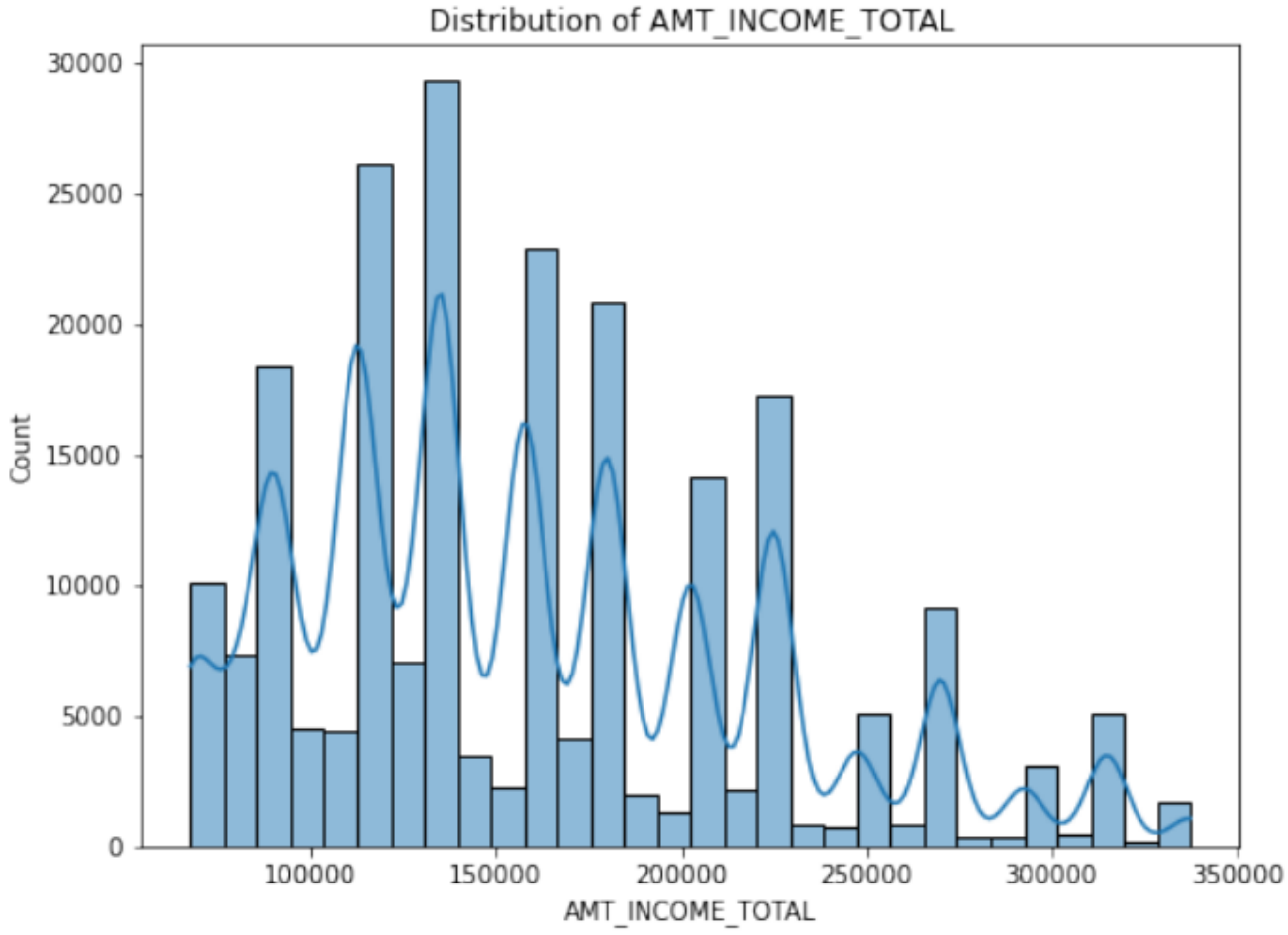
PROJECT: HOUSE LOAN DATA ANALYSIS (PREDICTING DEFAULT PAYMENTS)

SCREENSHOTS OF DATA VISUALIZATION

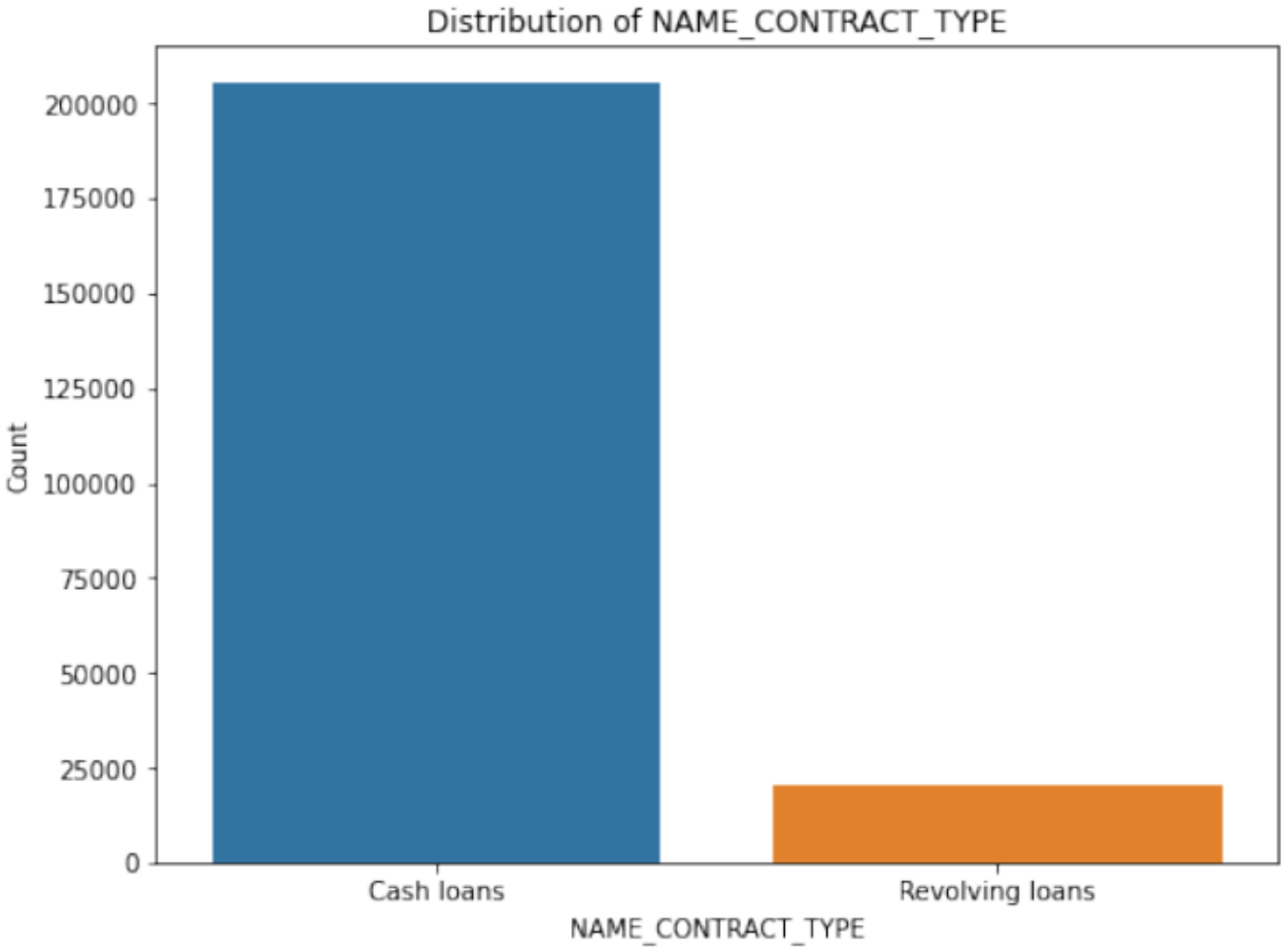
DEEP LEARNING (TENSORFLOW WITH KERAS)

```
# Histogram of a numerical column
plt.figure(figsize=(8, 6))
sns.histplot(data=df, x='AMT_INCOME_TOTAL', bins=30, kde=True)
plt.title('Distribution of AMT_INCOME_TOTAL')
plt.xlabel('AMT_INCOME_TOTAL')
plt.ylabel('Count')
plt.show()
```

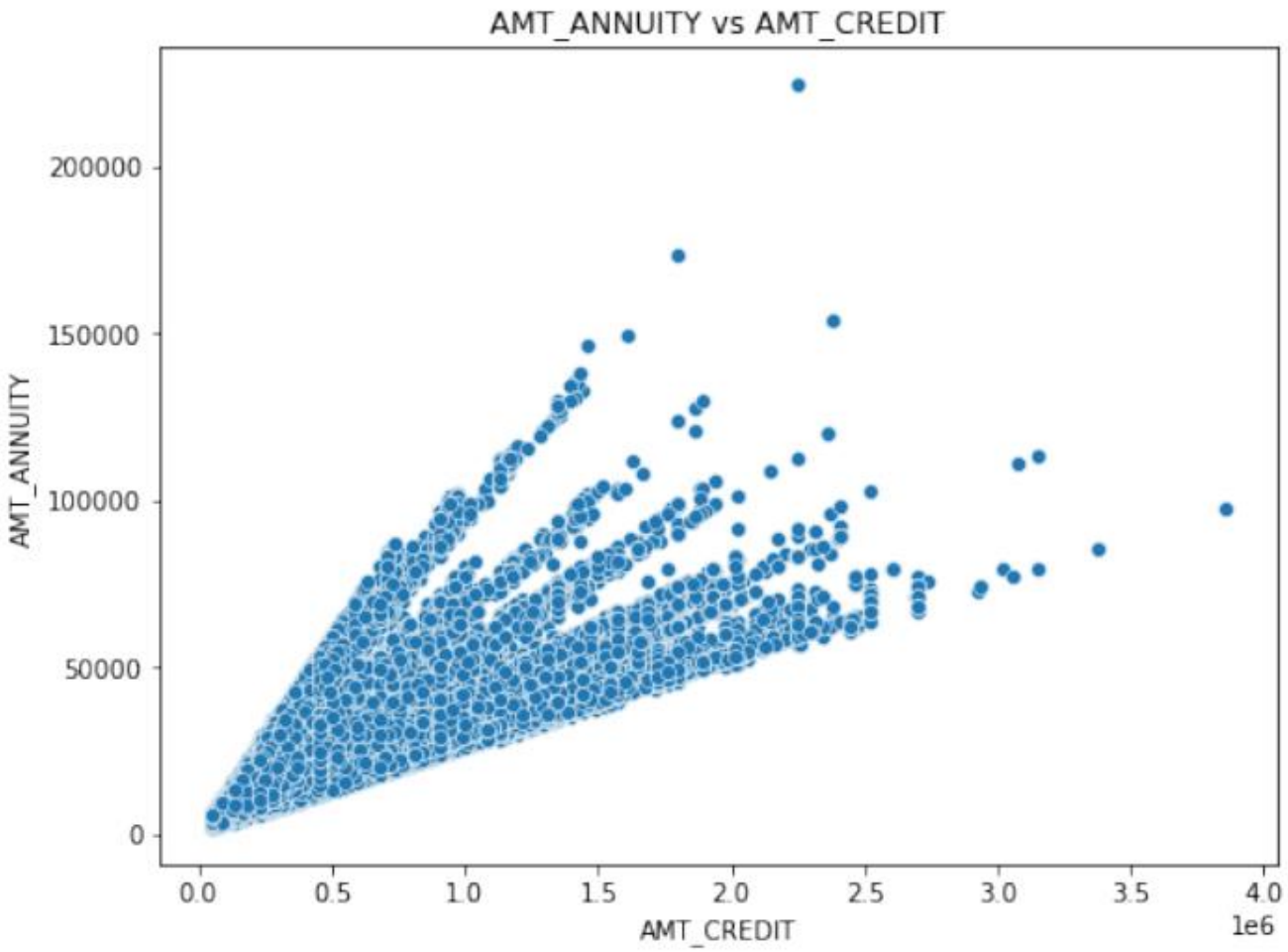


`

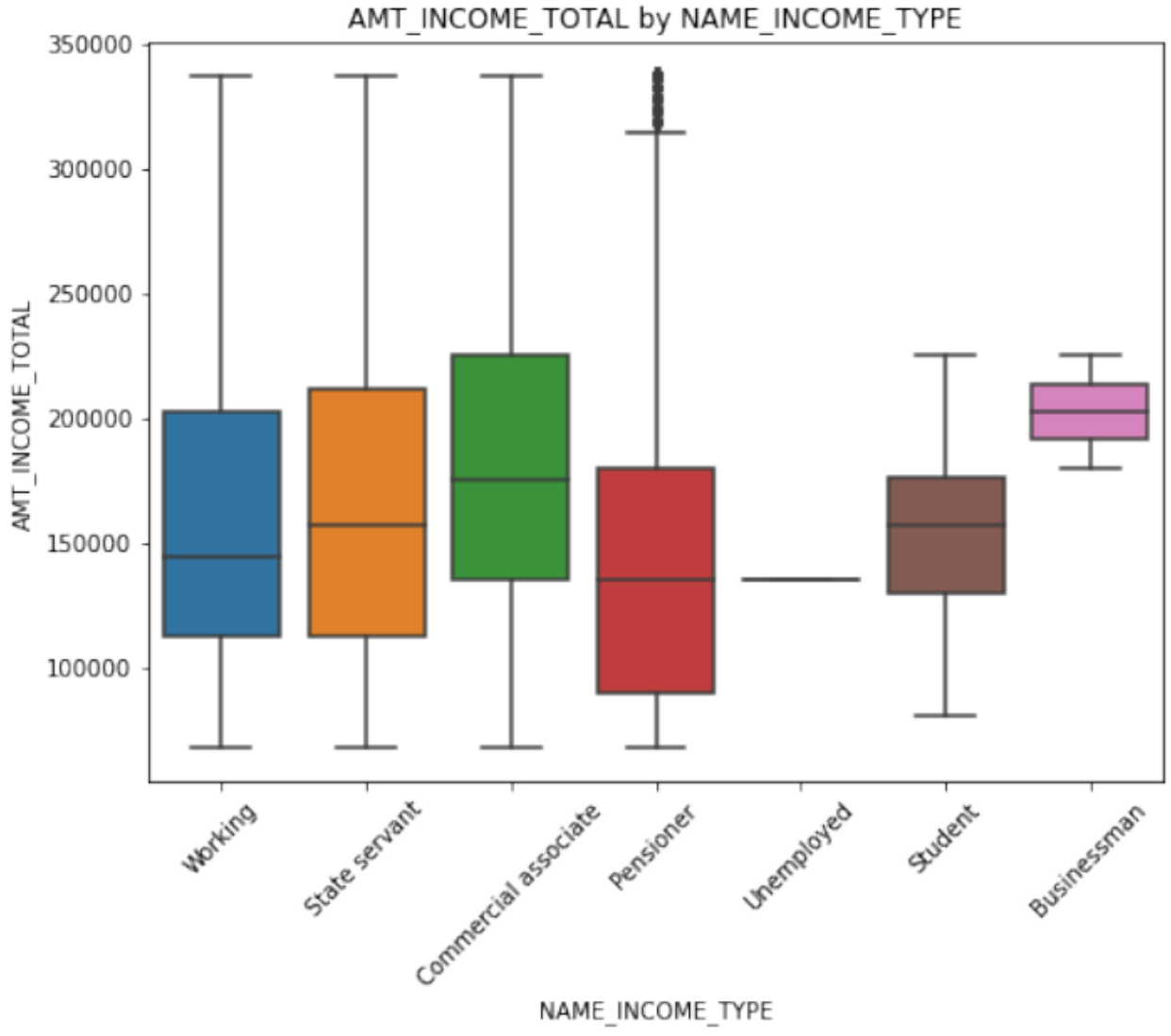
```
# Bar chart of a categorical column
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='NAME_CONTRACT_TYPE')
plt.title('Distribution of NAME_CONTRACT_TYPE')
plt.xlabel('NAME_CONTRACT_TYPE')
plt.ylabel('Count')
plt.show()
```



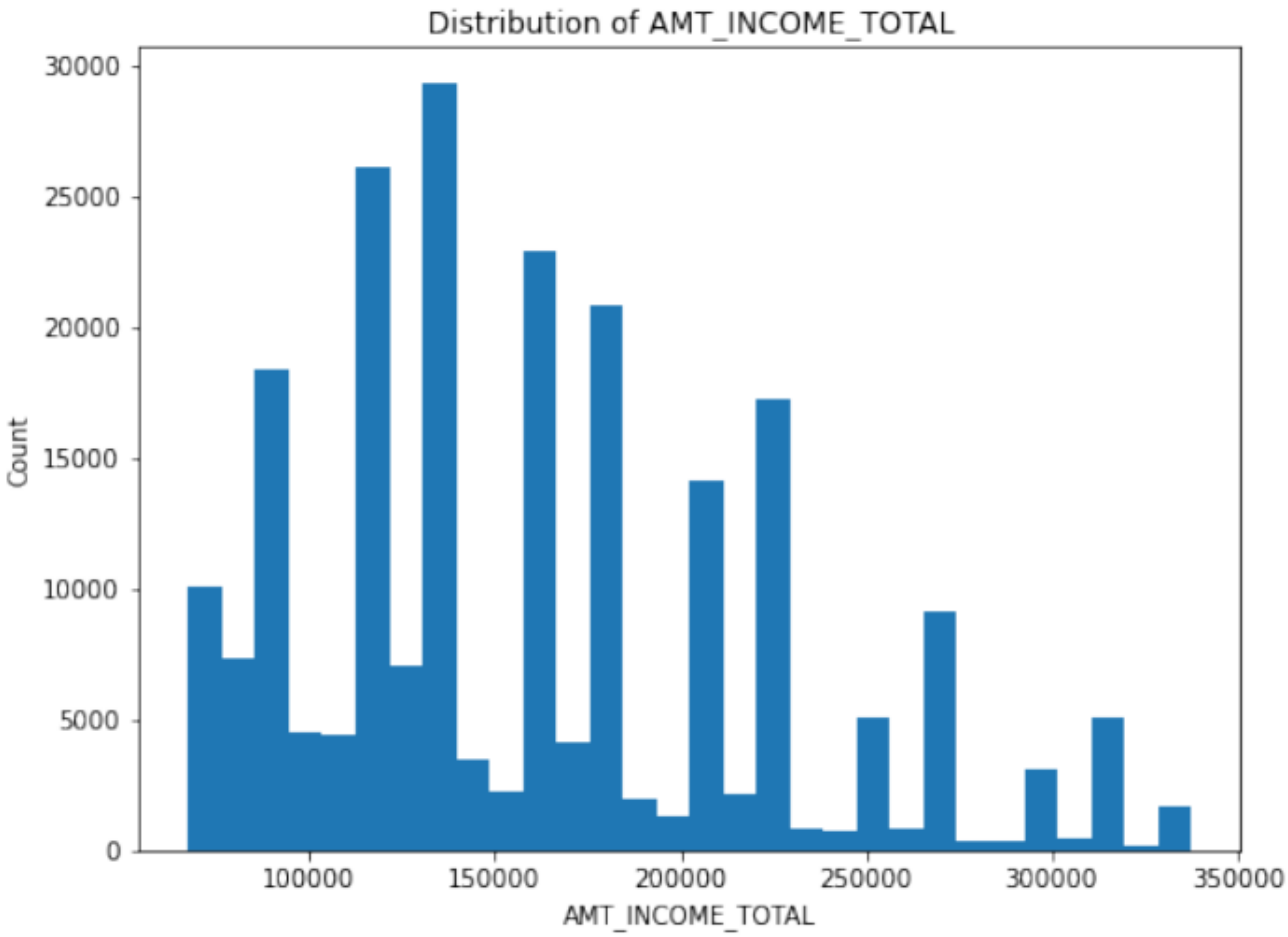
```
# Scatter plot of two numerical columns
plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='AMT_CREDIT', y='AMT_ANNUITY')
plt.title('AMT_ANNUITY vs AMT_CREDIT')
plt.xlabel('AMT_CREDIT')
plt.ylabel('AMT_ANNUITY')
plt.show()
```



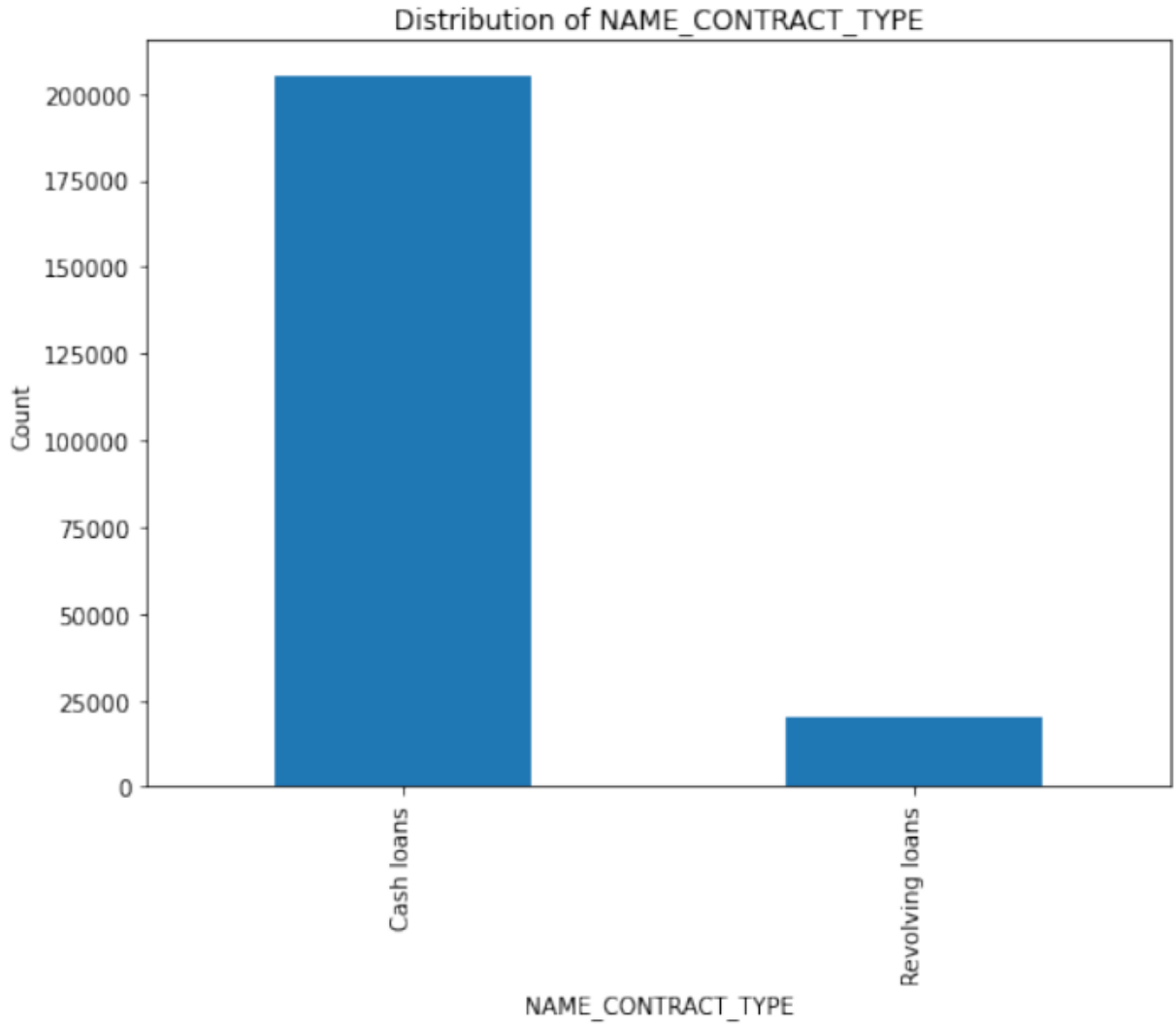
```
# Box plot of a numerical column grouped by a categorical column
plt.figure(figsize=(8, 6))
sns.boxplot(data=df, x='NAME_INCOME_TYPE', y='AMT_INCOME_TOTAL')
plt.title('AMT_INCOME_TOTAL by NAME_INCOME_TYPE')
plt.xlabel('NAME_INCOME_TYPE')
plt.ylabel('AMT_INCOME_TOTAL')
plt.xticks(rotation=45)
plt.show()
```



```
plt.figure(figsize=(8, 6))
plt.hist(df['AMT_INCOME_TOTAL'], bins=30)
plt.title("Distribution of AMT_INCOME_TOTAL")
plt.xlabel("AMT_INCOME_TOTAL")
plt.ylabel("Count")
plt.show()
```



```
# Explore the distribution of a categorical column
plt.figure(figsize=(8, 6))
df['NAME_CONTRACT_TYPE'].value_counts().plot(kind='bar')
plt.title("Distribution of NAME_CONTRACT_TYPE")
plt.xlabel("NAME_CONTRACT_TYPE")
plt.ylabel("Count")
plt.show()
```

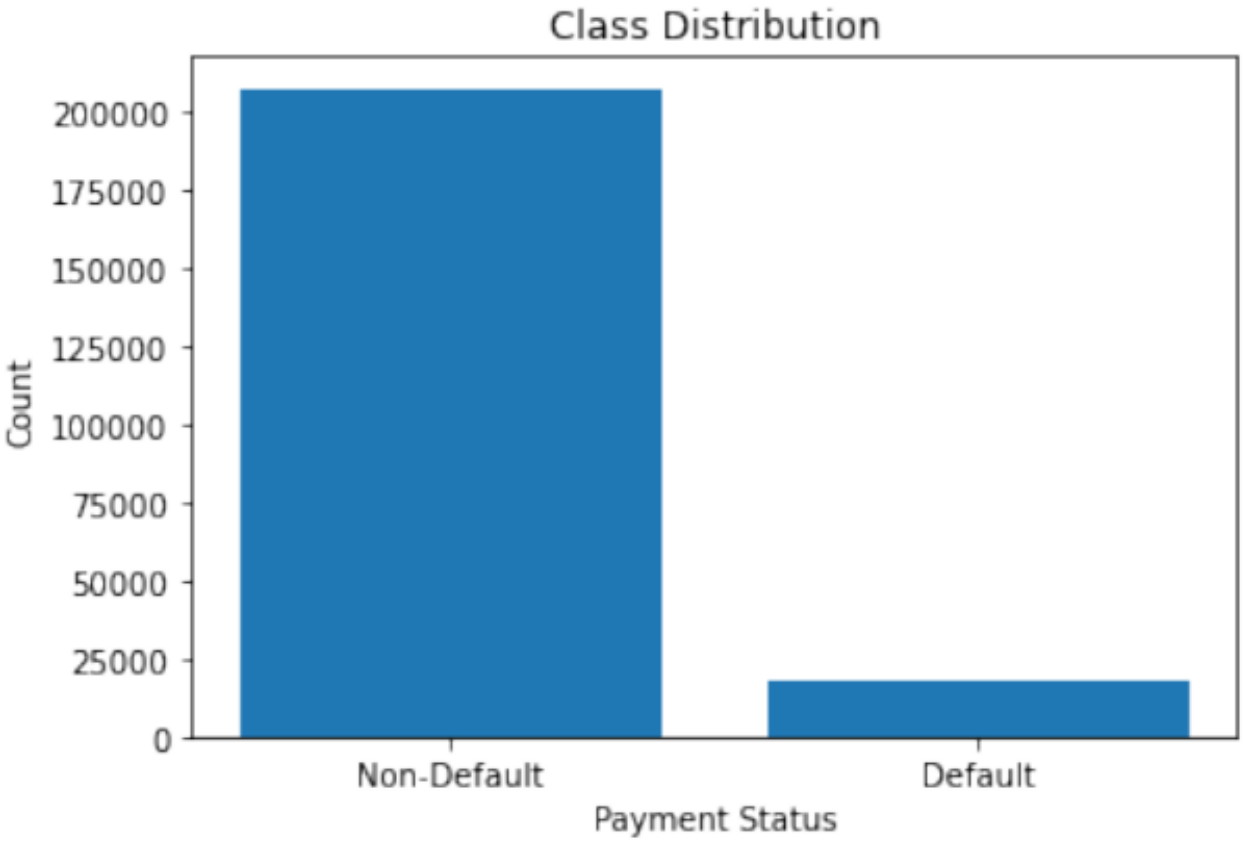


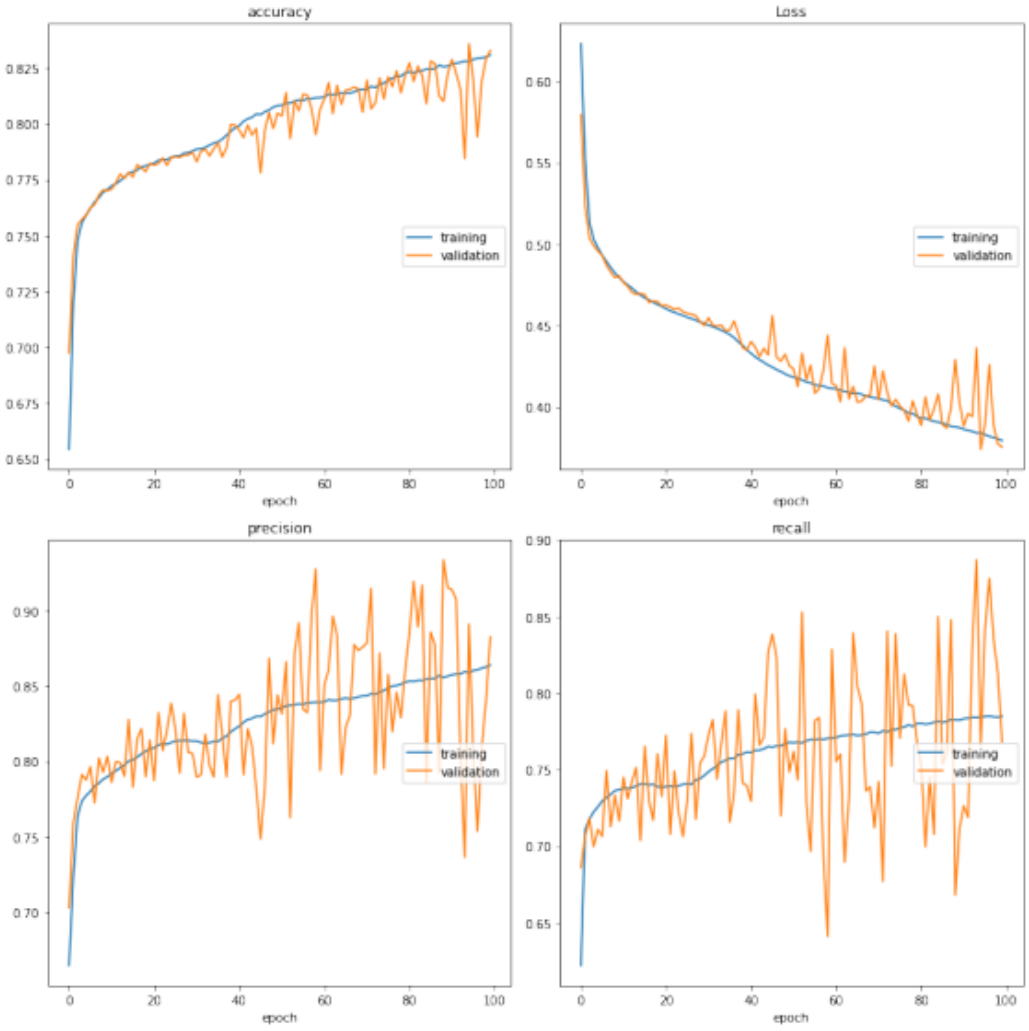
```python
import matplotlib.pyplot as plt

# Assuming you have a DataFrame called 'data' with a column 'TARGET' indicating␣
 ↪default payments

# Count the number of default and non-default payments
default_count = final_df[final_df['TARGET'] == 1].shape[0]
non_default_count = final_df[final_df['TARGET'] == 0].shape[0]

# Create a bar plot to visualize the class distribution
plt.bar(['Non-Default', 'Default'], [non_default_count, default_count])
plt.xlabel('Payment Status')
plt.ylabel('Count')
plt.title('Class Distribution')
plt.show()
```






```
accuracy
        training                (min:    0.654, max:    0.831, cur:    0.831)
        validation              (min:    0.698, max:    0.836, cur:    0.833)
Loss
        training                (min:    0.380, max:    0.623, cur:    0.380)
        validation              (min:    0.374, max:    0.580, cur:    0.376)
precision
        training                (min:    0.665, max:    0.864, cur:    0.864)
        validation              (min:    0.703, max:    0.934, cur:    0.883)
recall
        training                (min:    0.622, max:    0.785, cur:    0.785)
        validation              (min:    0.641, max:    0.887, cur:    0.768)
```

```python
# Plot ROC Curve
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, label="ROC curve (AUC = {:.4f})".format(auc))
plt.plot([0, 1], [0, 1], "r--")
plt.xlim([0, 1])
plt.ylim([0, 1])
plt.title("Receiver Operating Characteristic (ROC) Curve")
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.legend(loc="lower right")
plt.show()
```