
```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt="d", cmap='Blues', xticklabels=['Not␣
 ↪Distressed', 'Distressed'], yticklabels=['Not Distressed', 'Distressed'])
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.title('Confusion Matrix')
plt.show()
```



[31]:
```
!pip install keras
!pip install tensorflow
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from keras.models import Sequential
from keras.layers import Dense
from sklearn.metrics import classification_report, accuracy_score
```

```
Requirement already satisfied: keras in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (3.8.0)
Requirement already satisfied: absl-py in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
keras) (2.1.0)
Requirement already satisfied: numpy in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
keras) (2.0.2)
Requirement already satisfied: rich in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
```

```
keras) (13.9.4)
Requirement already satisfied: namex in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
keras) (0.0.8)
Requirement already satisfied: h5py in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
keras) (3.12.1)
Requirement already satisfied: optree in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
keras) (0.14.0)
Requirement already satisfied: ml-dtypes in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
keras) (0.4.1)
Requirement already satisfied: packaging in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
keras) (24.2)
Requirement already satisfied: typing-extensions>=4.5.0 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
optree->keras) (4.12.2)
Requirement already satisfied: markdown-it-py>=2.2.0 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
rich->keras) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
rich->keras) (2.19.1)
Requirement already satisfied: mdurl~=0.1 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
markdown-it-py>=2.2.0->rich->keras) (0.1.2)


[notice] A new release of pip available: 22.3.1 -> 25.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip

Requirement already satisfied: tensorflow in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (2.18.0)
Requirement already satisfied: tensorflow-intel==2.18.0 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
tensorflow) (2.18.0)
Requirement already satisfied: absl-py>=1.0.0 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
tensorflow-intel==2.18.0->tensorflow) (2.1.0)
Requirement already satisfied: astunparse>=1.6.0 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
tensorflow-intel==2.18.0->tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=24.3.25 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
tensorflow-intel==2.18.0->tensorflow) (25.2.10)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
```

```
tensorflow-intel==2.18.0->tensorflow) (0.6.0)
Requirement already satisfied: google-pasta>=0.1.1 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
tensorflow-intel==2.18.0->tensorflow) (0.2.0)
Requirement already satisfied: libclang>=13.0.0 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
tensorflow-intel==2.18.0->tensorflow) (18.1.1)
Requirement already satisfied: opt-einsum>=2.3.2 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
tensorflow-intel==2.18.0->tensorflow) (3.4.0)
Requirement already satisfied: packaging in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
tensorflow-intel==2.18.0->tensorflow) (24.2)
Requirement already satisfied:
protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<6.0.0dev,>=3.20.3
in c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
tensorflow-intel==2.18.0->tensorflow) (5.29.3)
Requirement already satisfied: requests<3,>=2.21.0 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
tensorflow-intel==2.18.0->tensorflow) (2.32.3)
Requirement already satisfied: setuptools in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
tensorflow-intel==2.18.0->tensorflow) (65.5.0)
Requirement already satisfied: six>=1.12.0 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
tensorflow-intel==2.18.0->tensorflow) (1.17.0)
Requirement already satisfied: termcolor>=1.1.0 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
tensorflow-intel==2.18.0->tensorflow) (2.5.0)
Requirement already satisfied: typing-extensions>=3.6.6 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
tensorflow-intel==2.18.0->tensorflow) (4.12.2)
Requirement already satisfied: wrapt>=1.11.0 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
tensorflow-intel==2.18.0->tensorflow) (1.17.2)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
tensorflow-intel==2.18.0->tensorflow) (1.70.0)
Requirement already satisfied: tensorboard<2.19,>=2.18 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
tensorflow-intel==2.18.0->tensorflow) (2.18.0)
Requirement already satisfied: keras>=3.5.0 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
tensorflow-intel==2.18.0->tensorflow) (3.8.0)
Requirement already satisfied: numpy<2.1.0,>=1.26.0 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
tensorflow-intel==2.18.0->tensorflow) (2.0.2)
Requirement already satisfied: h5py>=3.11.0 in
```

```
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
tensorflow-intel==2.18.0->tensorflow) (3.12.1)
Requirement already satisfied: ml-dtypes<0.5.0,>=0.4.0 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
tensorflow-intel==2.18.0->tensorflow) (0.4.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
tensorflow-intel==2.18.0->tensorflow) (0.31.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
astunparse>=1.6.0->tensorflow-intel==2.18.0->tensorflow) (0.45.1)
Requirement already satisfied: rich in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
keras>=3.5.0->tensorflow-intel==2.18.0->tensorflow) (13.9.4)
Requirement already satisfied: namex in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
keras>=3.5.0->tensorflow-intel==2.18.0->tensorflow) (0.0.8)
Requirement already satisfied: optree in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
keras>=3.5.0->tensorflow-intel==2.18.0->tensorflow) (0.14.0)
Requirement already satisfied: charset-normalizer<4,>=2 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
requests<3,>=2.21.0->tensorflow-intel==2.18.0->tensorflow) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
requests<3,>=2.21.0->tensorflow-intel==2.18.0->tensorflow) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
requests<3,>=2.21.0->tensorflow-intel==2.18.0->tensorflow) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
requests<3,>=2.21.0->tensorflow-intel==2.18.0->tensorflow) (2025.1.31)
Requirement already satisfied: markdown>=2.6.8 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
tensorboard<2.19,>=2.18->tensorflow-intel==2.18.0->tensorflow) (3.7)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
tensorboard<2.19,>=2.18->tensorflow-intel==2.18.0->tensorflow) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
tensorboard<2.19,>=2.18->tensorflow-intel==2.18.0->tensorflow) (3.1.3)
Requirement already satisfied: MarkupSafe>=2.1.1 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
werkzeug>=1.0.1->tensorboard<2.19,>=2.18->tensorflow-intel==2.18.0->tensorflow)
(3.0.2)
Requirement already satisfied: markdown-it-py>=2.2.0 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
rich->keras>=3.5.0->tensorflow-intel==2.18.0->tensorflow) (3.0.0)
```

```
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
rich->keras>=3.5.0->tensorflow-intel==2.18.0->tensorflow) (2.19.1)
Requirement already satisfied: mdurl~=0.1 in
c:\users\gim\appdata\local\programs\python\python311\lib\site-packages (from
markdown-it-py>=2.2.0->rich->keras>=3.5.0->tensorflow-intel==2.18.0->tensorflow)
(0.1.2)


[notice] A new release of pip available: 22.3.1 -> 25.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

[32]:
```
#Splitting data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

model = Sequential()
model.add(Dense(10, activation='relu', input_shape=(X_train.shape[1],)))
model.add(Dense(10, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
```

```
C:\Users\GIM\AppData\Local\Programs\Python\Python311\Lib\site-
packages\keras\src\layers\core\dense.py:87: UserWarning: Do not pass an
`input_shape`/`input_dim` argument to a layer. When using Sequential models,
prefer using an `Input(shape)` object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

[33]:
```
model.compile(optimizer='adam', loss='binary_crossentropy',␣
 ↪metrics=['accuracy'])


model.fit(X_train, y_train, epochs=25, batch_size=10, verbose=1)


y_pred = (model.predict(X_test) > 0.5).astype("int32")

print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
Epoch 1/25
11511/11511             15s 1ms/step
- accuracy: 0.7141 - loss: 0.6122
Epoch 2/25
11511/11511             14s 1ms/step
- accuracy: 0.7171 - loss: 0.5957
```

```
Epoch 3/25
11511/11511             13s 1ms/step
- accuracy: 0.7132 - loss: 0.5993
Epoch 4/25
11511/11511             13s 1ms/step
- accuracy: 0.7160 - loss: 0.5968
Epoch 5/25
11511/11511             13s 1ms/step
- accuracy: 0.7177 - loss: 0.5951
Epoch 6/25
11511/11511             13s 1ms/step
- accuracy: 0.7159 - loss: 0.5969
Epoch 7/25
11511/11511             13s 1ms/step
- accuracy: 0.7166 - loss: 0.5962
Epoch 8/25
11511/11511             13s 1ms/step
- accuracy: 0.7147 - loss: 0.5979
Epoch 9/25
11511/11511             13s 1ms/step
- accuracy: 0.7165 - loss: 0.5963
Epoch 10/25
11511/11511             13s 1ms/step
- accuracy: 0.7144 - loss: 0.5982
Epoch 11/25
11511/11511             13s 1ms/step
- accuracy: 0.7159 - loss: 0.5968
Epoch 12/25
11511/11511             13s 1ms/step
- accuracy: 0.7125 - loss: 0.5999
Epoch 13/25
11511/11511             13s 1ms/step
- accuracy: 0.7150 - loss: 0.5977
Epoch 14/25
11511/11511             13s 1ms/step
- accuracy: 0.7160 - loss: 0.5967
Epoch 15/25
11511/11511             13s 1ms/step
- accuracy: 0.7163 - loss: 0.5965
Epoch 16/25
11511/11511             13s 1ms/step
- accuracy: 0.7143 - loss: 0.5983
Epoch 17/25
11511/11511             13s 1ms/step
- accuracy: 0.7157 - loss: 0.5970
Epoch 18/25
11511/11511             13s 1ms/step
- accuracy: 0.7125 - loss: 0.5999
```

```
Epoch 19/25
11511/11511             13s 1ms/step
- accuracy: 0.7139 - loss: 0.5987
Epoch 20/25
11511/11511             13s 1ms/step
- accuracy: 0.7176 - loss: 0.5952
Epoch 21/25
11511/11511             13s 1ms/step
- accuracy: 0.7155 - loss: 0.5972
Epoch 22/25
11511/11511             13s 1ms/step
- accuracy: 0.7142 - loss: 0.5984
Epoch 23/25
11511/11511             13s 1ms/step
- accuracy: 0.7136 - loss: 0.5989
Epoch 24/25
11511/11511             13s 1ms/step
- accuracy: 0.7174 - loss: 0.5954
Epoch 25/25
11511/11511             13s 1ms/step
- accuracy: 0.7136 - loss: 0.5990
900/900           1s 690us/step
Accuracy: 0.7136701647091528
              precision    recall  f1-score   support

           0       0.71      1.00      0.83     20538
           1       0.00      0.00      0.00      8240

    accuracy                           0.71     28778
   macro avg       0.36      0.50      0.42     28778
weighted avg       0.51      0.71      0.59     28778

C:\Users\GIM\AppData\Local\Programs\Python\Python311\Lib\site-
packages\sklearn\metrics\_classification.py:1565: UndefinedMetricWarning:
Precision is ill-defined and being set to 0.0 in labels with no predicted
samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\GIM\AppData\Local\Programs\Python\Python311\Lib\site-
packages\sklearn\metrics\_classification.py:1565: UndefinedMetricWarning:
Precision is ill-defined and being set to 0.0 in labels with no predicted
samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\GIM\AppData\Local\Programs\Python\Python311\Lib\site-
packages\sklearn\metrics\_classification.py:1565: UndefinedMetricWarning:
Precision is ill-defined and being set to 0.0 in labels with no predicted
samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

[34]:

```
y_prob=model.predict(X_test)
y_pred=(y_prob > 0.5).astype("int32")

risk_scores=y_prob.flatten()

print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
900/900            1s 551us/step
Accuracy: 0.7136701647091528
              precision    recall  f1-score   support

           0       0.71      1.00      0.83     20538
           1       0.00      0.00      0.00      8240

    accuracy                           0.71     28778
   macro avg       0.36      0.50      0.42     28778
weighted avg       0.51      0.71      0.59     28778

C:\Users\GIM\AppData\Local\Programs\Python\Python311\Lib\site-
packages\sklearn\metrics\_classification.py:1565: UndefinedMetricWarning:
Precision is ill-defined and being set to 0.0 in labels with no predicted
samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\GIM\AppData\Local\Programs\Python\Python311\Lib\site-
packages\sklearn\metrics\_classification.py:1565: UndefinedMetricWarning:
Precision is ill-defined and being set to 0.0 in labels with no predicted
samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\GIM\AppData\Local\Programs\Python\Python311\Lib\site-
packages\sklearn\metrics\_classification.py:1565: UndefinedMetricWarning:
Precision is ill-defined and being set to 0.0 in labels with no predicted
samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

[35]:

```
dff.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 143887 entries, 0 to 143886
Data columns (total 26 columns):
 #   Column                   Non-Null Count   Dtype
---  ------                   --------------   -----
 0   female                   76585 non-null   float64
 1   age                      143887 non-null  float64
 2   educ                     143887 non-null  int64
 3   inc_q                    143887 non-null  int8
 4   emp_in                   143887 non-null  float64
 5   account                  143887 non-null  int64
```

```
 6   fin4                        143887 non-null  int64
 7   fin7                        143887 non-null  int64
 8   fin8                        143887 non-null  int64
 9   fin10                       143887 non-null  int64
 10  fin24a                      143887 non-null  float64
 11  fin24b                      143887 non-null  float64
 12  fin44a                      143887 non-null  float64
 13  fin44b                      143887 non-null  int64
 14  fin44c                      143887 non-null  int64
 15  fin44d                      143887 non-null  int64
 16  fin45                       143887 non-null  int8
 17  saved                       143887 non-null  int64
 18  borrowed                    143887 non-null  int64
 19  pay_utilities               46841 non-null   float64
 20  remittances                 27621 non-null   float64
 21  year                        143887 non-null  int8
 22  age_group                   143887 non-null  int8
 23  educ_level                  143887 non-null  int8
 24  financial_distress_score    143887 non-null  int64
 25  financial_distress          143887 non-null  int64
dtypes: float64(8), int64(13), int8(5)
memory usage: 23.7 MB
```

[36]:
```
# Define the features used during training
FEATURE_NAMES = [
    'age_group', 'female', 'educ_level', 'inc_q', 'account',
    'fin4', 'fin7', 'fin8', 'fin10', 'saved', 'borrowed',
    'pay_utilities', 'remittances', 'fin44a', 'fin45', 'year'
]

# Test values for low-risk individual
test_values_low_risk = {
    'age_group': 2.0, 'female': 0.0, 'educ_level': 2.0, 'inc_q': 4.0, 'account':
 ↪ 1.0,
    'fin4': 1.0, 'fin7': 1.0, 'fin8': 1.0, 'fin10': 1.0, 'saved': 1.0,␣
 ↪'borrowed': 0.0,
    'pay_utilities': 1.0, 'remittances': 0.0, 'fin44a': 0.2, 'fin45': 4.0,␣
 ↪'year': 1.0
}

# Test values for moderate-risk individual
test_values_moderate_risk = {
    'age_group': 1.0, 'female': 1.0, 'educ_level': 1.0, 'inc_q': 2.0, 'account':
 ↪ 1.0,
    'fin4': 0.0, 'fin7': 0.0, 'fin8': 0.0, 'fin10': 0.0, 'saved': 0.0,␣
 ↪'borrowed': 1.0,
```

```
    'pay_utilities': 1.0, 'remittances': 0.0, 'fin44a': 0.6, 'fin45': 2.0,␣
 ↪'year': 1.0
}

# Test values for high-risk individual
test_values_high_risk = {
    'age_group': 0.0, 'female': 1.0, 'educ_level': 0.0, 'inc_q': 0.0, 'account':
 ↪ 0.0,
    'fin4': 0.0, 'fin7': 0.0, 'fin8': 0.0, 'fin10': 0.0, 'saved': 0.0,␣
 ↪'borrowed': 1.0,
    'pay_utilities': 0.0, 'remittances': 1.0, 'fin44a': 0.9, 'fin45': 0.0,␣
 ↪'year': 0.0
}



# Function to make predictions
def predict_financial_distress(test_data, model, scaler, feature_names):
    # Convert dictionary to ordered NumPy array
    test_array = np.array([test_data[feature] for feature in feature_names]).
 ↪reshape(1, -1)

    print("\nOriginal Input Data:", test_data)
    print("Ordered Feature Array:", test_array)

    # Scale the input
    test_array_scaled = scaler.transform(test_array)

    print("Scaled Input Data:", test_array_scaled)

    # Make prediction using the trained model
    prediction = model.predict(test_array_scaled)

    print("Raw Model Prediction:", prediction)

    return prediction

# Set the data types of the input
test_values_low_risk = {k: np.float64(v) for k, v in test_values_low_risk.
 ↪items()}
test_values_moderate_risk = {k: np.float64(v) for k, v in␣
 ↪test_values_moderate_risk.items()}
test_values_high_risk = {k: np.float64(v) for k, v in test_values_high_risk.
 ↪items()}

# Make predictions for each risk level
```

```
prediction_low = predict_financial_distress(test_values_low_risk, model,␣
 ↪scaler, FEATURE_NAMES)
prediction_moderate = predict_financial_distress(test_values_moderate_risk,␣
 ↪model, scaler, FEATURE_NAMES)
prediction_high = predict_financial_distress(test_values_high_risk, model,␣
 ↪scaler, FEATURE_NAMES)

print("Prediction for low-risk individual:", prediction_low)
print("Prediction for moderate-risk individual:", prediction_moderate)
print("Prediction for high-risk individual:", prediction_high)
```

```
Original Input Data: {'age_group': np.float64(2.0), 'female': np.float64(0.0),
'educ_level': np.float64(2.0), 'inc_q': np.float64(4.0), 'account':
np.float64(1.0), 'fin4': np.float64(1.0), 'fin7': np.float64(1.0), 'fin8':
np.float64(1.0), 'fin10': np.float64(1.0), 'saved': np.float64(1.0), 'borrowed':
np.float64(0.0), 'pay_utilities': np.float64(1.0), 'remittances':
np.float64(0.0), 'fin44a': np.float64(0.2), 'fin45': np.float64(4.0), 'year':
np.float64(1.0)}
Ordered Feature Array: [[2.  0.  2.  4.  1.  1.  1.  1.  1.  1.  0.  1.  0.  0.2
4.  1. ]]
Scaled Input Data: [[-0.08352539 -1.          0.88747001  1.24325012  0.64135898
1.28752914
   1.84890031  2.08029817  1.01205258  0.92147211 -1.05083485  0.
  -1.         -0.86378526  1.70617856  2.82990702]]
1/1             0s 26ms/step
Raw Model Prediction: [[0.2828166]]

Original Input Data: {'age_group': np.float64(1.0), 'female': np.float64(1.0),
'educ_level': np.float64(1.0), 'inc_q': np.float64(2.0), 'account':
np.float64(1.0), 'fin4': np.float64(0.0), 'fin7': np.float64(0.0), 'fin8':
np.float64(0.0), 'fin10': np.float64(0.0), 'saved': np.float64(0.0), 'borrowed':
np.float64(1.0), 'pay_utilities': np.float64(1.0), 'remittances':
np.float64(0.0), 'fin44a': np.float64(0.6), 'fin45': np.float64(2.0), 'year':
np.float64(1.0)}
Ordered Feature Array: [[1.  1.  1.  2.  1.  0.  0.  0.  0.  0.  1.  1.  0.  0.6
2.  1. ]]
Scaled Input Data: [[-0.69234782  0.         -0.33099117 -0.16565244  0.64135898
-0.77668145
  -0.54086204 -0.48070032 -0.98809096 -1.08522003  0.95162432  0.
  -1.          1.7846301   0.14729053  2.82990702]]
1/1             0s 31ms/step
Raw Model Prediction: [[0.2828166]]

Original Input Data: {'age_group': np.float64(0.0), 'female': np.float64(1.0),
'educ_level': np.float64(0.0), 'inc_q': np.float64(0.0), 'account':
np.float64(0.0), 'fin4': np.float64(0.0), 'fin7': np.float64(0.0), 'fin8':
```

```
np.float64(0.0), 'fin10': np.float64(0.0), 'saved': np.float64(0.0), 'borrowed':
np.float64(1.0), 'pay_utilities': np.float64(0.0), 'remittances':
np.float64(1.0), 'fin44a': np.float64(0.9), 'fin45': np.float64(0.0), 'year':
np.float64(0.0)}
Ordered Feature Array: [[0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  1.  0.9
0.  0. ]]
Scaled Input Data: [[-1.30117025  0.         -1.54945236 -1.574555   -1.5591892
-0.77668145
  -0.54086204 -0.48070032 -0.98809096 -1.08522003  0.95162432 -1.
   0.          3.77094161 -1.41159749 -0.3533685 ]]

C:\Users\GIM\AppData\Local\Programs\Python\Python311\Lib\site-
packages\sklearn\utils\validation.py:2739: UserWarning: X does not have valid
feature names, but StandardScaler was fitted with feature names
  warnings.warn(
C:\Users\GIM\AppData\Local\Programs\Python\Python311\Lib\site-
packages\sklearn\utils\validation.py:2739: UserWarning: X does not have valid
feature names, but StandardScaler was fitted with feature names
  warnings.warn(
C:\Users\GIM\AppData\Local\Programs\Python\Python311\Lib\site-
packages\sklearn\utils\validation.py:2739: UserWarning: X does not have valid
feature names, but StandardScaler was fitted with feature names
  warnings.warn(

1/1             0s 43ms/step
Raw Model Prediction: [[0.2828166]]
Prediction for low-risk individual: [[0.2828166]]
Prediction for moderate-risk individual: [[0.2828166]]
Prediction for high-risk individual: [[0.2828166]]
```

[ ]: