

The image shows a Jupyter Notebook interface within a VS Code editor. The notebook is titled 'credit_scoring_segmentation.ipynb'. It contains three code cells:

- Cell [1]: `import pandas as pd` (Execution time: 1.1s)
- Cell [2]: `df = pd.read_csv(r'/Users/subirams/Desktop/VSCODE/credit_scoring.csv')` (Execution time: 0.0s)
- Cell [3]: `df` (Execution time: 0.0s)

The output of Cell [3] displays a DataFrame with 1000 rows and 12 columns. The columns are: Age, Gender, Marital Status, Education Level, Employment Status, Credit Utilization Ratio, Payment History, Number of Credit Accounts, Loan Amount, Interest Rate, Loan Term, and Type of Loan. The first few rows of the DataFrame are shown below:

	Age	Gender	Marital Status	Education Level	Employment Status	Credit Utilization Ratio	Payment History	Number of Credit Accounts	Loan Amount	Interest Rate	Loan Term	Type of Loan
0	60	Male	Married	Master	Employed	0.22	2685.0	2	4675000	2.65	48	Personal Loan
1	25	Male	Married	High School	Unemployed	0.20	2371.0	9	3619000	5.19	60	Auto Loan
2	30	Female	Single	Master	Employed	0.22	2771.0	6	957000	2.76	12	Auto Loan
3	58	Female	Married	PhD	Unemployed	0.12	1371.0	2	4731000	6.57	60	Auto Loan
4	32	Male	Married	Bachelor	Self-Employed	0.99	828.0	2	3289000	6.28	36	Personal Loan
...
995	59	Male	Divorced	High School	Employed	0.74	1285.0	8	3530000	12.99	48	Auto Loan
996	64	Male	Divorced	Bachelor	Unemployed	0.77	1857.0	2	1377000	18.02	60	Home Loan
997	63	Female	Single	Master	Self-Employed	0.18	2628.0	10	2443000	18.95	12	Personal Loan
998	51	Female	Married	PhD	Self-Employed	0.32	1142.0	3	1301000	1.80	24	Auto Loan
999	37	Female	Married	Master	Self-Employed	0.17	1028.0	5	4182000	9.34	24	Auto Loan

The bottom of the image shows a macOS dock with various application icons, including Finder, Launchpad, Safari, Messages, Apple TV, Mail, Photos, Videos, Calendar (showing May 29), Contacts, Notes, Music, Podcasts, WhatsApp, Reminders, App Store, System Settings, YouTube, Google Chrome, Microsoft Edge, Microsoft Teams, Microsoft Word, Microsoft PowerPoint, and a Trash can.

← →

Search

Untitled-1.ipynb • credit_scoring_segmentation.ipynb • Release Notes: 1.89.1

Users > subirams > Desktop > VSCODE > credit_scoring_segmentation.ipynb > ...

+ Code + Markdown | ▶ Run All ↺ Restart ≡ Clear All Outputs | Variables Outline ...

base (Python 3.11.5)

df.head(5)

[4] ✓ 0.0s Python

	Age	Gender	Marital Status	Education Level	Employment Status	Credit Utilization Ratio	Payment History	Number of Credit Accounts	Loan Amount	Interest Rate	Loan Term	Type of Loan
0	60	Male	Married	Master	Employed	0.22	2685.0	2	4675000	2.65	48	Personal Loan
1	25	Male	Married	High School	Unemployed	0.20	2371.0	9	3619000	5.19	60	Auto Loan
2	30	Female	Single	Master	Employed	0.22	2771.0	6	957000	2.76	12	Auto Loan
3	58	Female	Married	PhD	Unemployed	0.12	1371.0	2	4731000	6.57	60	Auto Loan
4	32	Male	Married	Bachelor	Self-Employed	0.99	828.0	2	3289000	6.28	36	Personal Loan

df.info()

[5] ✓ 0.0s Python

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 12 columns):

#	Column	Non-Null Count	Dtype
0	Age	1000 non-null	int64
1	Gender	1000 non-null	object
2	Marital Status	1000 non-null	object
3	Education Level	1000 non-null	object
4	Employment Status	1000 non-null	object
5	Credit Utilization Ratio	1000 non-null	float64
6	Payment History	1000 non-null	float64
7	Number of Credit Accounts	1000 non-null	int64
8	Loan Amount	1000 non-null	int64
9	Interest Rate	1000 non-null	float64
10	Loan Term	1000 non-null	int64
11	Type of Loan	1000 non-null	object

dtypes: float64(3), int64(4), object(5)
memory usage: 93.9+ KB

← →

Search

□ □ □ □

Untitled-1.ipynb • credit_scoring_segmentation.ipynb • Release Notes: 1.89.1

Users > subirams > Desktop > VSCODE > credit_scoring_segmentation.ipynb > ...

+ Code + Markdown | ▶ Run All ↺ Restart ≡ Clear All Outputs | Variables Outline ...

base (Python 3.11.5)

9 Interest Rate 1000 non-null float64

10 Loan Term 1000 non-null int64

11 Type of Loan 1000 non-null object

dtypes: float64(3), int64(4), object(5)

memory usage: 93.9+ KB

[6] ✓ 0.0s Python

df.describe()

	Age	Credit Utilization Ratio	Payment History	Number of Credit Accounts	Loan Amount	Interest Rate	Loan Term
count	1000.000000	1000.000000	1000.000000	1000.000000	1.000000e+03	1000.000000	1000.000000
mean	42.702000	0.509950	1452.814000	5.580000	2.471401e+06	10.686600	37.128000
std	13.266771	0.291057	827.934146	2.933634	1.387047e+06	5.479058	17.436274
min	20.000000	0.000000	0.000000	1.000000	1.080000e+05	1.010000	12.000000
25%	31.000000	0.250000	763.750000	3.000000	1.298000e+06	6.022500	24.000000
50%	42.000000	0.530000	1428.000000	6.000000	2.437500e+06	10.705000	36.000000
75%	54.000000	0.750000	2142.000000	8.000000	3.653250e+06	15.440000	48.000000
max	65.000000	1.000000	2857.000000	10.000000	4.996000e+06	19.990000	60.000000

[7] ✓ 0.0s Python

import plotly.graph_objects as go

[8] ✓ 0.4s Python

import plotly.express as px
import plotly.io as pio

[9] ✓ 0.6s Python

credit_utilization_fig = px.box(df, y='Credit Utilization Ratio',
title='Credit Utilization Ratio Distribution')
credit_utilization_fig.show()

Apple Mail

Calendar

Photos

App Store

YouTube

Google

Microsoft Teams

Visual Studio Code

PowerPoint

← →

Search

Untitled-1.ipynb • credit_scoring_segmentation.ipynb • Release Notes: 1.89.1

Users > subiram > Desktop > VSCODE > credit_scoring_segmentation.ipynb > ...

+ Code + Markdown ▶ Run All ↺ Restart ≡ Clear All Outputs | Variables Outline ...

base (Python 3.11.5)

[7] ✓ 0.0s Python

[8] ✓ 0.4s Python

[9] ✓ 0.6s Python

```
import plotly.graph_objects as go

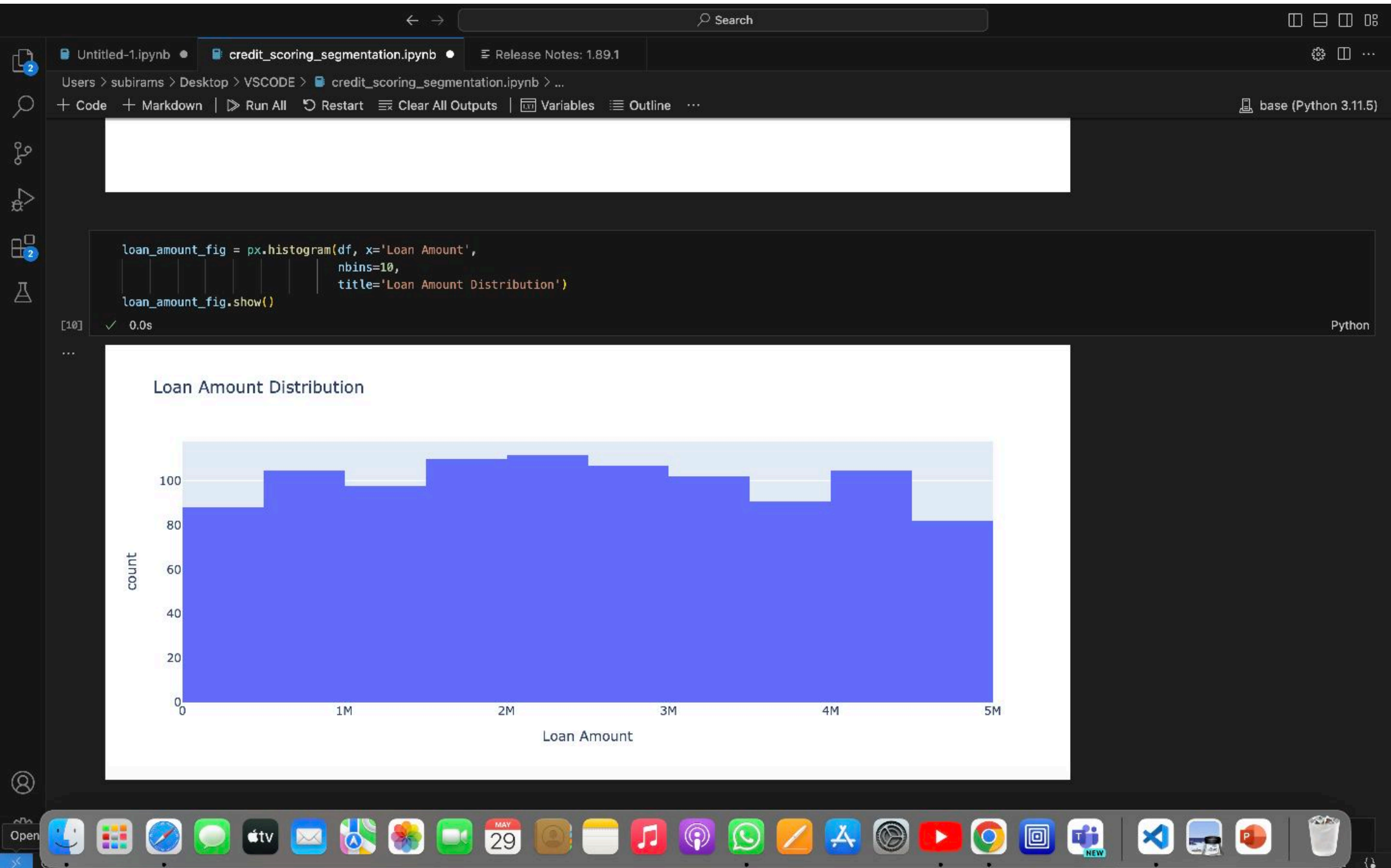
import plotly.express as px
import plotly.io as pio

credit_utilization_fig = px.box(df, y='Credit Utilization Ratio',
                                title='Credit Utilization Ratio Distribution')
credit_utilization_fig.show()
```

Credit Utilization Ratio Distribution

Statistic	Value
Minimum	0.0
First Quartile (Q1)	0.25
Median	0.52
Third Quartile (Q3)	0.75
Maximum	1.0





← →

Search

Untitled-1.ipynb • credit_scoring_segmentation.ipynb • Release Notes: 1.89.1

Users > subirams > Desktop > VSCODE > credit_scoring_segmentation.ipynb > ...

+ Code + Markdown | ▶ Run All ↺ Restart ≡ Clear All Outputs | Variables Outline ...

base (Python 3.11.5)

```
numeric_df = df[['Credit Utilization Ratio',
                 'Payment History',
                 'Number of Credit Accounts',
                 'Loan Amount', 'Interest Rate',
                 'Loan Term']]
correlation_fig = px.imshow(numeric_df.corr(),
                             title='Correlation Heatmap')
correlation_fig.show()
```

[11] ✓ 0.0s Python

Correlation Heatmap

The heatmap displays the correlation matrix for the following variables:

- Credit Utilization Ratio
- Payment History
- Number of Credit Accounts
- Loan Amount
- Interest Rate
- Loan Term

The color scale ranges from 0 (dark blue) to 1 (yellow).

← →

Search

Untitled-1.ipynb • credit_scoring_segmentation.ipynb • Release Notes: 1.89.1

Users > subirams > Desktop > VSCODE > credit_scoring_segmentation.ipynb > ...

+ Code + Markdown | ▶ Run All ↺ Restart ≡ Clear All Outputs | Variables Outline ...

base (Python 3.11.5)

[12] ✓ 0.0s Python

education_level_mapping = {'High School': 1, 'Bachelor': 2, 'Master': 3, 'PhD': 4}
employment_status_mapping = {'Unemployed': 0, 'Employed': 1, 'Self-Employed': 2}

[13] ✓ 0.0s Python

df['Education Level'] = df['Education Level'].map(education_level_mapping)
df['Employment Status'] = df['Employment Status'].map(employment_status_mapping)

[14] ✓ 0.0s Python

credit_scores = []

for index, row in df.iterrows():
 payment_history = row['Payment History']
 credit_utilization_ratio = row['Credit Utilization Ratio']
 number_of_credit_accounts = row['Number of Credit Accounts']
 education_level = row['Education Level']
 employment_status = row['Employment Status']

 credit_score = (payment_history * 0.35) + (credit_utilization_ratio * 0.30) + (number_of_credit_accounts * 0.15) + (education_level * 0.10) + (employment_status * 0.10)
 credit_scores.append(credit_score) #fico formula

df['Credit Score'] = credit_scores

df.head()

	Age	Gender	Marital Status	Education Level	Employment Status	Credit Utilization Ratio	Payment History	Number of Credit Accounts	Loan Amount	Interest Rate	Loan Term	Type of Loan	Credit Score
0	60	Male	Married	3	1	0.22	2685.0	2	4675000	2.65	48	Personal Loan	940.516
1	25	Male	Married	1	0	0.20	2371.0	9	3619000	5.19	60	Auto Loan	831.360
2	30	Female	Single	3	1	0.22	2771.0	6	957000	2.76	12	Auto Loan	971.216

Open

← →

Search

Untitled-1.ipynb • credit_scoring_segmentation.ipynb • Release Notes: 1.89.1

base (Python 3.11.5)

Users > subirams > Desktop > VSCODE > credit_scoring_segmentation.ipynb > ...

+ Code + Markdown ▶ Run All ↺ Restart ≡ Clear All Outputs | Variables Outline ...

...

	Age	Gender	Marital Status	Education Level	Employment Status	Credit Utilization Ratio	Payment History	Number of Credit Accounts	Loan Amount	Interest Rate	Loan Term	Type of Loan	Credit Score
0	60	Male	Married	3	1	0.22	2685.0	2	4675000	2.65	48	Personal Loan	940.516
1	25	Male	Married	1	0	0.20	2371.0	9	3619000	5.19	60	Auto Loan	831.360
2	30	Female	Single	3	1	0.22	2771.0	6	957000	2.76	12	Auto Loan	971.216
3	58	Female	Married	4	0	0.12	1371.0	2	4731000	6.57	60	Auto Loan	480.586
4	32	Male	Married	2	2	0.99	828.0	2	3289000	6.28	36	Personal Loan	290.797

```
from sklearn.cluster import KMeans

X = df[['Credit Score']]
kmeans = KMeans(n_clusters=4, n_init=10, random_state=42)
kmeans.fit(X)
df['Segment'] = kmeans.labels_
```

[15] ✓ 1.8s Python

```
df['Segment'] = df['Segment'].map({2: 'Very Low',
0: 'Low',
1: 'Good',
3: "Excellent"})

df['Segment'] = df['Segment'].astype('category')
```

```
fig = px.scatter(df, x=df.index, y='Credit Score', color='Segment',
color_discrete_sequence=['green', 'blue', 'yellow', 'red'])
fig.update_layout(
    xaxis_title='Customer Index',
    yaxis_title='Credit Score',
    title='Customer Segmentation based on Credit Scores')
```

Open

← →

Search

Release Notes: 1.89.1

Untitled-1.ipynb • credit_scoring_segmentation.ipynb •

Users > subirams > Desktop > VSCODE > credit_scoring_segmentation.ipynb > ...

+ Code

+ Markdown

▶ Run All

↺ Restart

≡ Clear All Outputs

📄 Variables

📖 Outline

⋮

base (Python 3.11.5)

```
}  
fig.show()  
[16] ✓ 0.0s
```

Customer Segmentation based on Credit Scores

Segment

- Excellent
- Low
- Very Low
- Good

[]

Python

[]

Python

Visual Studio Code

