Financial Transaction Analysis and Prediction Using Large Language Models (LLMs)

In [3]:

```
#Installing necessary libraries
!pip install catboost
!pip install transformers
!pip install torch
!pip install pandas numpy matplotlib scikit-learn imbalanced-learn plotly panel
!pip install keras-tuner
!pip install gpt4all
!pip install langchain
!pip install langchain_community
!pip install scikit-learn
!pip install transformers
!pip install ollama
```

```
Requirement already satisfied: catboost in c:\users\amuly\anaconda3\lib\site-package
s (1.2.5)
Requirement already satisfied: graphviz in c:\users\amuly\anaconda3\lib\site-package
s (from catboost) (0.20.3)
Requirement already satisfied: matplotlib in c:\users\amuly\anaconda3\lib\site-packa
ges (from catboost) (3.8.4)
Requirement already satisfied: numpy>=1.16.0 in c:\users\amuly\anaconda3\lib\site-pa
ckages (from catboost) (1.26.4)
Requirement already satisfied: pandas>=0.24 in c:\users\amuly\anaconda3\lib\site-pac
kages (from catboost) (2.2.2)
Requirement already satisfied: scipy in c:\users\amuly\anaconda3\lib\site-packages
(from catboost) (1.13.1)
Requirement already satisfied: plotly in c:\users\amuly\anaconda3\lib\site-packages
(from catboost) (5.22.0)
Requirement already satisfied: six in c:\users\amuly\anaconda3\lib\site-packages (fr
om catboost) (1.16.0)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\amuly\anaconda3\li
b\site-packages (from pandas>=0.24->catboost) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\amuly\anaconda3\lib\site-pac
kages (from pandas>=0.24->catboost) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\amuly\anaconda3\lib\site-p
ackages (from pandas>=0.24->catboost) (2023.3)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\amuly\anaconda3\lib\site
-packages (from matplotlib->catboost) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\amuly\anaconda3\lib\site-pac
kages (from matplotlib->catboost) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\amuly\anaconda3\lib\sit
e-packages (from matplotlib->catboost) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\amuly\anaconda3\lib\sit
e-packages (from matplotlib->catboost) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\amuly\anaconda3\lib\site-
packages (from matplotlib->catboost) (23.2)
Requirement already satisfied: pillow>=8 in c:\users\amuly\anaconda3\lib\site-packag
es (from matplotlib->catboost) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\amuly\anaconda3\lib\site
-packages (from matplotlib->catboost) (3.0.9)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\amuly\anaconda3\lib\site-
packages (from plotly->catboost) (8.2.2)
Requirement already satisfied: transformers in c:\users\amuly\anaconda3\lib\site-pac
kages (4.36.2)
Requirement already satisfied: filelock in c:\users\amuly\anaconda3\lib\site-package
s (from transformers) (3.13.1)
Requirement already satisfied: huggingface-hub<1.0,>=0.19.3 in c:\users\amuly\anacon
da3\lib\site-packages (from transformers) (0.23.1)
Requirement already satisfied: numpy>=1.17 in c:\users\amuly\anaconda3\lib\site-pack
ages (from transformers) (1.26.4)
Requirement already satisfied: packaging>=20.0 in c:\users\amuly\anaconda3\lib\site-
packages (from transformers) (23.2)
Requirement already satisfied: pyyaml>=5.1 in c:\users\amuly\anaconda3\lib\site-pack
ages (from transformers) (6.0.1)
Requirement already satisfied: regex!=2019.12.17 in c:\users\amuly\anaconda3\lib\sit
e-packages (from transformers) (2023.10.3)
Requirement already satisfied: requests in c:\users\amuly\anaconda3\lib\site-package
s (from transformers) (2.32.2)
Requirement already satisfied: tokenizers<0.19,>=0.14 in c:\users\amuly\anaconda3\li
b\site-packages (from transformers) (0.15.1)
```

```
Requirement already satisfied: safetensors>=0.3.1 in c:\users\amuly\anaconda3\lib\si
te-packages (from transformers) (0.4.2)
Requirement already satisfied: tqdm>=4.27 in c:\users\amuly\anaconda3\lib\site-packa
ges (from transformers) (4.66.4)
Requirement already satisfied: fsspec>=2023.5.0 in c:\users\amuly\anaconda3\lib\site
-packages (from huggingface-hub<1.0,>=0.19.3->transformers) (2024.2.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in c:\users\amuly\anaconda
3\lib\site-packages (from huggingface-hub<1.0,>=0.19.3->transformers) (4.11.0)
Requirement already satisfied: colorama in c:\users\amuly\anaconda3\lib\site-package
s (from tqdm>=4.27->transformers) (0.4.6)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\amuly\anaconda3
\lib\site-packages (from requests->transformers) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\amuly\anaconda3\lib\site-pac
kages (from requests->transformers) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\amuly\anaconda3\lib\si
te-packages (from requests->transformers) (2.2.2)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\amuly\anaconda3\lib\si
te-packages (from requests->transformers) (2024.6.2)
Requirement already satisfied: torch in c:\users\amuly\anaconda3\lib\site-packages
(2.2.2)
Requirement already satisfied: filelock in c:\users\amuly\anaconda3\lib\site-package
s (from torch) (3.13.1)
Requirement already satisfied: typing-extensions>=4.8.0 in c:\users\amuly\anaconda3
\lib\site-packages (from torch) (4.11.0)
Requirement already satisfied: sympy in c:\users\amuly\anaconda3\lib\site-packages
(from torch) (1.12)
Requirement already satisfied: networkx in c:\users\amuly\anaconda3\lib\site-package
s (from torch) (3.3)
Requirement already satisfied: jinja2 in c:\users\amuly\anaconda3\lib\site-packages
(from torch) (3.1.4)
Requirement already satisfied: fsspec in c:\users\amuly\anaconda3\lib\site-packages
(from torch) (2024.2.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\amuly\anaconda3\lib\site-
packages (from jinja2->torch) (2.1.3)
Requirement already satisfied: mpmath>=0.19 in c:\users\amuly\anaconda3\lib\site-pac
kages (from sympy->torch) (1.3.0)
Requirement already satisfied: pandas in c:\users\amuly\anaconda3\lib\site-packages
(2.2.2)
Requirement already satisfied: numpy in c:\users\amuly\anaconda3\lib\site-packages
(1.26.4)
Requirement already satisfied: matplotlib in c:\users\amuly\anaconda3\lib\site-packa
ges (3.8.4)
Requirement already satisfied: scikit-learn in c:\users\amuly\anaconda3\lib\site-pac
kages (1.4.2)
Requirement already satisfied: imbalanced-learn in c:\users\amuly\anaconda3\lib\site
-packages (0.11.0)
Requirement already satisfied: plotly in c:\users\amuly\anaconda3\lib\site-packages
(5.22.0)
Requirement already satisfied: panel in c:\users\amuly\anaconda3\lib\site-packages
(1.4.4)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\amuly\anaconda3\li
b\site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\amuly\anaconda3\lib\site-pac
kages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\amuly\anaconda3\lib\site-p
ackages (from pandas) (2023.3)
```

```
Requirement already satisfied: contourpy>=1.0.1 in c:\users\amuly\anaconda3\lib\site
-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\amuly\anaconda3\lib\site-pac
kages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\amuly\anaconda3\lib\sit
e-packages (from matplotlib) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\amuly\anaconda3\lib\sit
e-packages (from matplotlib) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\amuly\anaconda3\lib\site-
packages (from matplotlib) (23.2)
Requirement already satisfied: pillow>=8 in c:\users\amuly\anaconda3\lib\site-packag
es (from matplotlib) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\amuly\anaconda3\lib\site
-packages (from matplotlib) (3.0.9)
Requirement already satisfied: scipy>=1.6.0 in c:\users\amuly\anaconda3\lib\site-pac
kages (from scikit-learn) (1.13.1)
Requirement already satisfied: joblib>=1.2.0 in c:\users\amuly\anaconda3\lib\site-pa
ckages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\amuly\anaconda3\lib
\site-packages (from scikit-learn) (3.5.0)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\amuly\anaconda3\lib\site-
packages (from plotly) (8.2.2)
Requirement already satisfied: bokeh<3.5.0,>=3.4.0 in c:\users\amuly\anaconda3\lib\s
ite-packages (from panel) (3.4.1)
Requirement already satisfied: param<3.0,>=2.1.0 in c:\users\amuly\anaconda3\lib\sit
e-packages (from panel) (2.1.1)
Requirement already satisfied: pyviz-comms>=2.0.0 in c:\users\amuly\anaconda3\lib\si
te-packages (from panel) (3.0.2)
Requirement already satisfied: xyzservices>=2021.09.1 in c:\users\amuly\anaconda3\li
b\site-packages (from panel) (2022.9.0)
Requirement already satisfied: markdown in c:\users\amuly\anaconda3\lib\site-package
s (from panel) (3.4.1)
Requirement already satisfied: markdown-it-py in c:\users\amuly\anaconda3\lib\site-p
ackages (from panel) (2.2.0)
Requirement already satisfied: linkify-it-py in c:\users\amuly\anaconda3\lib\site-pa
ckages (from panel) (2.0.0)
Requirement already satisfied: mdit-py-plugins in c:\users\amuly\anaconda3\lib\site-
packages (from panel) (0.3.0)
Requirement already satisfied: requests in c:\users\amuly\anaconda3\lib\site-package
s (from panel) (2.32.2)
Requirement already satisfied: tqdm>=4.48.0 in c:\users\amuly\anaconda3\lib\site-pac
kages (from panel) (4.66.4)
Requirement already satisfied: bleach in c:\users\amuly\anaconda3\lib\site-packages
(from panel) (4.1.0)
Requirement already satisfied: typing-extensions in c:\users\amuly\anaconda3\lib\sit
e-packages (from panel) (4.11.0)
Requirement already satisfied: Jinja2>=2.9 in c:\users\amuly\anaconda3\lib\site-pack
ages (from bokeh<3.5.0,>=3.4.0->panel) (3.1.4)
Requirement already satisfied: PyYAML>=3.10 in c:\users\amuly\anaconda3\lib\site-pac
kages (from bokeh<3.5.0,>=3.4.0->panel) (6.0.1)
Requirement already satisfied: tornado>=6.2 in c:\users\amuly\anaconda3\lib\site-pac
kages (from bokeh<3.5.0,>=3.4.0->panel) (6.4.1)
Requirement already satisfied: six>=1.5 in c:\users\amuly\anaconda3\lib\site-package
s (from python-dateutil>=2.8.2->pandas) (1.16.0)
Requirement already satisfied: colorama in c:\users\amuly\anaconda3\lib\site-package
s (from tqdm>=4.48.0->panel) (0.4.6)
```

```
Requirement already satisfied: webencodings in c:\users\amuly\anaconda3\lib\site-pac
kages (from bleach->panel) (0.5.1)
Requirement already satisfied: uc-micro-py in c:\users\amuly\anaconda3\lib\site-pack
ages (from linkify-it-py->panel) (1.0.1)
Requirement already satisfied: mdurl~=0.1 in c:\users\amuly\anaconda3\lib\site-packa
ges (from markdown-it-py->panel) (0.1.0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\amuly\anaconda3
\lib\site-packages (from requests->panel) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\amuly\anaconda3\lib\site-pac
kages (from requests->panel) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\amuly\anaconda3\lib\si
te-packages (from requests->panel) (2.2.2)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\amuly\anaconda3\lib\si
te-packages (from requests->panel) (2024.6.2)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\amuly\anaconda3\lib\site-
packages (from Jinja2>=2.9->bokeh<3.5.0,>=3.4.0->panel) (2.1.3)
Requirement already satisfied: keras-tuner in c:\users\amuly\anaconda3\lib\site-pack
ages (1.4.7)
Requirement already satisfied: keras in c:\users\amuly\anaconda3\lib\site-packages
(from keras-tuner) (3.1.1)
Requirement already satisfied: packaging in c:\users\amuly\anaconda3\lib\site-packag
es (from keras-tuner) (23.2)
Requirement already satisfied: requests in c:\users\amuly\anaconda3\lib\site-package
s (from keras-tuner) (2.32.2)
Requirement already satisfied: kt-legacy in c:\users\amuly\anaconda3\lib\site-packag
es (from keras-tuner) (1.0.5)
Requirement already satisfied: absl-py in c:\users\amuly\anaconda3\lib\site-packages
(from keras->keras-tuner) (2.1.0)
Requirement already satisfied: numpy in c:\users\amuly\anaconda3\lib\site-packages
(from keras->keras-tuner) (1.26.4)
Requirement already satisfied: rich in c:\users\amuly\anaconda3\lib\site-packages (f
rom keras->keras-tuner) (13.3.5)
Requirement already satisfied: namex in c:\users\amuly\anaconda3\lib\site-packages
(from keras->keras-tuner) (0.0.7)
Requirement already satisfied: h5py in c:\users\amuly\anaconda3\lib\site-packages (f
rom keras->keras-tuner) (3.11.0)
Requirement already satisfied: optree in c:\users\amuly\anaconda3\lib\site-packages
(from keras->keras-tuner) (0.11.0)
Requirement already satisfied: ml-dtypes in c:\users\amuly\anaconda3\lib\site-packag
es (from keras->keras-tuner) (0.3.2)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\amuly\anaconda3
\lib\site-packages (from requests->keras-tuner) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\amuly\anaconda3\lib\site-pac
kages (from requests->keras-tuner) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\amuly\anaconda3\lib\si
te-packages (from requests->keras-tuner) (2.2.2)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\amuly\anaconda3\lib\si
te-packages (from requests->keras-tuner) (2024.6.2)
Requirement already satisfied: typing-extensions>=4.0.0 in c:\users\amuly\anaconda3
\lib\site-packages (from optree->keras->keras-tuner) (4.11.0)
Requirement already satisfied: markdown-it-py<3.0.0,>=2.2.0 in c:\users\amuly\anacon
da3\lib\site-packages (from rich->keras->keras-tuner) (2.2.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in c:\users\amuly\anaconda3\l
ib\site-packages (from rich->keras->keras-tuner) (2.15.1)
Requirement already satisfied: mdurl~=0.1 in c:\users\amuly\anaconda3\lib\site-packa
ges (from markdown-it-py<3.0.0,>=2.2.0->rich->keras->keras-tuner) (0.1.0)
```

```
Requirement already satisfied: gpt4all in c:\users\amuly\anaconda3\lib\site-packages
(2.8.2)
Requirement already satisfied: requests in c:\users\amuly\anaconda3\lib\site-package
s (from gpt4all) (2.32.2)
Requirement already satisfied: tqdm in c:\users\amuly\anaconda3\lib\site-packages (f
rom gpt4all) (4.66.4)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\amuly\anaconda3
\lib\site-packages (from requests->gpt4all) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\amuly\anaconda3\lib\site-pac
kages (from requests->gpt4all) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\amuly\anaconda3\lib\si
te-packages (from requests->gpt4all) (2.2.2)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\amuly\anaconda3\lib\si
te-packages (from requests->gpt4all) (2024.6.2)
Requirement already satisfied: colorama in c:\users\amuly\anaconda3\lib\site-package
s (from tqdm->gpt4all) (0.4.6)
Requirement already satisfied: langchain in c:\users\amuly\anaconda3\lib\site-packag
es (0.2.6)
Requirement already satisfied: PyYAML>=5.3 in c:\users\amuly\anaconda3\lib\site-pack
ages (from langchain) (6.0.1)
Requirement already satisfied: SQLAlchemy<3,>=1.4 in c:\users\amuly\anaconda3\lib\si
te-packages (from langchain) (2.0.30)
Requirement already satisfied: aiohttp<4.0.0,>=3.8.3 in c:\users\amuly\anaconda3\lib
\site-packages (from langchain) (3.9.5)
Requirement already satisfied: langchain-core<0.3.0,>=0.2.10 in c:\users\amuly\anaco
nda3\lib\site-packages (from langchain) (0.2.10)
Requirement already satisfied: langchain-text-splitters<0.3.0,>=0.2.0 in c:\users\am
uly\anaconda3\lib\site-packages (from langchain) (0.2.2)
Requirement already satisfied: langsmith<0.2.0,>=0.1.17 in c:\users\amuly\anaconda3
\lib\site-packages (from langchain) (0.1.82)
Requirement already satisfied: numpy<2,>=1 in c:\users\amuly\anaconda3\lib\site-pack
ages (from langchain) (1.26.4)
Requirement already satisfied: pydantic<3,>=1 in c:\users\amuly\anaconda3\lib\site-p
ackages (from langchain) (2.5.3)
Requirement already satisfied: requests<3,>=2 in c:\users\amuly\anaconda3\lib\site-p
ackages (from langchain) (2.32.2)
Requirement already satisfied: tenacity!=8.4.0,<9.0.0,>=8.1.0 in c:\users\amuly\anac
onda3\lib\site-packages (from langchain) (8.2.2)
Requirement already satisfied: aiosignal>=1.1.2 in c:\users\amuly\anaconda3\lib\site
-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (1.2.0)
Requirement already satisfied: attrs>=17.3.0 in c:\users\amuly\anaconda3\lib\site-pa
ckages (from aiohttp<4.0.0,>=3.8.3->langchain) (23.1.0)
Requirement already satisfied: frozenlist>=1.1.1 in c:\users\amuly\anaconda3\lib\sit
e-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (1.4.0)
Requirement already satisfied: multidict<7.0,>=4.5 in c:\users\amuly\anaconda3\lib\s
ite-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (6.0.4)
Requirement already satisfied: yarl<2.0,>=1.0 in c:\users\amuly\anaconda3\lib\site-p
ackages (from aiohttp<4.0.0,>=3.8.3->langchain) (1.9.3)
Requirement already satisfied: jsonpatch<2.0,>=1.33 in c:\users\amuly\anaconda3\lib
\site-packages (from langchain-core<0.3.0,>=0.2.10->langchain) (1.33)
Requirement already satisfied: packaging<25,>=23.2 in c:\users\amuly\anaconda3\lib\s
ite-packages (from langchain-core<0.3.0,>=0.2.10->langchain) (23.2)
Requirement already satisfied: orjson<4.0.0,>=3.9.14 in c:\users\amuly\anaconda3\lib
\site-packages (from langsmith<0.2.0,>=0.1.17->langchain) (3.10.5)
Requirement already satisfied: annotated-types>=0.4.0 in c:\users\amuly\anaconda3\li
b\site-packages (from pydantic<3,>=1->langchain) (0.6.0)
```

```
Requirement already satisfied: pydantic-core==2.14.6 in c:\users\amuly\anaconda3\lib
\site-packages (from pydantic<3,>=1->langchain) (2.14.6)
Requirement already satisfied: typing-extensions>=4.6.1 in c:\users\amuly\anaconda3
\lib\site-packages (from pydantic<3,>=1->langchain) (4.11.0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\amuly\anaconda3
\lib\site-packages (from requests<3,>=2->langchain) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\amuly\anaconda3\lib\site-pac
kages (from requests<3,>=2->langchain) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\amuly\anaconda3\lib\si
te-packages (from requests<3,>=2->langchain) (2.2.2)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\amuly\anaconda3\lib\si
te-packages (from requests<3,>=2->langchain) (2024.6.2)
Requirement already satisfied: greenlet!=0.4.17 in c:\users\amuly\anaconda3\lib\site
-packages (from SQLAlchemy<3,>=1.4->langchain) (3.0.1)
Requirement already satisfied: jsonpointer>=1.9 in c:\users\amuly\anaconda3\lib\site
-packages (from jsonpatch<2.0,>=1.33->langchain-core<0.3.0,>=0.2.10->langchain) (2.
1)
Requirement already satisfied: langchain_community in c:\users\amuly\anaconda3\lib\s
ite-packages (0.2.6)
Requirement already satisfied: PyYAML>=5.3 in c:\users\amuly\anaconda3\lib\site-pack
ages (from langchain_community) (6.0.1)
Requirement already satisfied: SQLAlchemy<3,>=1.4 in c:\users\amuly\anaconda3\lib\si
te-packages (from langchain_community) (2.0.30)
Requirement already satisfied: aiohttp<4.0.0,>=3.8.3 in c:\users\amuly\anaconda3\lib
\site-packages (from langchain_community) (3.9.5)
Requirement already satisfied: dataclasses-json<0.7,>=0.5.7 in c:\users\amuly\anacon
da3\lib\site-packages (from langchain_community) (0.6.7)
Requirement already satisfied: langchain<0.3.0,>=0.2.6 in c:\users\amuly\anaconda3\l
ib\site-packages (from langchain_community) (0.2.6)
Requirement already satisfied: langchain-core<0.3.0,>=0.2.10 in c:\users\amuly\anaco
nda3\lib\site-packages (from langchain_community) (0.2.10)
Requirement already satisfied: langsmith<0.2.0,>=0.1.0 in c:\users\amuly\anaconda3\l
ib\site-packages (from langchain_community) (0.1.82)
Requirement already satisfied: numpy<2,>=1 in c:\users\amuly\anaconda3\lib\site-pack
ages (from langchain_community) (1.26.4)
Requirement already satisfied: requests<3,>=2 in c:\users\amuly\anaconda3\lib\site-p
ackages (from langchain_community) (2.32.2)
Requirement already satisfied: tenacity!=8.4.0,<9.0.0,>=8.1.0 in c:\users\amuly\anac
onda3\lib\site-packages (from langchain_community) (8.2.2)
Requirement already satisfied: aiosignal>=1.1.2 in c:\users\amuly\anaconda3\lib\site
-packages (from aiohttp<4.0.0,>=3.8.3->langchain_community) (1.2.0)
Requirement already satisfied: attrs>=17.3.0 in c:\users\amuly\anaconda3\lib\site-pa
ckages (from aiohttp<4.0.0,>=3.8.3->langchain_community) (23.1.0)
Requirement already satisfied: frozenlist>=1.1.1 in c:\users\amuly\anaconda3\lib\sit
e-packages (from aiohttp<4.0.0,>=3.8.3->langchain_community) (1.4.0)
Requirement already satisfied: multidict<7.0,>=4.5 in c:\users\amuly\anaconda3\lib\s
ite-packages (from aiohttp<4.0.0,>=3.8.3->langchain_community) (6.0.4)
Requirement already satisfied: yarl<2.0,>=1.0 in c:\users\amuly\anaconda3\lib\site-p
ackages (from aiohttp<4.0.0,>=3.8.3->langchain_community) (1.9.3)
Requirement already satisfied: marshmallow<4.0.0,>=3.18.0 in c:\users\amuly\anaconda
3\lib\site-packages (from dataclasses-json<0.7,>=0.5.7->langchain_community) (3.21.
3)
Requirement already satisfied: typing-inspect<1,>=0.4.0 in c:\users\amuly\anaconda3
\lib\site-packages (from dataclasses-json<0.7,>=0.5.7->langchain_community) (0.9.0)
Requirement already satisfied: langchain-text-splitters<0.3.0,>=0.2.0 in c:\users\am
uly\anaconda3\lib\site-packages (from langchain<0.3.0,>=0.2.6->langchain_community)
```

```
(0.2.2)
Requirement already satisfied: pydantic<3,>=1 in c:\users\amuly\anaconda3\lib\site-p
ackages (from langchain<0.3.0,>=0.2.6->langchain_community) (2.5.3)
Requirement already satisfied: jsonpatch<2.0,>=1.33 in c:\users\amuly\anaconda3\lib
\site-packages (from langchain-core<0.3.0,>=0.2.10->langchain_community) (1.33)
Requirement already satisfied: packaging<25,>=23.2 in c:\users\amuly\anaconda3\lib\s
ite-packages (from langchain-core<0.3.0,>=0.2.10->langchain_community) (23.2)
Requirement already satisfied: orjson<4.0.0,>=3.9.14 in c:\users\amuly\anaconda3\lib
\site-packages (from langsmith<0.2.0,>=0.1.0->langchain_community) (3.10.5)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\amuly\anaconda3
\lib\site-packages (from requests<3,>=2->langchain_community) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\amuly\anaconda3\lib\site-pac
kages (from requests<3,>=2->langchain_community) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\amuly\anaconda3\lib\si
te-packages (from requests<3,>=2->langchain_community) (2.2.2)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\amuly\anaconda3\lib\si
te-packages (from requests<3,>=2->langchain_community) (2024.6.2)
Requirement already satisfied: typing-extensions>=4.6.0 in c:\users\amuly\anaconda3
\lib\site-packages (from SQLAlchemy<3,>=1.4->langchain_community) (4.11.0)
Requirement already satisfied: greenlet!=0.4.17 in c:\users\amuly\anaconda3\lib\site
-packages (from SQLAlchemy<3,>=1.4->langchain_community) (3.0.1)
Requirement already satisfied: jsonpointer>=1.9 in c:\users\amuly\anaconda3\lib\site
-packages (from jsonpatch<2.0,>=1.33->langchain-core<0.3.0,>=0.2.10->langchain_commu
nity) (2.1)
Requirement already satisfied: annotated-types>=0.4.0 in c:\users\amuly\anaconda3\li
b\site-packages (from pydantic<3,>=1->langchain<0.3.0,>=0.2.6->langchain_community)
(0.6.0)
Requirement already satisfied: pydantic-core==2.14.6 in c:\users\amuly\anaconda3\lib
\site-packages (from pydantic<3,>=1->langchain<0.3.0,>=0.2.6->langchain_community)
(2.14.6)
Requirement already satisfied: mypy-extensions>=0.3.0 in c:\users\amuly\anaconda3\li
b\site-packages (from typing-inspect<1,>=0.4.0->dataclasses-json<0.7,>=0.5.7->langch
ain_community) (1.0.0)
Requirement already satisfied: scikit-learn in c:\users\amuly\anaconda3\lib\site-pac
kages (1.4.2)
Requirement already satisfied: numpy>=1.19.5 in c:\users\amuly\anaconda3\lib\site-pa
ckages (from scikit-learn) (1.26.4)
Requirement already satisfied: scipy>=1.6.0 in c:\users\amuly\anaconda3\lib\site-pac
kages (from scikit-learn) (1.13.1)
Requirement already satisfied: joblib>=1.2.0 in c:\users\amuly\anaconda3\lib\site-pa
ckages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\amuly\anaconda3\lib
\site-packages (from scikit-learn) (3.5.0)
Requirement already satisfied: transformers in c:\users\amuly\anaconda3\lib\site-pac
kages (4.36.2)
Requirement already satisfied: filelock in c:\users\amuly\anaconda3\lib\site-package
s (from transformers) (3.13.1)
Requirement already satisfied: huggingface-hub<1.0,>=0.19.3 in c:\users\amuly\anacon
da3\lib\site-packages (from transformers) (0.23.1)
Requirement already satisfied: numpy>=1.17 in c:\users\amuly\anaconda3\lib\site-pack
ages (from transformers) (1.26.4)
Requirement already satisfied: packaging>=20.0 in c:\users\amuly\anaconda3\lib\site-
packages (from transformers) (23.2)
Requirement already satisfied: pyyaml>=5.1 in c:\users\amuly\anaconda3\lib\site-pack
ages (from transformers) (6.0.1)
Requirement already satisfied: regex!=2019.12.17 in c:\users\amuly\anaconda3\lib\sit
```

```
e-packages (from transformers) (2023.10.3)
Requirement already satisfied: requests in c:\users\amuly\anaconda3\lib\site-package
s (from transformers) (2.32.2)
Requirement already satisfied: tokenizers<0.19,>=0.14 in c:\users\amuly\anaconda3\li
b\site-packages (from transformers) (0.15.1)
Requirement already satisfied: safetensors>=0.3.1 in c:\users\amuly\anaconda3\lib\si
te-packages (from transformers) (0.4.2)
Requirement already satisfied: tqdm>=4.27 in c:\users\amuly\anaconda3\lib\site-packa
ges (from transformers) (4.66.4)
Requirement already satisfied: fsspec>=2023.5.0 in c:\users\amuly\anaconda3\lib\site
-packages (from huggingface-hub<1.0,>=0.19.3->transformers) (2024.2.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in c:\users\amuly\anaconda
3\lib\site-packages (from huggingface-hub<1.0,>=0.19.3->transformers) (4.11.0)
Requirement already satisfied: colorama in c:\users\amuly\anaconda3\lib\site-package
s (from tqdm>=4.27->transformers) (0.4.6)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\amuly\anaconda3
\lib\site-packages (from requests->transformers) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\amuly\anaconda3\lib\site-pac
kages (from requests->transformers) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\amuly\anaconda3\lib\si
te-packages (from requests->transformers) (2.2.2)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\amuly\anaconda3\lib\si
te-packages (from requests->transformers) (2024.6.2)
Requirement already satisfied: ollama in c:\users\amuly\anaconda3\lib\site-packages
(0.3.1)
Requirement already satisfied: httpx<0.28.0,>=0.27.0 in c:\users\amuly\anaconda3\lib
\site-packages (from ollama) (0.27.0)
Requirement already satisfied: anyio in c:\users\amuly\anaconda3\lib\site-packages
(from httpx<0.28.0,>=0.27.0->ollama) (4.2.0)
Requirement already satisfied: certifi in c:\users\amuly\anaconda3\lib\site-packages
(from httpx<0.28.0,>=0.27.0->ollama) (2024.6.2)
Requirement already satisfied: httpcore==1.* in c:\users\amuly\anaconda3\lib\site-pa
ckages (from httpx<0.28.0,>=0.27.0->ollama) (1.0.2)
Requirement already satisfied: idna in c:\users\amuly\anaconda3\lib\site-packages (f
rom httpx<0.28.0,>=0.27.0->ollama) (3.7)
Requirement already satisfied: sniffio in c:\users\amuly\anaconda3\lib\site-packages
(from httpx<0.28.0,>=0.27.0->ollama) (1.3.0)
Requirement already satisfied: h11<0.15,>=0.13 in c:\users\amuly\anaconda3\lib\site-
packages (from httpcore==1.*->httpx<0.28.0,>=0.27.0->ollama) (0.14.0)
```

In [3]:

```
# Cell 1: Imports and Initial Setup
import pandas as pd
import numpy as np
import panel as pn
import plotly.express as px
import plotly.graph_objects as go
from datetime import datetime
from sklearn.preprocessing import MinMaxScaler, StandardScaler
from sklearn.ensemble import GradientBoostingClassifier
from xgboost import XGBClassifier
from catboost import CatBoostClassifier
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import GRU, Dense, Dropout, Input, BatchNormalization,
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.model_selection import StratifiedKFold, GridSearchCV, train_test_split
```

```python
import matplotlib.pyplot as plt
from imblearn.over_sampling import SMOTE
from transformers import GPT2LMHeadModel, GPT2Tokenizer
import warnings
import logging
from transformers import pipeline
from langchain_community.llms import Ollama
from sklearn.ensemble import RandomForestClassifier, IsolationForest
from sklearn.svm import SVC


# Initializing Panel extension
pn.extension('plotly')

# Initializing GPT-2 model, tokenizer and ollama model
model_name = "gpt2"
tokenizer = GPT2Tokenizer.from_pretrained(model_name)
gpt_model = GPT2LMHeadModel.from_pretrained(model_name)
ollama_llm = Ollama(model="llama2")  # Using Ollama for categorization

# Suppress specific warnings and logging
warnings.filterwarnings("ignore", message="Setting `pad_token_id` to `eos_token_id`
logging.getLogger("transformers").setLevel(logging.ERROR)

# Explicitly set pad_token_id to eos_token_id
tokenizer.pad_token_id = tokenizer.eos_token_id
```

```
C:\Users\amuly\anaconda3\Lib\site-packages\huggingface_hub\file_download.py:1132: Fu
tureWarning: `resume_download` is deprecated and will be removed in version 1.0.0. D
ownloads always resume when possible. If you want to force a new download, use `forc
e_download=True`.
  warnings.warn(
```

In [5]:

```python
# Cell 2: Controlled Data and Helper Functions(used for creation of categories )
controlled_data = {
    "Groceries": ["Supermarket", "Grocery Store", "Food Market", "Farmer's Market"]
    "Rent": ["Rent Payment", "Mortgage Payment"],
    "Utilities": ["Electric Bill", "Water Bill", "Gas Bill", "Internet Bill", "Phon
    "Entertainment": ["Cinema", "Concert", "Theater", "Streaming Service"],
    "Miscellaneous": ["Random Purchase", "Miscellaneous Expense", "Gift", "Donation
    "Salary": ["Monthly Salary", "Bonus", "Freelance Payment"],
    "Transportation": ["Bus Fare", "Train Ticket", "Fuel", "Taxi"],
    "Dining": ["Restaurant", "Cafe", "Fast Food"],
    "Health": ["Gym Membership", "Doctor Visit", "Pharmacy"]
}

# Fallback function
def generate_fallback_transaction(date):
    category = np.random.choice(list(controlled_data.keys()))
    description = np.random.choice(controlled_data[category])
    amount = np.random.uniform(2000, 5000) if category == "Salary" else np.random.u
    expense_income = "Income" if category == "Salary" else "Expense"
    return {
        "Date": date,
        "Description": description,
        "Category": category,
        "Expense/Income": expense_income,
        "Amount": amount
    }

# Use Ollama for categorizing transactions
def categorize_transactions(transaction_names, llm):
    prompt = (
        "Can you add an appropriate category to the following expenses? "
        "For example: Spotify AB by Adyen - Entertainment, Beta Boulders Ams Amster
        "Categories should be less than 4 words. "
        + transaction_names
    )
    response = llm.invoke(prompt)
    response = response.split('\n')
    categories_df = pd.DataFrame({'Transaction vs category': response})
    categories_df[['Transaction', 'Category']] = categories_df['Transaction vs cate
    return categories_df.dropna()
```

In [7]:

```
# Cell 3: Generate Synthetic Transaction Data
def generate_synthetic_transaction_data_llm(start_date, end_date, num_samples=500,
    date_range = pd.date_range(start_date, end_date, periods=num_samples)
    transaction_data = []
    unique_transactions = []

    for date in date_range:
        for attempt in range(max_retries):
            # Generate GPT response
            prompt = f"Generate a synthetic transaction for date {date.date()} with
            inputs = tokenizer.encode(prompt, return_tensors="pt")
            attention_mask = inputs.ne(tokenizer.pad_token_id)
            outputs = gpt_model.generate(inputs, max_length=50, num_return_sequence
            response_text = tokenizer.decode(outputs[0], skip_special_tokens=True).

            # Attempt to parse the GPT response
            description, amount = response_text.rsplit(' ', 1)
            try:
                amount = float(amount)
            except ValueError:
                amount = None

            if description and amount:
                unique_transactions.append(description)
                category_df = categorize_transactions(description, ollama_llm)
                category = category_df['Category'].values[0] if not category_df.emp
                expense_income = "Income" if category == "Salary" else "Expense"
                transaction_data.append({
                    "Date": date,
                    "Description": description,
                    "Category": category,
                    "Expense/Income": expense_income,
                    "Amount": amount
                })
                break  # Exit retry loop on success
            else:
                continue  # Try again without printing
        else:
            # Fallback if all attempts fail
            transaction_data.append(generate_fallback_transaction(date))

    return pd.DataFrame(transaction_data)

# Generate synthetic transaction data
df = generate_synthetic_transaction_data_llm('2022-01-01', '2023-12-31', num_sample

# Ensuring valid transactions were generated
if not df.empty:
    df['Date'] = pd.to_datetime(df['Date'])
    print(df)
else:
    print("No valid transactions generated.")
```

```
                             Date            Description       Category  \
0   2022-01-01 00:00:00.000000000  Miscellaneous Expense  Miscellaneous
1   2022-01-02 11:03:43.647294589            Supermarket      Groceries
2   2022-01-03 22:07:27.294589178      Freelance Payment         Salary
3   2022-01-05 09:11:10.941883767             Restaurant         Dining
4   2022-01-06 20:14:54.589178356             Phone Bill      Utilities
..                            ...                    ...            ...
495 2023-12-25 03:45:05.410821640           Rent Payment           Rent
496 2023-12-26 14:48:49.058116232                   Fuel  Transportation
497 2023-12-28 01:52:32.705410816             Restaurant         Dining
498 2023-12-29 12:56:16.352705408      Freelance Payment         Salary
499 2023-12-31 00:00:00.000000000                   Cafe         Dining

    Expense/Income       Amount
0          Expense   266.603412
1          Expense   250.973427
2           Income  4517.710250
3          Expense   106.381042
4          Expense   372.773355
..             ...          ...
495        Expense   288.385089
496        Expense   363.677244
497        Expense   181.418857
498         Income  2438.125025
499        Expense   216.705536

[500 rows x 5 columns]
```

In [8]:

```
#eda
```

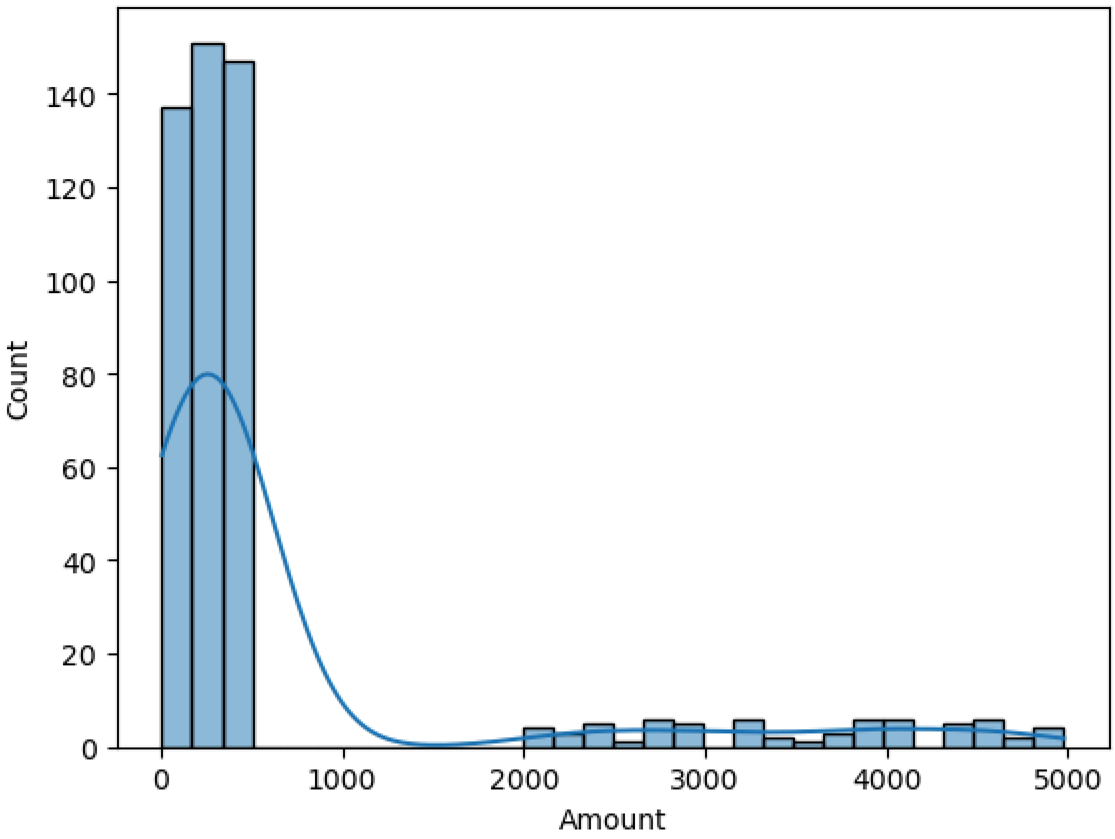
In [9]:

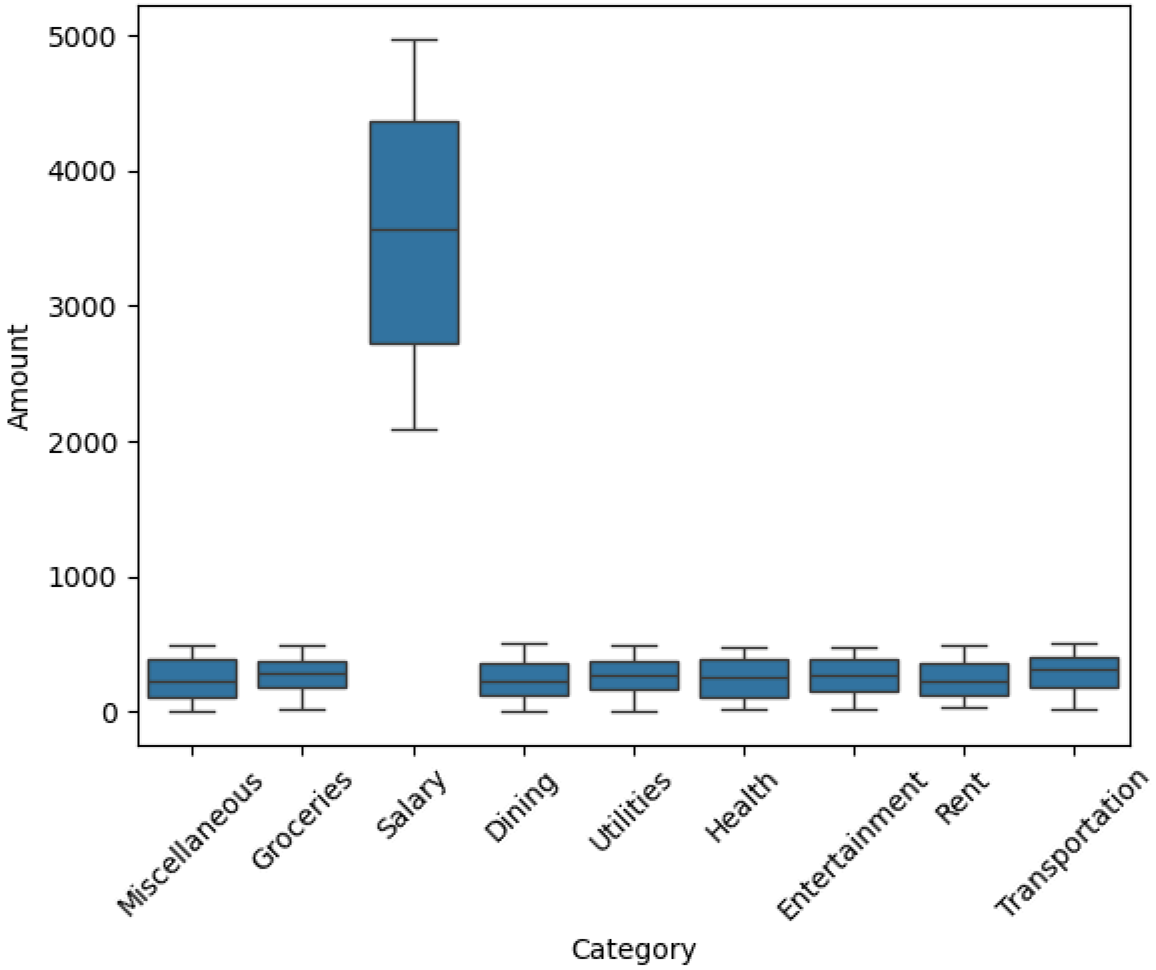
```
# Summary statistics for numerical features
print(df.describe())
```

```
                      Date       Amount
count                  500   500.000000
mean   2022-12-31 12:00:00   679.405162
min    2022-01-01 00:00:00     5.252022
25%    2022-07-02 06:00:00   158.859894
50%    2022-12-31 12:00:00   299.697209
75%    2023-07-01 18:00:00   424.555811
max    2023-12-31 00:00:00  4978.245703
std                    NaN  1144.821433
```

In [10]:

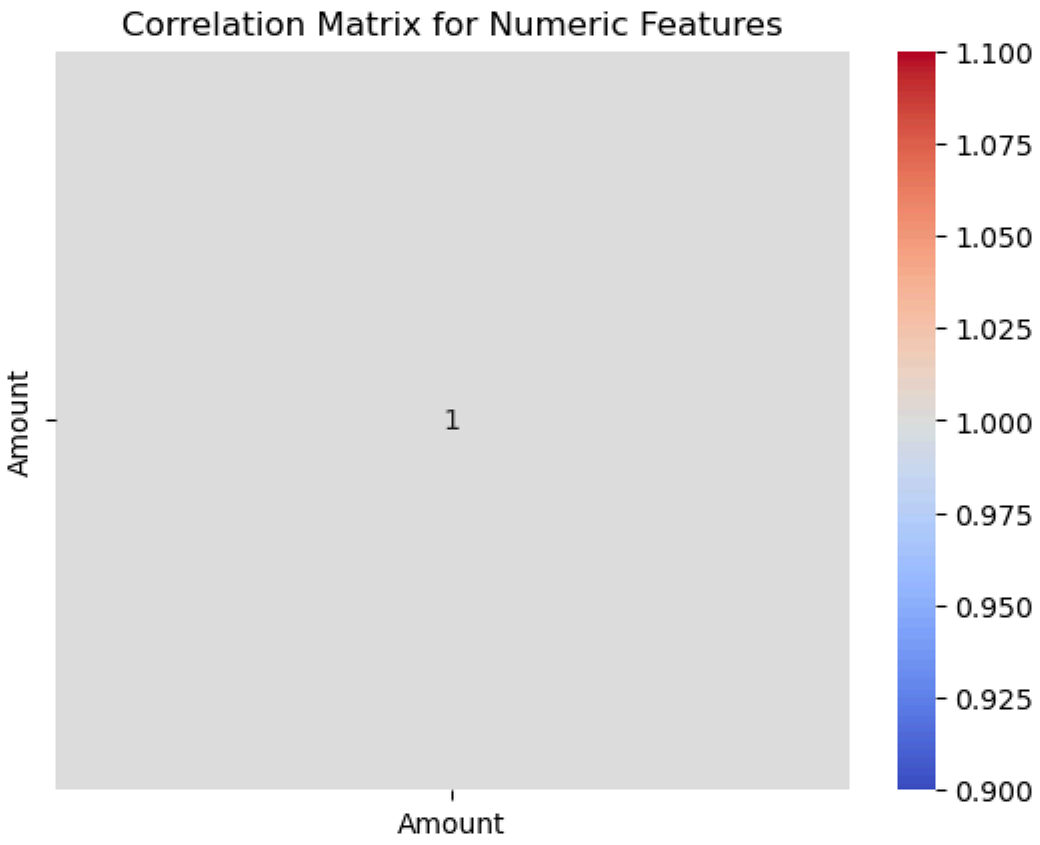
```
#Visualization of Data Distributions
import seaborn as sns
import matplotlib.pyplot as plt

# Histogram of amounts
sns.histplot(df['Amount'], bins=30, kde=True)
plt.show()

# Box plot for transaction amounts
sns.boxplot(x=df['Category'], y=df['Amount'])
plt.xticks(rotation=45)
plt.show()
```





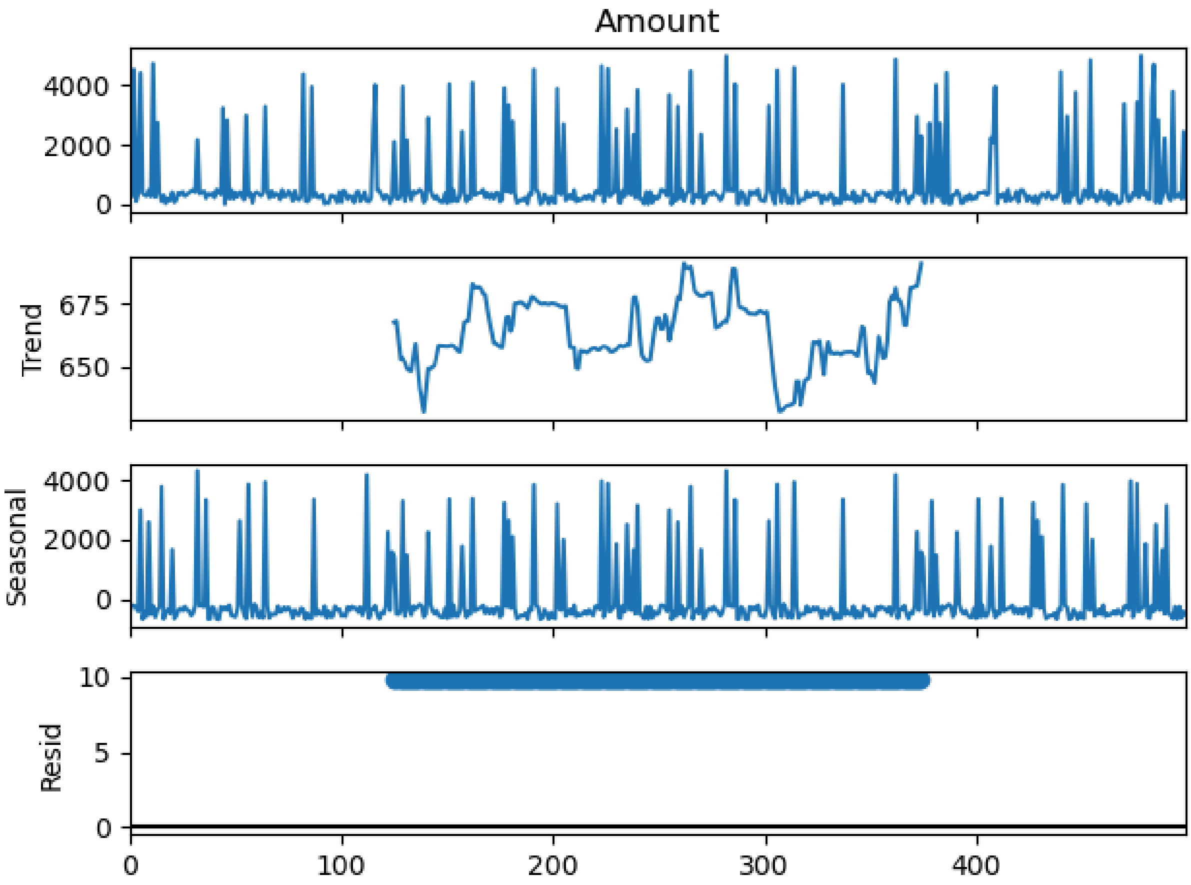
In [11]:

```
#Correlation Analysis (Understand how different numeric features correlate with eac
#and dimensionality reduction.)
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Assuming 'df' is your DataFrame and it includes categorical data

# Convert categorical variable to numeric using one-hot encoding if it makes sense
if 'Category' in df.columns:
    df = pd.get_dummies(df, columns=['Category'], drop_first=True)

# Ensure the DataFrame only contains numeric data
numeric_df = df.select_dtypes(include=[np.number])

# Compute the correlation matrix
correlation_matrix = numeric_df.corr()

# Generate a heatmap
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix for Numeric Features')
plt.show()
```



In [12]:

```
from statsmodels.tsa.seasonal import seasonal_decompose
import matplotlib.pyplot as plt

if len(df['Amount']) >= 500:  # Assuming you have at least 500 data points
    result = seasonal_decompose(df['Amount'], model='additive', period=int(len(df['
    result.plot()
    plt.show()
else:
    print("Not enough data points to perform decomposition.")
```



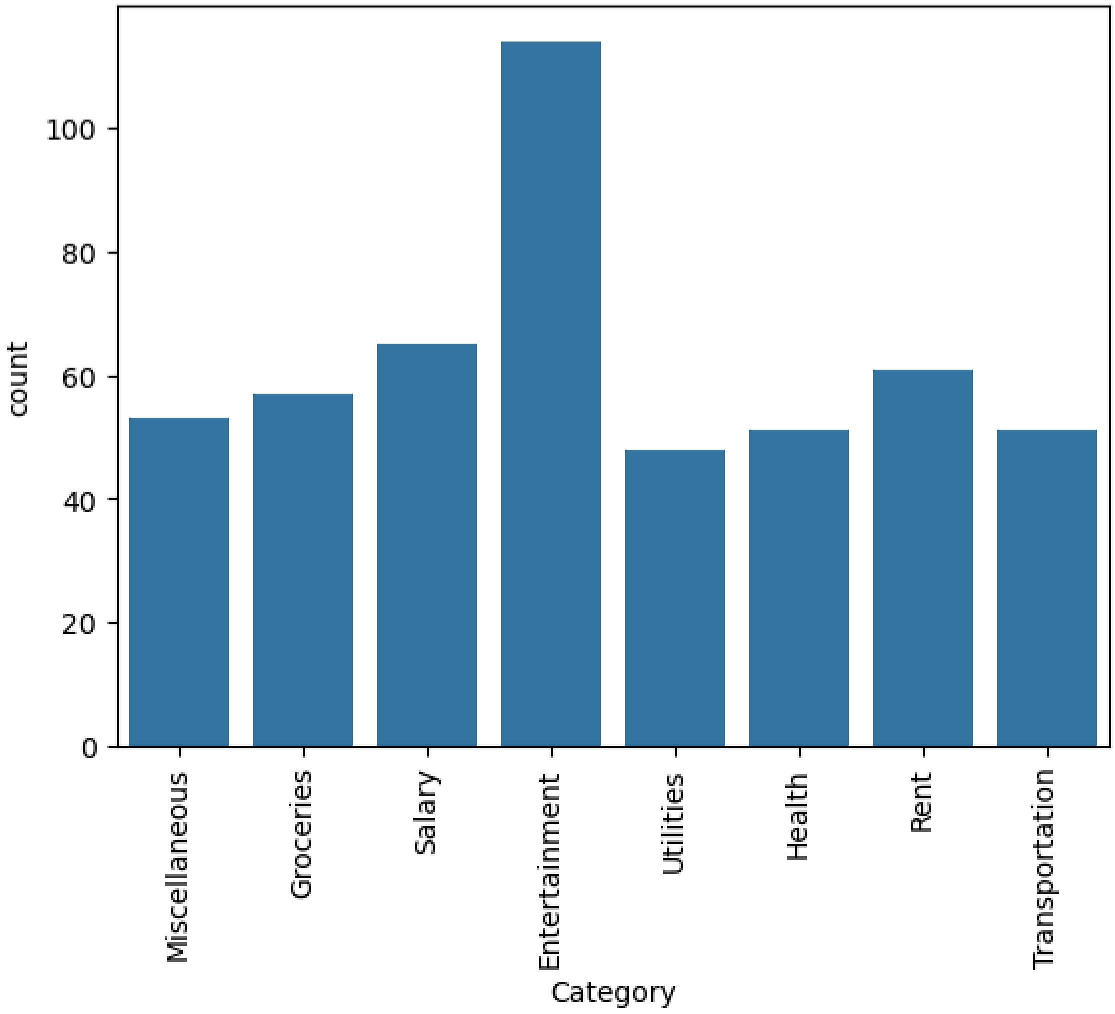
In [13]:

```
# Creating a new column 'Category' by checking which category columns are True
category_columns = [col for col in df.columns if 'Category_' in col]
df['Category'] = df[category_columns].idxmax(axis=1).str.replace('Category_', '')

# Display the DataFrame to verify the new column
print(df[['Category']].head())
```

```
        Category
0  Miscellaneous
1      Groceries
2         Salary
3  Entertainment
4      Utilities
```

In [14]:

```
import seaborn as sns
import matplotlib.pyplot as plt

# Count plot for categories
sns.countplot(x='Category', data=df)
plt.xticks(rotation=90)  # Rotate labels for better readability
plt.show()
```



In [15]:

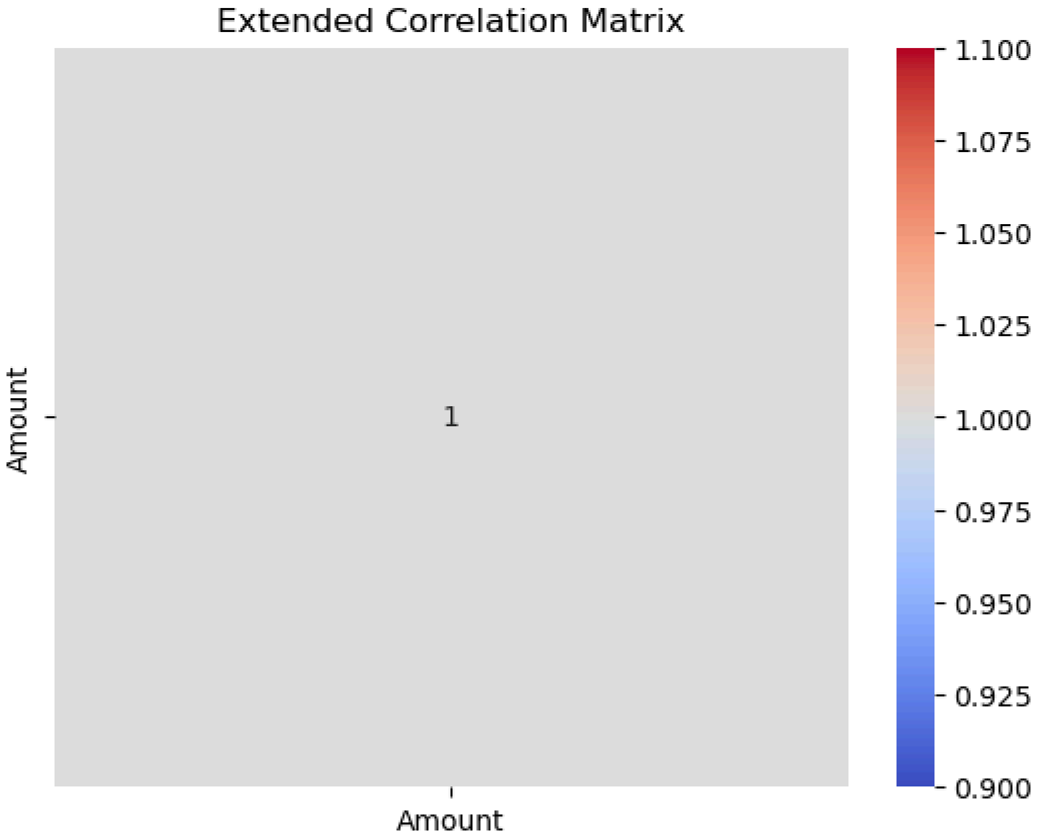
```
# Check for missing values
print(df.isnull().sum())

# Check for duplicates
print(df.duplicated().sum())
```

```
Date                          0
Description                   0
Expense/Income                0
Amount                        0
Category_Entertainment        0
Category_Groceries            0
Category_Health               0
Category_Miscellaneous        0
Category_Rent                 0
Category_Salary               0
Category_Transportation       0
Category_Utilities            0
Category                      0
dtype: int64
0
```

In [16]:

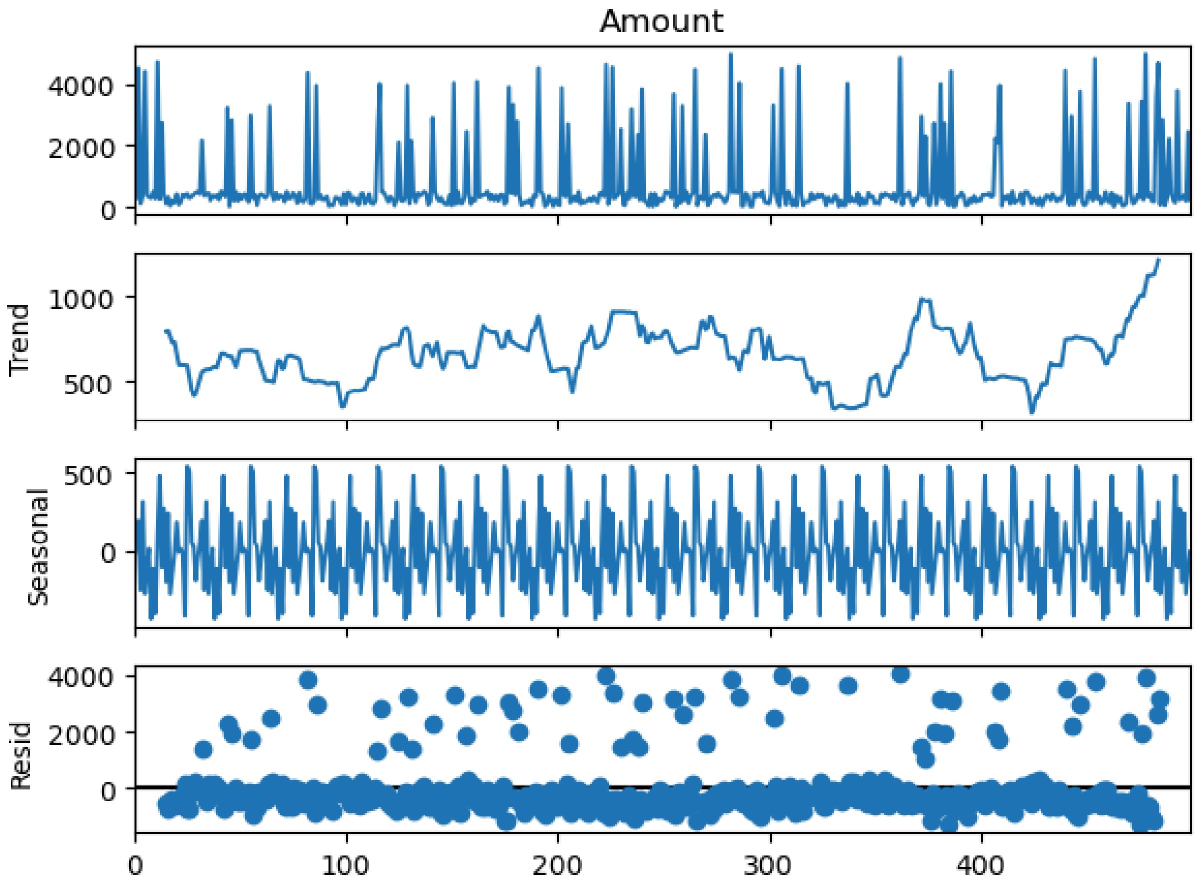
```
# Extended correlation matrix
numeric_df = df.select_dtypes(include=[np.number])
correlation_matrix = numeric_df.corr()
```

```
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Extended Correlation Matrix')
plt.show()
```



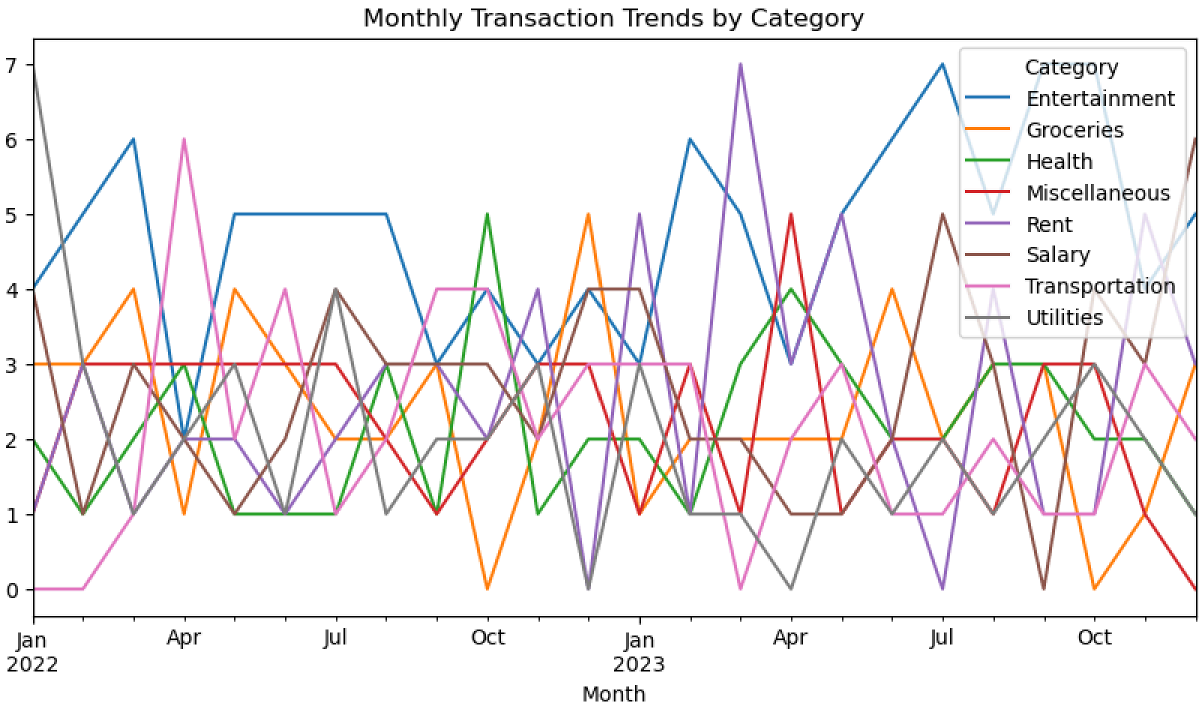
In [17]:

```
#Adjust Seasonal Decomposition Parameters
from statsmodels.tsa.seasonal import seasonal_decompose
result = seasonal_decompose(df['Amount'], model='additive', period=30)  # Monthly c
result.plot()
plt.show()
```



In [18]:

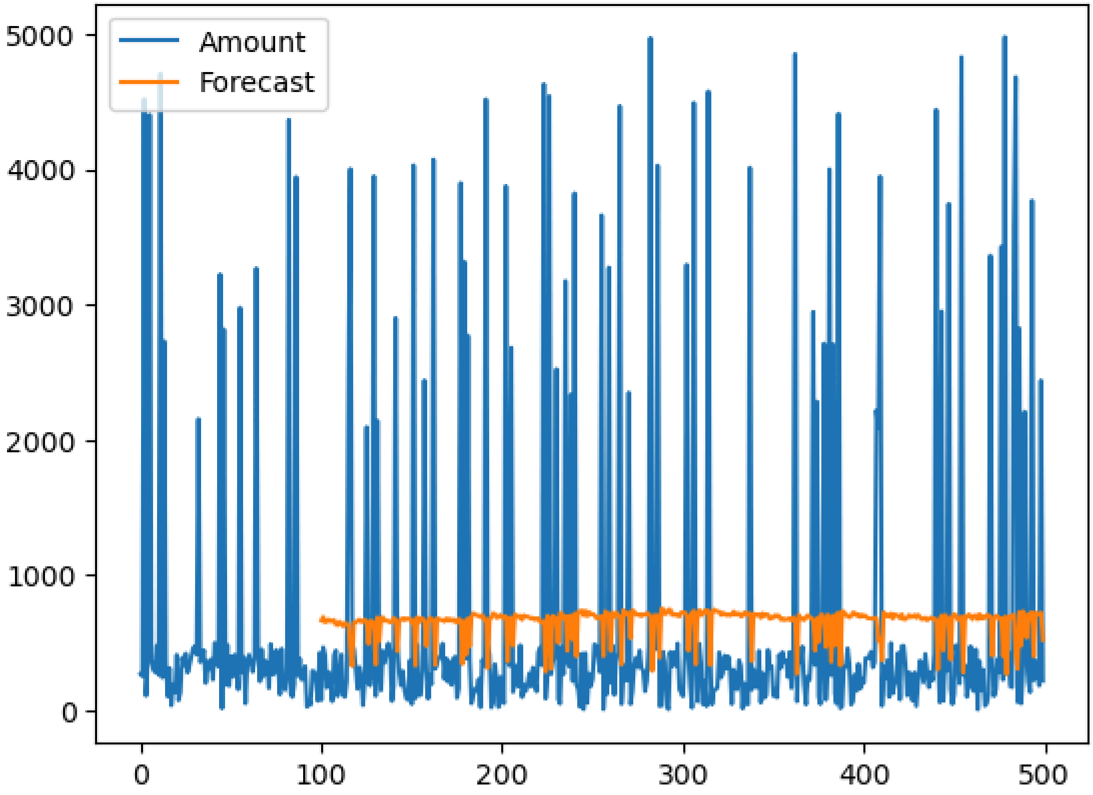
```
df['Month'] = df['Date'].dt.to_period('M')
category_trends = df.groupby(['Month', 'Category']).size().unstack(fill_value=0)
category_trends.plot(kind='line', figsize=(10, 5))
plt.title('Monthly Transaction Trends by Category')
plt.show()
```



In [19]:

```
from statsmodels.tsa.arima.model import ARIMA
model = ARIMA(df['Amount'], order=(1, 1, 1))
result = model.fit()
df['Forecast'] = result.predict(start=100, end=len(df), typ='levels')
df[['Amount', 'Forecast']].plot()
plt.show()
```

```
C:\Users\amuly\anaconda3\Lib\site-packages\statsmodels\tsa\statespace\representatio
n.py:374: FutureWarning: Unknown keyword arguments: dict_keys(['typ']).Passing unkno
wn keyword arguments will raise a TypeError beginning in version 0.15.
  warnings.warn(msg, FutureWarning)
```



In [20]:

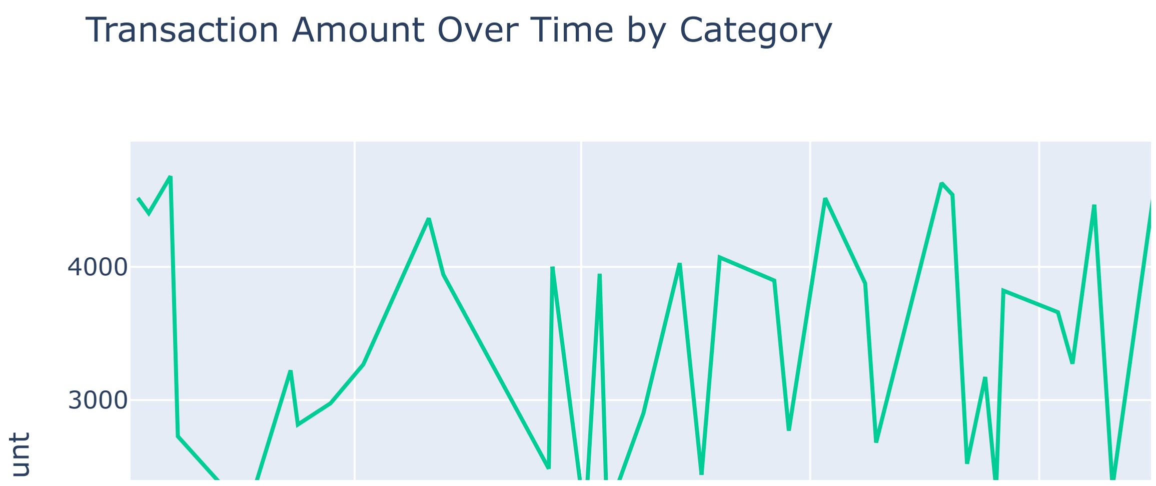
```
#Outlier Treatment
q_low = df['Amount'].quantile(0.01)
q_high = df['Amount'].quantile(0.99)
df['Amount'] = np.clip(df['Amount'], q_low, q_high)
```

In [21]:

```
#Feature engineering
df['DayOfWeek'] = df['Date'].dt.dayofweek
df['Hour'] = df['Date'].dt.hour
```

In [22]:

```
#Considering using Plotly or Bokeh for interactive visualizations that allow deeper
import plotly.express as px
fig = px.line(df, x='Date', y='Amount', color='Category', title='Transaction Amount
fig.show()
```



In [50]:

```
#generate synthetic data
```

In [52]:

```
# Cell 4: Generate Synthetic Stock Data
if not df.empty:
    np.random.seed(42)
    stock_data = pd.DataFrame({
        'Close': np.random.rand(len(df)) * 100 + 100  # Random stock close prices
    }, index=df['Date'])

    # Add Lag Features for Stock Data
    for lag in range(1, 11):
        stock_data[f'Lag{lag}'] = stock_data['Close'].shift(lag)
    stock_data.dropna(inplace=True)

    if not stock_data.empty:
        # Feature Scaling
        scaler = MinMaxScaler()
        scaled_close = scaler.fit_transform(stock_data[['Close']])
        scaled_data = np.hstack([scaled_close, stock_data[[f'Lag{i}' for i in range

        # Sequence generation
        def create_sequences(data, seq_length):
            x, y = [], []
            for i in range(len(data) - seq_length):
                x.append(data[i:i + seq_length])
                y.append(data[i + seq_length, 0])
            return np.array(x), np.array(y)

        seq_length = 30
        x, y = create_sequences(scaled_data, seq_length)

        # Confirm shapes before proceeding
        if len(x) > 0 and len(y) > 0:
            print(f"x shape: {x.shape}")
            print(f"y shape: {y.shape}")

            # Anomaly Detection using Isolation Forest
            iso_forest = IsolationForest(contamination=0.02, random_state=42)
            anomalies = iso_forest.fit_predict(x.reshape(x.shape[0], -1))

            # Filter out anomalies (anomalies are labeled as -1)
            x_filtered = x[anomalies != -1]
            y_filtered = y[anomalies != -1]

            # Split data into training and test sets
            x_train, x_test, y_train, y_test = train_test_split(x_filtered, y_filte

            # Convert y to binary classes based on median
            y_train_class = (y_train > np.median(y_train)).astype(int)
            y_test_class = (y_test > np.median(y_test)).astype(int)

            # Apply SMOTE to handle any class imbalance
            if len(x_train) > 5:  # Ensure there are more samples than neighbors
                smote = SMOTE(random_state=42, k_neighbors=2)
                x_train_resampled, y_train_resampled = smote.fit_resample(
                    x_train.reshape(-1, seq_length * x_train.shape[2]), y_train_cla
                x_train_resampled = x_train_resampled.reshape(-1, seq_length, x_tra
```

```
x shape: (460, 30, 11)
y shape: (460,)
```

In [56]:

```
# Cell 5: LSTM Model with Cross-Validation and Training (Experiment 1)
def create_lstm_model():
    lstm_model = Sequential([
        Input(shape=(seq_length, x_train.shape[2])),
        LSTM(128, return_sequences=True),
        Dropout(0.3),
        LSTM(64, return_sequences=False),
        Dropout(0.3),
        BatchNormalization(),
        Dense(32, activation='relu', kernel_regularizer='l2'),
        Dense(1, activation='sigmoid')
    ])
    lstm_model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accu
    return lstm_model

# Early stopping and learning rate reduction callback
early_stopping = EarlyStopping(monitor='loss', patience=5, restore_best_weights=Tru
reduce_lr = ReduceLROnPlateau(monitor='loss', factor=0.5, patience=3, min_lr=0.0000
# Check the minimum number of samples in each class
min_samples_per_class = min(np.bincount(y_train_resampled))

# Set the number of splits to the minimum of 5 or the minimum number of samples per
n_splits = min(5, min_samples_per_class)

# Proceed with cross-validation only if the number of splits is at least 2
if n_splits >= 2:
    kfold = StratifiedKFold(n_splits=n_splits, shuffle=True, random_state=42)
    lstm_cv_scores = []

    for train, val in kfold.split(x_train_resampled, y_train_resampled):
        lstm_model = create_lstm_model()
        history = lstm_model.fit(x_train_resampled[train], y_train_resampled[train]
                validation_data=(x_train_resampled[val], y_train_resampled[val]),
                epochs=50, batch_size=32, verbose=1, callbacks=[early_stopping, red
        scores = lstm_model.evaluate(x_train_resampled[val], y_train_resampled[val]
        lstm_cv_scores.append(scores[1])  # Append accuracy

    print(f"LSTM Cross-Validation Accuracy: {np.mean(lstm_cv_scores)}")
else:
    print("Not enough samples to perform cross-validation.")

# Train final LSTM Model with reduced learning rate
lstm_model = create_lstm_model()
history = lstm_model.fit(x_train_resampled, y_train_resampled, epochs=50, batch_siz

# Predict using LSTM Model
lstm_class_prob = lstm_model.predict(x_test)
lstm_class = (lstm_class_prob > 0.5).astype(int)
```

```
Epoch 1/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 10s 172ms/step - accuracy: 0.4869 - loss: 1.2768 - val_accu
racy: 0.5417 - val_loss: 1.1177 - learning_rate: 0.0010
Epoch 2/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.5524 - loss: 1.1986 - val_accura
cy: 0.5000 - val_loss: 1.1094 - learning_rate: 0.0010
Epoch 3/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.5046 - loss: 1.1467 - val_accura
cy: 0.5000 - val_loss: 1.0990 - learning_rate: 0.0010
Epoch 4/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.5482 - loss: 1.1421 - val_accura
cy: 0.5000 - val_loss: 1.0819 - learning_rate: 0.0010
Epoch 5/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.5005 - loss: 1.1189 - val_accura
cy: 0.5000 - val_loss: 1.0715 - learning_rate: 0.0010
Epoch 6/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.4582 - loss: 1.1429 - val_accura
cy: 0.5000 - val_loss: 1.0598 - learning_rate: 0.0010
Epoch 7/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.5577 - loss: 1.0904 - val_accura
cy: 0.5000 - val_loss: 1.0572 - learning_rate: 0.0010
Epoch 8/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.5553 - loss: 1.0441 - val_accura
cy: 0.5000 - val_loss: 1.0433 - learning_rate: 0.0010
Epoch 9/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.5021 - loss: 1.0508 - val_accura
cy: 0.5000 - val_loss: 1.0313 - learning_rate: 0.0010
Epoch 10/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.5625 - loss: 1.0341 - val_accura
cy: 0.5000 - val_loss: 1.0259 - learning_rate: 0.0010
Epoch 11/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.6077 - loss: 1.0027 - val_accura
cy: 0.5000 - val_loss: 1.0137 - learning_rate: 0.0010
Epoch 12/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.5486 - loss: 1.0257 - val_accura
cy: 0.5000 - val_loss: 1.0016 - learning_rate: 0.0010
Epoch 13/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.5518 - loss: 0.9946 - val_accura
cy: 0.5000 - val_loss: 0.9910 - learning_rate: 0.0010
Epoch 14/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.5264 - loss: 0.9941 - val_accura
cy: 0.5000 - val_loss: 0.9819 - learning_rate: 0.0010
Epoch 15/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.5479 - loss: 0.9783 - val_accura
cy: 0.5000 - val_loss: 0.9793 - learning_rate: 0.0010
Epoch 16/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 45ms/step - accuracy: 0.4964 - loss: 1.0018 - val_accura
cy: 0.5000 - val_loss: 0.9719 - learning_rate: 0.0010
Epoch 17/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.5730 - loss: 0.9366 - val_accura
cy: 0.5000 - val_loss: 0.9584 - learning_rate: 0.0010
Epoch 18/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.4636 - loss: 1.0000 - val_accura
cy: 0.5000 - val_loss: 0.9493 - learning_rate: 0.0010
Epoch 19/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.6114 - loss: 0.9148 - val_accura
```

```
cy: 0.5000 - val_loss: 0.9414 - learning_rate: 0.0010
Epoch 20/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.5431 - loss: 0.9342 - val_accura
cy: 0.5000 - val_loss: 0.9344 - learning_rate: 0.0010
Epoch 21/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.4987 - loss: 0.9398 - val_accura
cy: 0.5000 - val_loss: 0.9265 - learning_rate: 0.0010
Epoch 22/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.4878 - loss: 0.9465 - val_accura
cy: 0.5000 - val_loss: 0.9198 - learning_rate: 0.0010
Epoch 23/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.5272 - loss: 0.9245 - val_accura
cy: 0.5000 - val_loss: 0.9171 - learning_rate: 5.0000e-04
Epoch 24/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.4547 - loss: 0.9287 - val_accura
cy: 0.5000 - val_loss: 0.9141 - learning_rate: 5.0000e-04
Epoch 1/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 10s 178ms/step - accuracy: 0.5193 - loss: 1.1963 - val_accu
racy: 0.5000 - val_loss: 1.1181 - learning_rate: 0.0010
Epoch 2/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.4848 - loss: 1.2148 - val_accura
cy: 0.5000 - val_loss: 1.1027 - learning_rate: 0.0010
Epoch 3/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.5516 - loss: 1.1518 - val_accura
cy: 0.5000 - val_loss: 1.0886 - learning_rate: 0.0010
Epoch 4/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.4592 - loss: 1.1895 - val_accura
cy: 0.4444 - val_loss: 1.0812 - learning_rate: 0.0010
Epoch 5/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.4914 - loss: 1.1311 - val_accura
cy: 0.5000 - val_loss: 1.0710 - learning_rate: 0.0010
Epoch 1/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 10s 181ms/step - accuracy: 0.5107 - loss: 1.1436 - val_accu
racy: 0.5139 - val_loss: 1.0969 - learning_rate: 0.0010
Epoch 2/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 2s 48ms/step - accuracy: 0.4861 - loss: 1.1303 - val_accura
cy: 0.5278 - val_loss: 1.0819 - learning_rate: 0.0010
Epoch 3/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.6096 - loss: 1.0708 - val_accura
cy: 0.5417 - val_loss: 1.0575 - learning_rate: 0.0010
Epoch 4/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.5547 - loss: 1.0882 - val_accura
cy: 0.5000 - val_loss: 1.0387 - learning_rate: 0.0010
Epoch 5/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.4687 - loss: 1.1124 - val_accura
cy: 0.5000 - val_loss: 1.0276 - learning_rate: 0.0010
Epoch 1/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 11s 196ms/step - accuracy: 0.5176 - loss: 1.3402 - val_accu
racy: 0.5000 - val_loss: 1.1488 - learning_rate: 0.0010
Epoch 2/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 51ms/step - accuracy: 0.5374 - loss: 1.1906 - val_accura
cy: 0.5000 - val_loss: 1.1227 - learning_rate: 0.0010
Epoch 3/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.4936 - loss: 1.2278 - val_accura
cy: 0.5000 - val_loss: 1.1114 - learning_rate: 0.0010
Epoch 4/50
```

```
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.5026 - loss: 1.2207 - val_accura
cy: 0.5000 - val_loss: 1.0862 - learning_rate: 0.0010
Epoch 5/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.5322 - loss: 1.1335 - val_accura
cy: 0.5000 - val_loss: 1.0680 - learning_rate: 0.0010
Epoch 1/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 11s 194ms/step - accuracy: 0.4893 - loss: 1.2167 - val_accu
racy: 0.5000 - val_loss: 1.0961 - learning_rate: 0.0010
Epoch 2/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 50ms/step - accuracy: 0.4861 - loss: 1.2307 - val_accura
cy: 0.5000 - val_loss: 1.0759 - learning_rate: 0.0010
Epoch 3/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.5783 - loss: 1.1022 - val_accura
cy: 0.5000 - val_loss: 1.0557 - learning_rate: 0.0010
Epoch 4/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 52ms/step - accuracy: 0.4597 - loss: 1.1799 - val_accura
cy: 0.5000 - val_loss: 1.0426 - learning_rate: 0.0010
Epoch 5/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.5374 - loss: 1.0890 - val_accura
cy: 0.5139 - val_loss: 1.0305 - learning_rate: 0.0010
LSTM Cross-Validation Accuracy: 0.5027777791023255
Epoch 1/50
12/12 ━━━━━━━━━━━━━━━━━━━━ 12s 65ms/step - accuracy: 0.5077 - loss: 1.3097 - learnin
g_rate: 0.0010
Epoch 2/50
12/12 ━━━━━━━━━━━━━━━━━━━━ 1s 40ms/step - accuracy: 0.5271 - loss: 1.1960 - learning
_rate: 0.0010
Epoch 3/50
12/12 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step - accuracy: 0.4551 - loss: 1.2036 - learning
_rate: 0.0010
Epoch 4/50
12/12 ━━━━━━━━━━━━━━━━━━━━ 1s 44ms/step - accuracy: 0.4822 - loss: 1.1407 - learning
_rate: 0.0010
Epoch 5/50
12/12 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step - accuracy: 0.4990 - loss: 1.1153 - learning
_rate: 0.0010
3/3 ━━━━━━━━━━━━━━━━━━━━ 2s 385ms/step
```

In [57]:

```
# Cell 6: RandomForest and SVM Models with Hyperparameter Tuning (Experiment 1)
# RandomForest model with extended hyperparameter tuning
rf_model = RandomForestClassifier(random_state=42)
rf_param_grid = {
    'n_estimators': [100, 500, 1000],
    'max_depth': [10, 20, 30, 40],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}
rf_grid_search = GridSearchCV(estimator=rf_model, param_grid=rf_param_grid, cv=n_sp

if n_splits >= 2:
    rf_grid_search.fit(x_train_resampled.reshape(x_train_resampled.shape[0], -1), y
    rf_best_model = rf_grid_search.best_estimator_
    rf_predictions = rf_best_model.predict(x_test.reshape(x_test.shape[0], -1))
else:
    print("Not enough samples to perform RandomForest cross-validation.")
```

```python
# SVM model with hyperparameter tuning
svm_model = SVC(probability=True, random_state=42)
svm_param_grid = {
    'C': [0.1, 1, 10, 100, 1000],
    'kernel': ['linear', 'rbf', 'poly', 'sigmoid'],
    'gamma': ['scale', 'auto', 0.001, 0.01, 0.1]
}
svm_grid_search = GridSearchCV(estimator=svm_model, param_grid=svm_param_grid, cv=n

if n_splits >= 2:
    svm_grid_search.fit(x_train_resampled.reshape(x_train_resampled.shape[0], -1),
    svm_best_model = svm_grid_search.best_estimator_
    svm_predictions = svm_best_model.predict(x_test.reshape(x_test.shape[0], -1))
    svm_class_prob = svm_best_model.predict_proba(x_test.reshape(x_test.shape[0], -
else:
    print("Not enough samples to perform SVM cross-validation.")
```

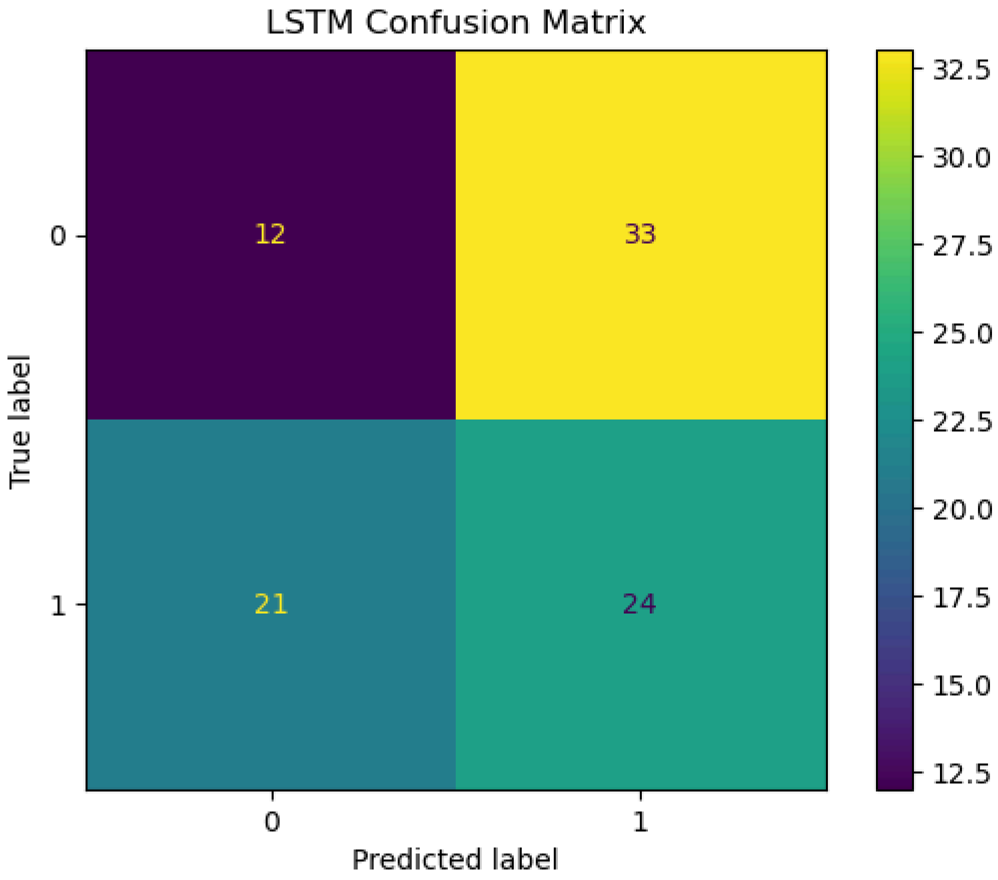
```
Fitting 5 folds for each of 108 candidates, totalling 540 fits
Fitting 5 folds for each of 100 candidates, totalling 500 fits
```

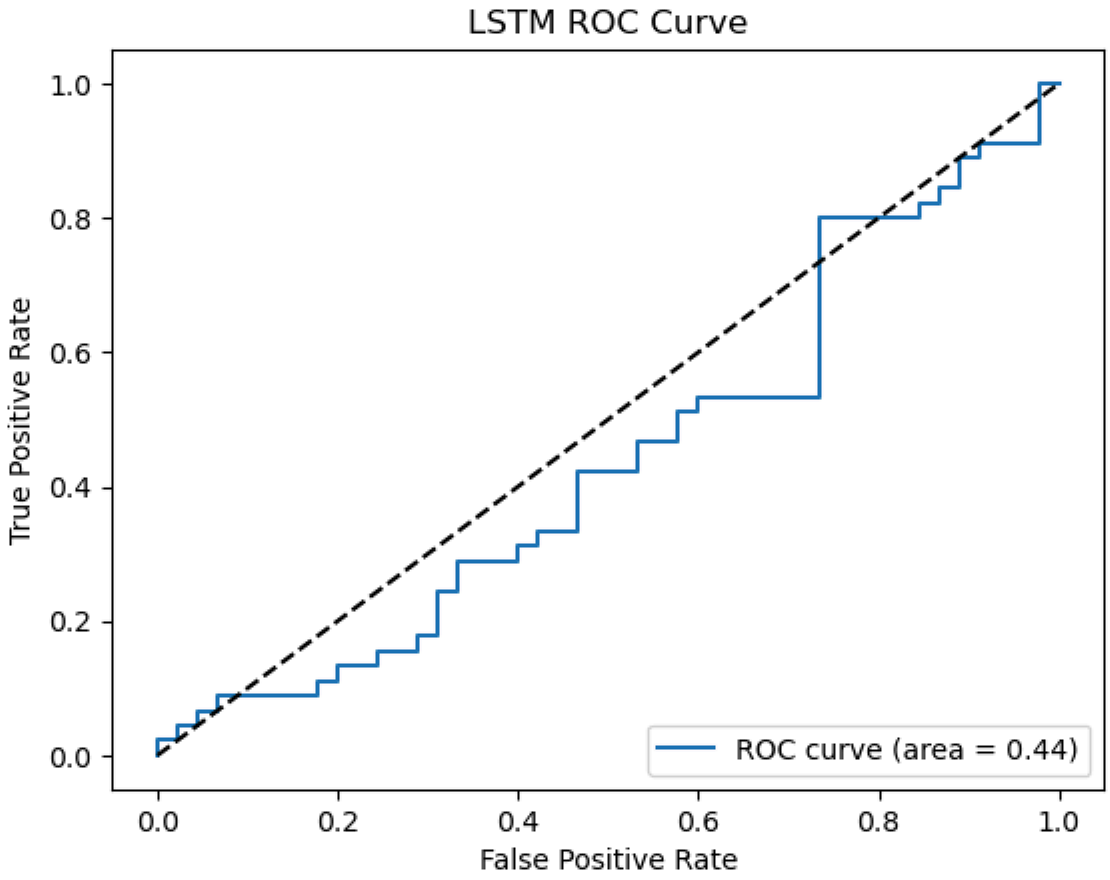
In [58]:

```python
# Cell 7: Metrics and Plots for Experiment 1
def print_metrics_and_plots(y_test_class, predictions, probs, model_name):
    accuracy = accuracy_score(y_test_class, predictions)
    precision = precision_score(y_test_class, predictions)
    recall = recall_score(y_test_class, predictions)
    f1 = f1_score(y_test_class, predictions)
    auc_score = roc_auc_score(y_test_class, probs)
    conf_matrix = confusion_matrix(y_test_class, predictions)

    print(f"{model_name} Classification Metrics:")
    print(f"Accuracy: {accuracy}")
    print(f"Precision: {precision}")
    print(f"Recall: {recall}")
    print(f"F1 Score: {f1}")
    print(f"AUC-ROC: {auc_score}\n")

    # Plot confusion matrix
    disp = ConfusionMatrixDisplay(confusion_matrix=conf_matrix)
    disp.plot()
    plt.title(f"{model_name} Confusion Matrix")
    plt.show()

    # Plot ROC curve
    fpr, tpr, _ = roc_curve(y_test_class, probs)
    plt.figure()
    plt.plot(fpr, tpr, label=f'ROC curve (area = {auc_score:.2f})')
    plt.plot([0, 1], [0, 1], 'k--')
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title(f'{model_name} ROC Curve')
    plt.legend(loc="lower right")
    plt.show()

# LSTM Metrics and Plots
print_metrics_and_plots(y_test_class, lstm_class, lstm_class_prob, "LSTM")
```

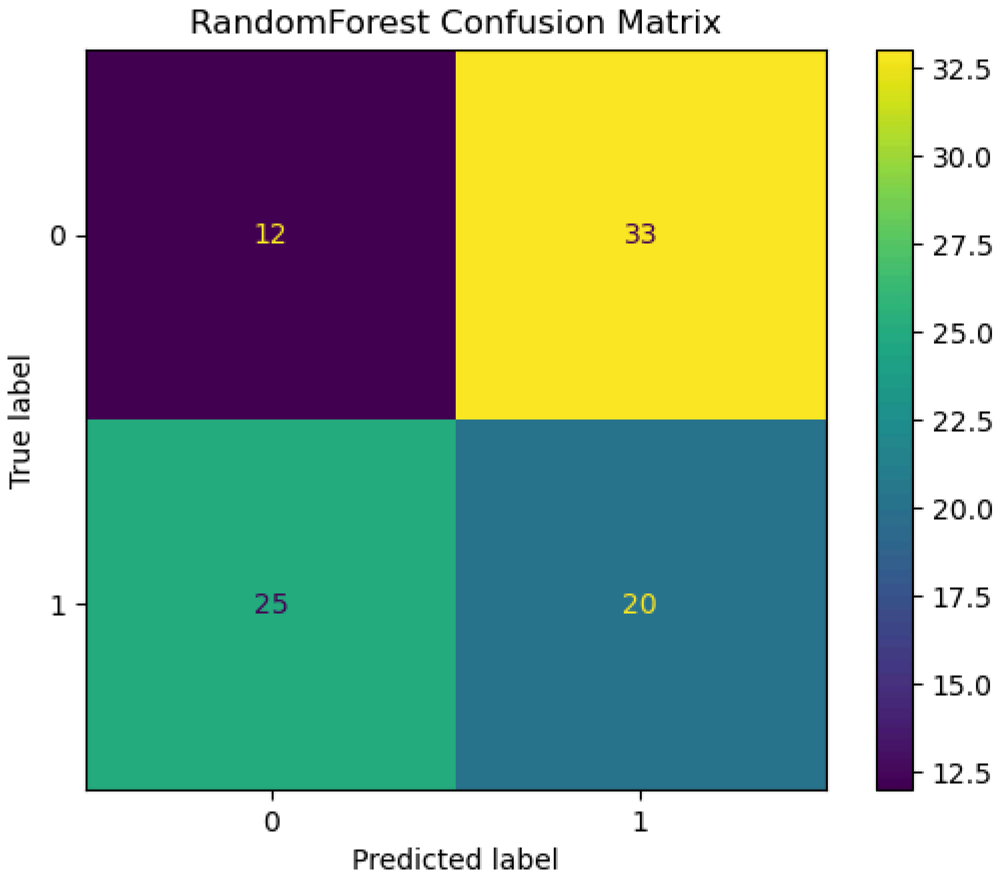
```python
# RandomForest Metrics and Plots
rf_probs = rf_best_model.predict_proba(x_test.reshape(x_test.shape[0], -1))[:, 1]
print_metrics_and_plots(y_test_class, rf_predictions, rf_probs, "RandomForest")

# SVM Metrics and Plots
print_metrics_and_plots(y_test_class, svm_predictions, svm_class_prob, "SVM")
```

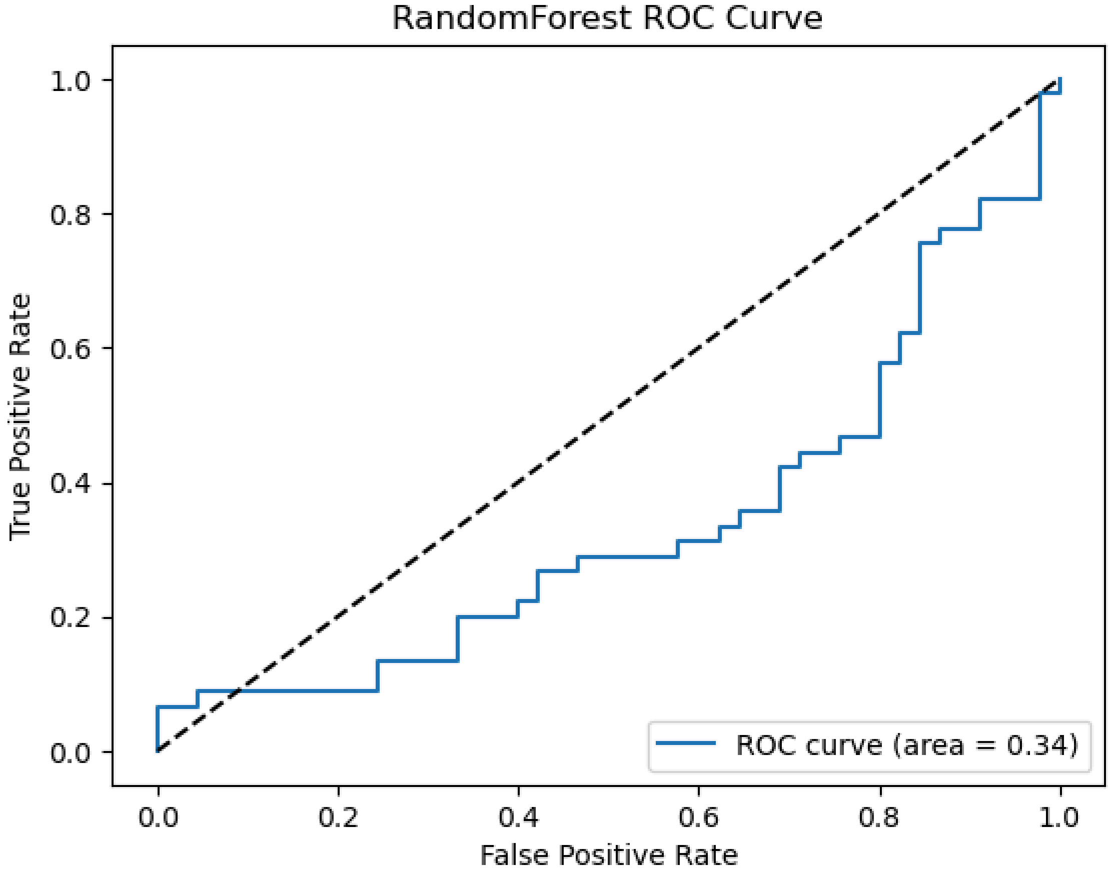
```
LSTM Classification Metrics:
Accuracy: 0.4
Precision: 0.42105263157894735
Recall: 0.5333333333333333
F1 Score: 0.47058823529411764
AUC-ROC: 0.43851851851851853
```



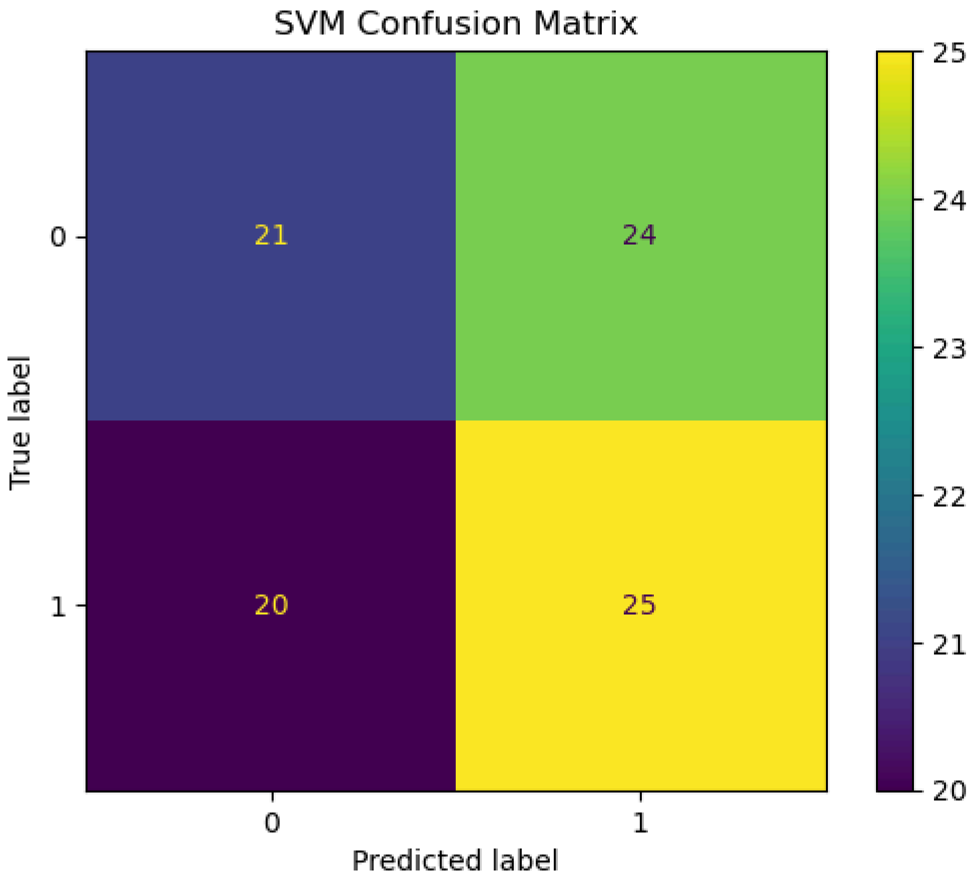


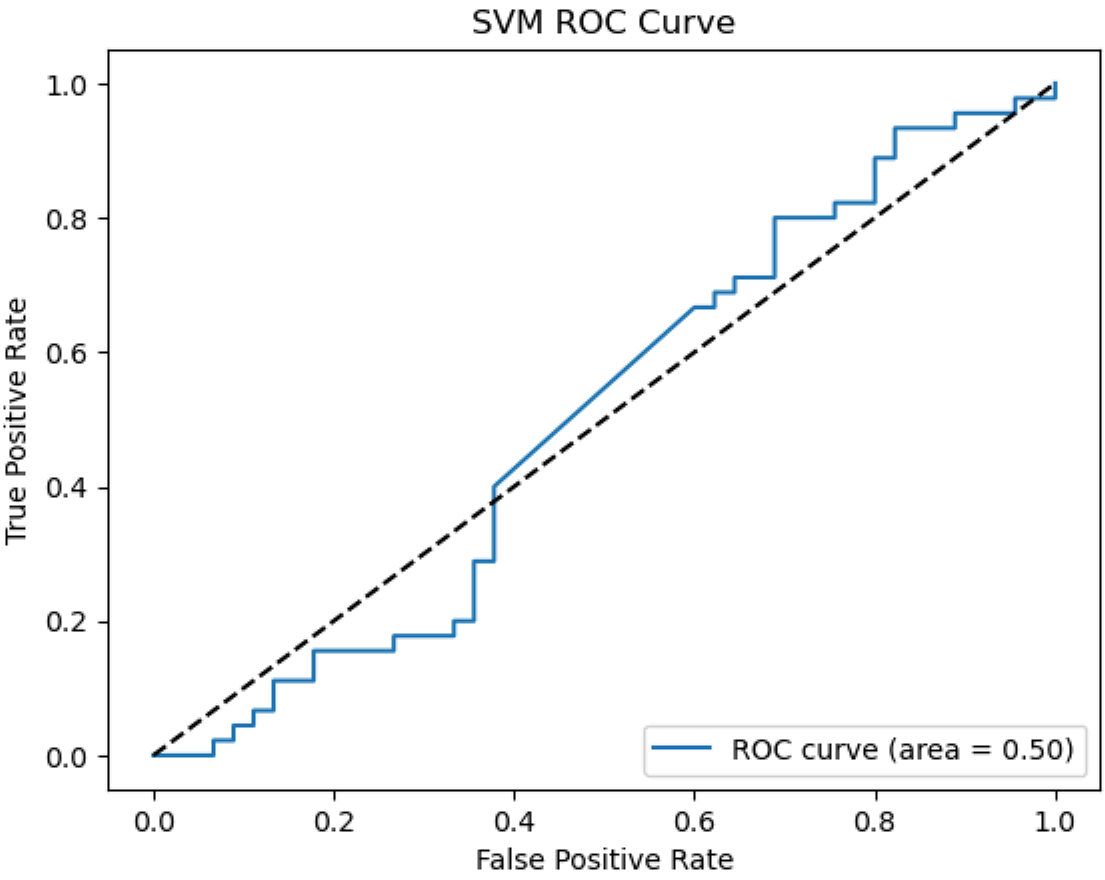
```
RandomForest Classification Metrics:
Accuracy: 0.35555555555555557
Precision: 0.37735849056603776
Recall: 0.4444444444444444
F1 Score: 0.40816326530612246
AUC-ROC: 0.3362962962962963
```


RandomForest Confusion Matrix
True label
Predicted label
0
1
0
1
12
33
25
20
32.5
30.0
27.5
25.0
22.5
20.0
17.5
15.0
12.5


RandomForest ROC Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (area = 0.34)

```
SVM Classification Metrics:
Accuracy: 0.5111111111111111
Precision: 0.5102040816326531
Recall: 0.5555555555555556
F1 Score: 0.5319148936170213
AUC-ROC: 0.5037037037037037
```





In [59]:

```
#experiment 2
```

In [60]:

```
# Cell 8: GRU Model with Cross-Validation and Training (Experiment 2)
def create_gru_model():
    gru_model = Sequential([
        Input(shape=(seq_length, x_train.shape[2])),
        GRU(128, return_sequences=True),
        Dropout(0.3),
        GRU(64, return_sequences=False),
        Dropout(0.3),
        BatchNormalization(),
        Dense(32, activation='relu', kernel_regularizer='l2'),
        Dense(1, activation='sigmoid')
    ])
    gru_model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accur
    return gru_model

# Early stopping and learning rate reduction callback
early_stopping = EarlyStopping(monitor='loss', patience=5, restore_best_weights=Tru
reduce_lr = ReduceLROnPlateau(monitor='loss', factor=0.5, patience=3, min_lr=0.0000

# Check the minimum number of samples in each class
min_samples_per_class = min(np.bincount(y_train_resampled))

# Set the number of splits to the minimum of 5 or the minimum number of samples per
n_splits = min(5, min_samples_per_class)

# Proceed with cross-validation only if the number of splits is at least 2
```

```
if n_splits >= 2:
    kfold = StratifiedKFold(n_splits=n_splits, shuffle=True, random_state=42)
    gru_cv_scores = []

    for train, val in kfold.split(x_train_resampled, y_train_resampled):
        gru_model = create_gru_model()
        history = gru_model.fit(x_train_resampled[train], y_train_resampled[train],
                validation_data=(x_train_resampled[val], y_train_resampled[val]),
                epochs=50, batch_size=32, verbose=1, callbacks=[early_stopping, red
        scores = gru_model.evaluate(x_train_resampled[val], y_train_resampled[val],
        gru_cv_scores.append(scores[1])  # Append accuracy

    print(f"GRU Cross-Validation Accuracy: {np.mean(gru_cv_scores)}")
else:
    print("Not enough samples to perform cross-validation.")

# Train final GRU Model with reduced learning rate
gru_model = create_gru_model()
history = gru_model.fit(x_train_resampled, y_train_resampled, epochs=50, batch_size

# Predict using GRU Model
gru_class_prob = gru_model.predict(x_test)
gru_class = (gru_class_prob > 0.5).astype(int)
```

```
Epoch 1/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 11s 215ms/step - accuracy: 0.4821 - loss: 1.3081 - val_accu
racy: 0.5000 - val_loss: 1.1232 - learning_rate: 0.0010
Epoch 2/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 55ms/step - accuracy: 0.4940 - loss: 1.1533 - val_accura
cy: 0.5000 - val_loss: 1.1076 - learning_rate: 0.0010
Epoch 3/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 61ms/step - accuracy: 0.5293 - loss: 1.1599 - val_accura
cy: 0.5000 - val_loss: 1.1001 - learning_rate: 0.0010
Epoch 4/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 53ms/step - accuracy: 0.4662 - loss: 1.1862 - val_accura
cy: 0.5000 - val_loss: 1.0962 - learning_rate: 0.0010
Epoch 5/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 63ms/step - accuracy: 0.5566 - loss: 1.0966 - val_accura
cy: 0.5000 - val_loss: 1.0688 - learning_rate: 0.0010
Epoch 6/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 54ms/step - accuracy: 0.5686 - loss: 1.0617 - val_accura
cy: 0.5000 - val_loss: 1.0516 - learning_rate: 5.0000e-04
Epoch 7/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 50ms/step - accuracy: 0.5182 - loss: 1.0941 - val_accura
cy: 0.5000 - val_loss: 1.0447 - learning_rate: 5.0000e-04
Epoch 8/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 57ms/step - accuracy: 0.5523 - loss: 1.0664 - val_accura
cy: 0.4583 - val_loss: 1.0342 - learning_rate: 5.0000e-04
Epoch 9/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 66ms/step - accuracy: 0.5720 - loss: 1.0724 - val_accura
cy: 0.5278 - val_loss: 1.0281 - learning_rate: 5.0000e-04
Epoch 10/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 56ms/step - accuracy: 0.4751 - loss: 1.0677 - val_accura
cy: 0.5417 - val_loss: 1.0185 - learning_rate: 5.0000e-04
Epoch 11/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 61ms/step - accuracy: 0.5900 - loss: 1.0200 - val_accura
cy: 0.5278 - val_loss: 1.0130 - learning_rate: 5.0000e-04
Epoch 12/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 55ms/step - accuracy: 0.5385 - loss: 1.0727 - val_accura
cy: 0.5278 - val_loss: 1.0091 - learning_rate: 5.0000e-04
Epoch 13/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 58ms/step - accuracy: 0.6047 - loss: 0.9955 - val_accura
cy: 0.4861 - val_loss: 1.0053 - learning_rate: 5.0000e-04
Epoch 14/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 58ms/step - accuracy: 0.5335 - loss: 1.0423 - val_accura
cy: 0.5000 - val_loss: 0.9985 - learning_rate: 5.0000e-04
Epoch 15/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 57ms/step - accuracy: 0.5673 - loss: 1.0082 - val_accura
cy: 0.5139 - val_loss: 0.9940 - learning_rate: 5.0000e-04
Epoch 16/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 53ms/step - accuracy: 0.5459 - loss: 0.9724 - val_accura
cy: 0.4861 - val_loss: 0.9887 - learning_rate: 5.0000e-04
Epoch 17/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 60ms/step - accuracy: 0.5632 - loss: 1.0168 - val_accura
cy: 0.4861 - val_loss: 0.9819 - learning_rate: 5.0000e-04
Epoch 18/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 52ms/step - accuracy: 0.5895 - loss: 0.9680 - val_accura
cy: 0.5417 - val_loss: 0.9776 - learning_rate: 5.0000e-04
Epoch 19/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 53ms/step - accuracy: 0.5523 - loss: 0.9792 - val_accura
```

```
cy: 0.5417 - val_loss: 0.9741 - learning_rate: 5.0000e-04
Epoch 20/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 49ms/step - accuracy: 0.5434 - loss: 0.9856 - val_accura
cy: 0.5417 - val_loss: 0.9599 - learning_rate: 5.0000e-04
Epoch 21/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 52ms/step - accuracy: 0.5740 - loss: 0.9529 - val_accura
cy: 0.5139 - val_loss: 0.9619 - learning_rate: 5.0000e-04
Epoch 22/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 58ms/step - accuracy: 0.5053 - loss: 1.0087 - val_accura
cy: 0.5556 - val_loss: 0.9599 - learning_rate: 2.5000e-04
Epoch 23/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 62ms/step - accuracy: 0.5200 - loss: 0.9923 - val_accura
cy: 0.5000 - val_loss: 0.9602 - learning_rate: 2.5000e-04
Epoch 1/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 10s 193ms/step - accuracy: 0.4836 - loss: 1.2421 - val_accu
racy: 0.5000 - val_loss: 1.1082 - learning_rate: 0.0010
Epoch 2/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 2s 53ms/step - accuracy: 0.5279 - loss: 1.1587 - val_accura
cy: 0.4722 - val_loss: 1.0758 - learning_rate: 0.0010
Epoch 3/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 61ms/step - accuracy: 0.4635 - loss: 1.1456 - val_accura
cy: 0.5000 - val_loss: 1.0701 - learning_rate: 0.0010
Epoch 4/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 48ms/step - accuracy: 0.5325 - loss: 1.1106 - val_accura
cy: 0.4722 - val_loss: 1.0512 - learning_rate: 0.0010
Epoch 5/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.5751 - loss: 1.0766 - val_accura
cy: 0.4722 - val_loss: 1.0347 - learning_rate: 0.0010
Epoch 1/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 10s 195ms/step - accuracy: 0.5074 - loss: 1.1583 - val_accu
racy: 0.5000 - val_loss: 1.1520 - learning_rate: 0.0010
Epoch 2/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 47ms/step - accuracy: 0.5555 - loss: 1.1106 - val_accura
cy: 0.5000 - val_loss: 1.1484 - learning_rate: 0.0010
Epoch 3/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 47ms/step - accuracy: 0.5478 - loss: 1.0946 - val_accura
cy: 0.5000 - val_loss: 1.1426 - learning_rate: 0.0010
Epoch 4/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 55ms/step - accuracy: 0.5726 - loss: 1.0529 - val_accura
cy: 0.5000 - val_loss: 1.1360 - learning_rate: 0.0010
Epoch 5/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.5566 - loss: 1.0495 - val_accura
cy: 0.5000 - val_loss: 1.1081 - learning_rate: 0.0010
Epoch 1/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 10s 201ms/step - accuracy: 0.5428 - loss: 1.2354 - val_accu
racy: 0.5000 - val_loss: 1.1375 - learning_rate: 0.0010
Epoch 2/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.5433 - loss: 1.1511 - val_accura
cy: 0.5000 - val_loss: 1.1264 - learning_rate: 0.0010
Epoch 3/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 55ms/step - accuracy: 0.4793 - loss: 1.1479 - val_accura
cy: 0.5000 - val_loss: 1.0978 - learning_rate: 0.0010
Epoch 4/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 0.5560 - loss: 1.0981 - val_accura
cy: 0.5000 - val_loss: 1.0726 - learning_rate: 0.0010
Epoch 5/50
```

```
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.5664 - loss: 1.0381 - val_accura
cy: 0.5000 - val_loss: 1.0691 - learning_rate: 0.0010
Epoch 1/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 10s 213ms/step - accuracy: 0.4741 - loss: 1.3204 - val_accu
racy: 0.5139 - val_loss: 1.1191 - learning_rate: 0.0010
Epoch 2/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 56ms/step - accuracy: 0.5142 - loss: 1.1907 - val_accura
cy: 0.4583 - val_loss: 1.0970 - learning_rate: 0.0010
Epoch 3/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 61ms/step - accuracy: 0.5040 - loss: 1.1687 - val_accura
cy: 0.5556 - val_loss: 1.0768 - learning_rate: 0.0010
Epoch 4/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 62ms/step - accuracy: 0.5825 - loss: 1.1323 - val_accura
cy: 0.5417 - val_loss: 1.0627 - learning_rate: 0.0010
Epoch 5/50
9/9 ━━━━━━━━━━━━━━━━━━━━ 1s 61ms/step - accuracy: 0.5329 - loss: 1.1013 - val_accura
cy: 0.5139 - val_loss: 1.0528 - learning_rate: 0.0010
GRU Cross-Validation Accuracy: 0.5111111164093017
Epoch 1/50
12/12 ━━━━━━━━━━━━━━━━━━━━ 11s 50ms/step - accuracy: 0.4797 - loss: 1.2806 - learnin
g_rate: 0.0010
Epoch 2/50
12/12 ━━━━━━━━━━━━━━━━━━━━ 1s 46ms/step - accuracy: 0.5957 - loss: 1.1253 - learning
_rate: 0.0010
Epoch 3/50
12/12 ━━━━━━━━━━━━━━━━━━━━ 1s 43ms/step - accuracy: 0.5258 - loss: 1.1183 - learning
_rate: 0.0010
Epoch 4/50
12/12 ━━━━━━━━━━━━━━━━━━━━ 1s 44ms/step - accuracy: 0.5222 - loss: 1.1051 - learning
_rate: 0.0010
Epoch 5/50
12/12 ━━━━━━━━━━━━━━━━━━━━ 1s 49ms/step - accuracy: 0.5197 - loss: 1.0934 - learning
_rate: 0.0010
3/3 ━━━━━━━━━━━━━━━━━━━━ 2s 409ms/step
```

In [72]:

```python
# Cell 9: XGBoost and CatBoost Models with Hyperparameter Tuning (Experiment 2)
# XGBoost model with hyperparameter tuning
xgb_model = XGBClassifier(random_state=42, use_label_encoder=False)
xgb_param_grid = {
    'n_estimators': [100, 500, 1000],
    'max_depth': [3, 5, 7],
    'learning_rate': [0.01, 0.1, 0.3],
    'subsample': [0.7, 0.8, 1.0]
}
xgb_grid_search = GridSearchCV(estimator=xgb_model, param_grid=xgb_param_grid, cv=n

if n_splits >= 2:
    xgb_grid_search.fit(x_train_resampled.reshape(x_train_resampled.shape[0], -1),
    xgb_best_model = xgb_grid_search.best_estimator_
    xgb_predictions = xgb_best_model.predict(x_test.reshape(x_test.shape[0], -1))
    xgb_class_prob = xgb_best_model.predict_proba(x_test.reshape(x_test.shape[0], -
else:
    print("Not enough samples to perform XGBoost cross-validation.")

# CatBoost model with hyperparameter tuning
catboost_model = CatBoostClassifier(random_state=42, verbose=0)
```

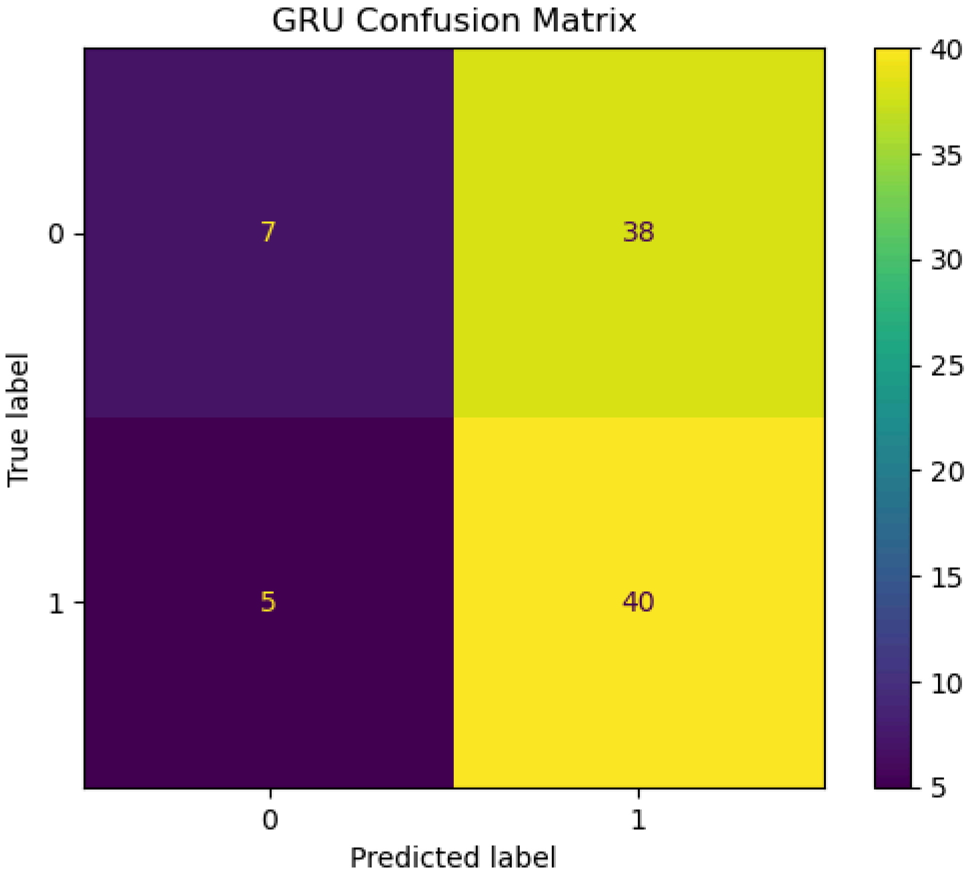
```
catboost_param_grid = {
    'iterations': [500, 1000],
    'depth': [6, 8, 10],
    'learning_rate': [0.01, 0.1, 0.3]
}
catboost_grid_search = GridSearchCV(estimator=catboost_model, param_grid=catboost_p

if n_splits >= 2:
    catboost_grid_search.fit(x_train_resampled.reshape(x_train_resampled.shape[0],
    catboost_best_model = catboost_grid_search.best_estimator_
    catboost_predictions = catboost_best_model.predict(x_test.reshape(x_test.shape[
    catboost_class_prob = catboost_best_model.predict_proba(x_test.reshape(x_test.s
else:
    print("Not enough samples to perform CatBoost cross-validation.")
```

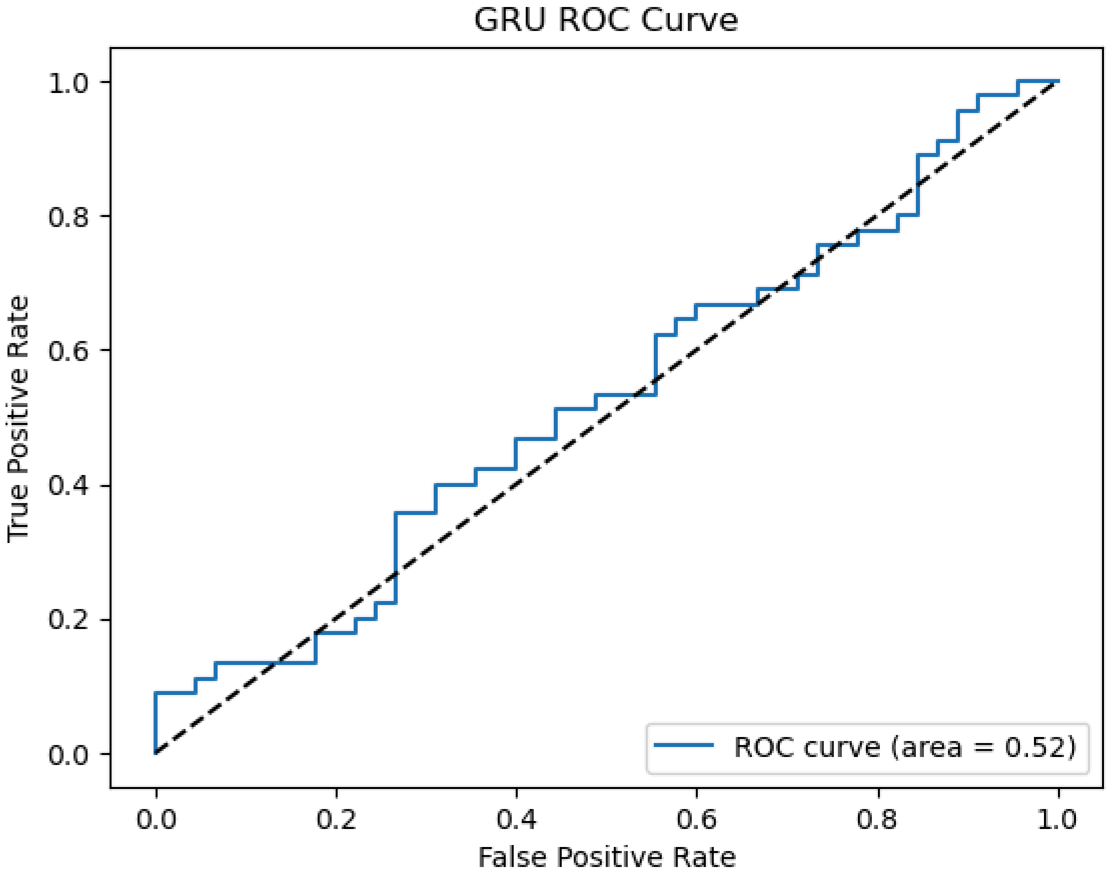
```
Fitting 5 folds for each of 81 candidates, totalling 405 fits
Fitting 5 folds for each of 18 candidates, totalling 90 fits
```

In [73]:

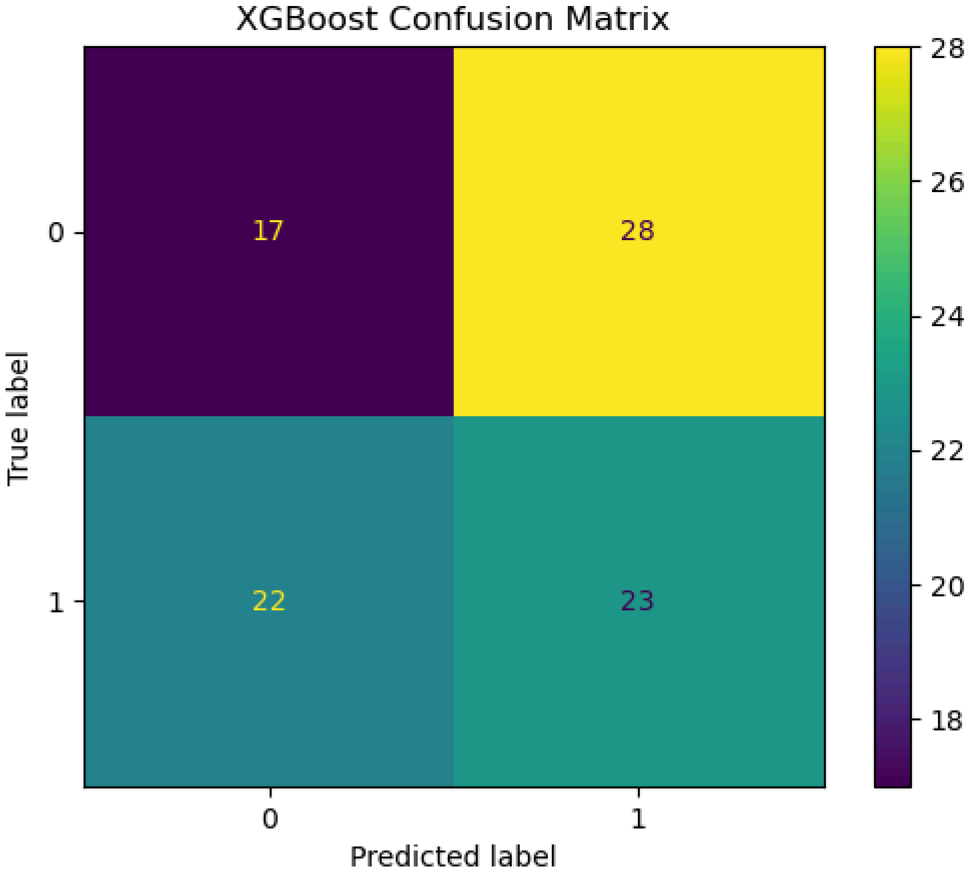
```
# Cell 10: Metrics and Plots for Experiment 2
# GRU Metrics and Plots
print_metrics_and_plots(y_test_class, gru_class, gru_class_prob, "GRU")

# XGBoost Metrics and Plots
print_metrics_and_plots(y_test_class, xgb_predictions, xgb_class_prob, "XGBoost")

# CatBoost Metrics and Plots
print_metrics_and_plots(y_test_class, catboost_predictions, catboost_class_prob, "C
```

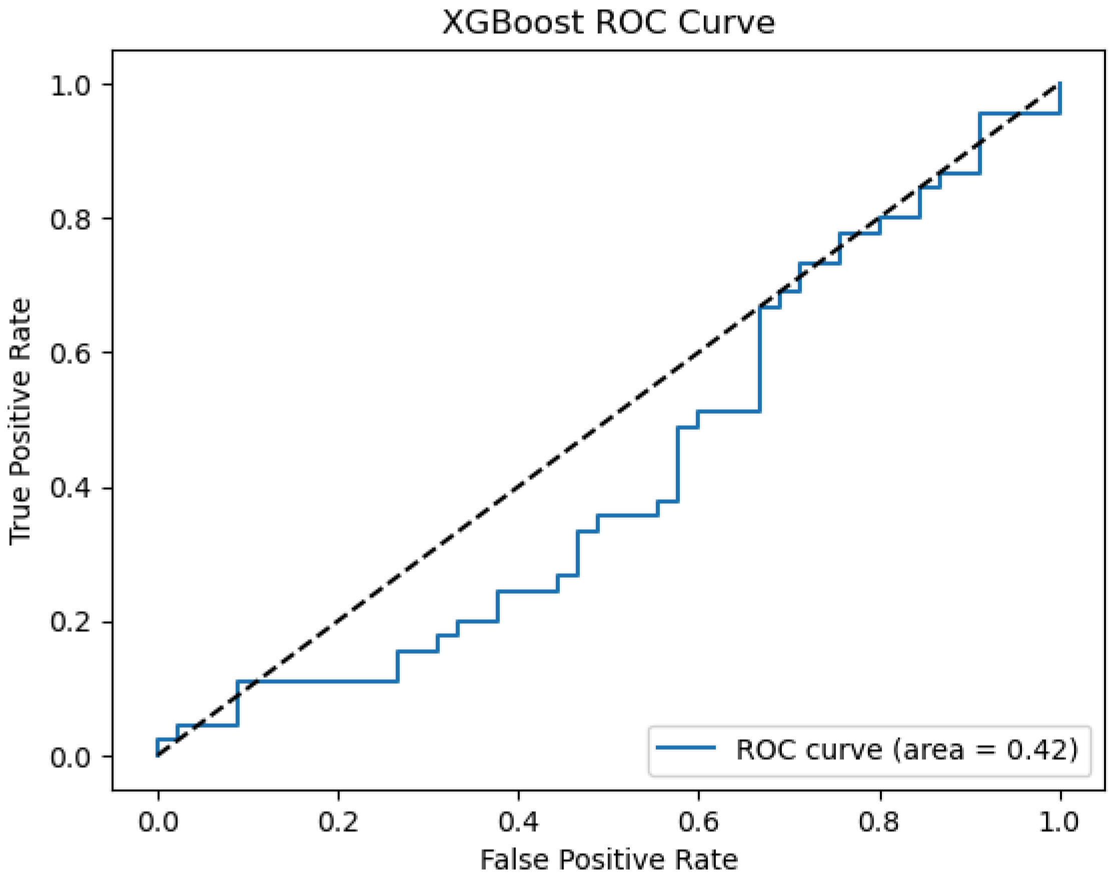
```
GRU Classification Metrics:
Accuracy: 0.5222222222222223
Precision: 0.5128205128205128
Recall: 0.8888888888888888
F1 Score: 0.6504065040650406
AUC-ROC: 0.5239506172839506
```


GRU Confusion Matrix

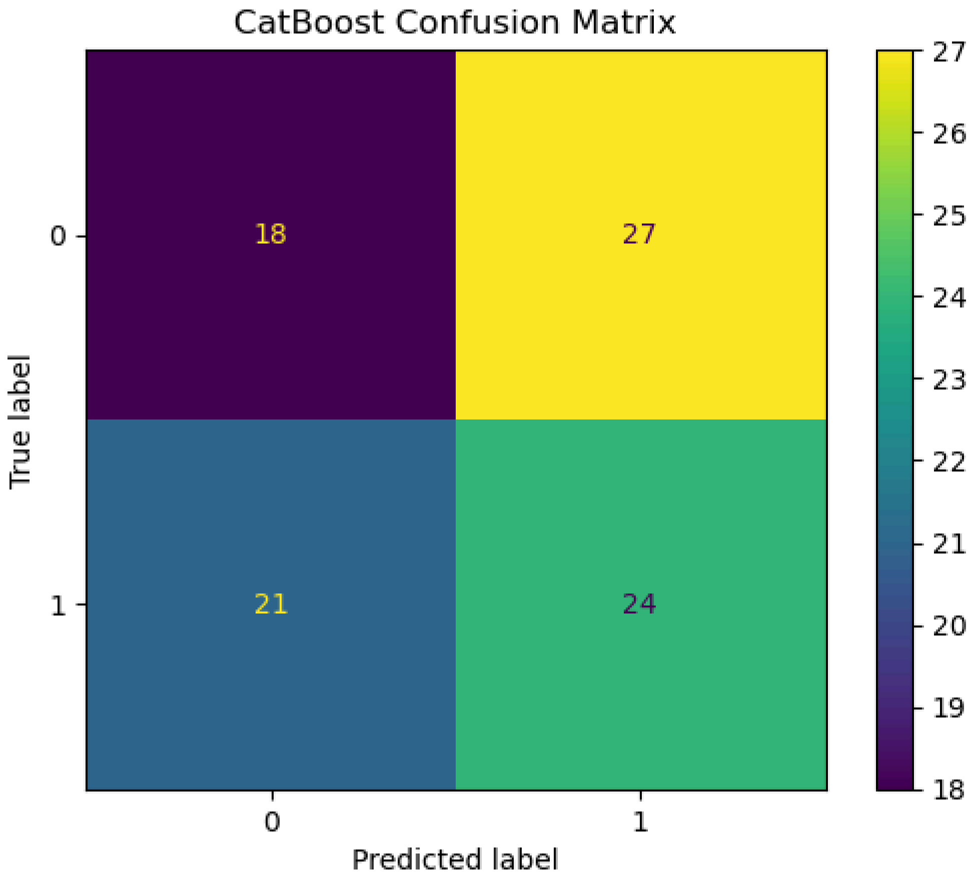

GRU ROC Curve

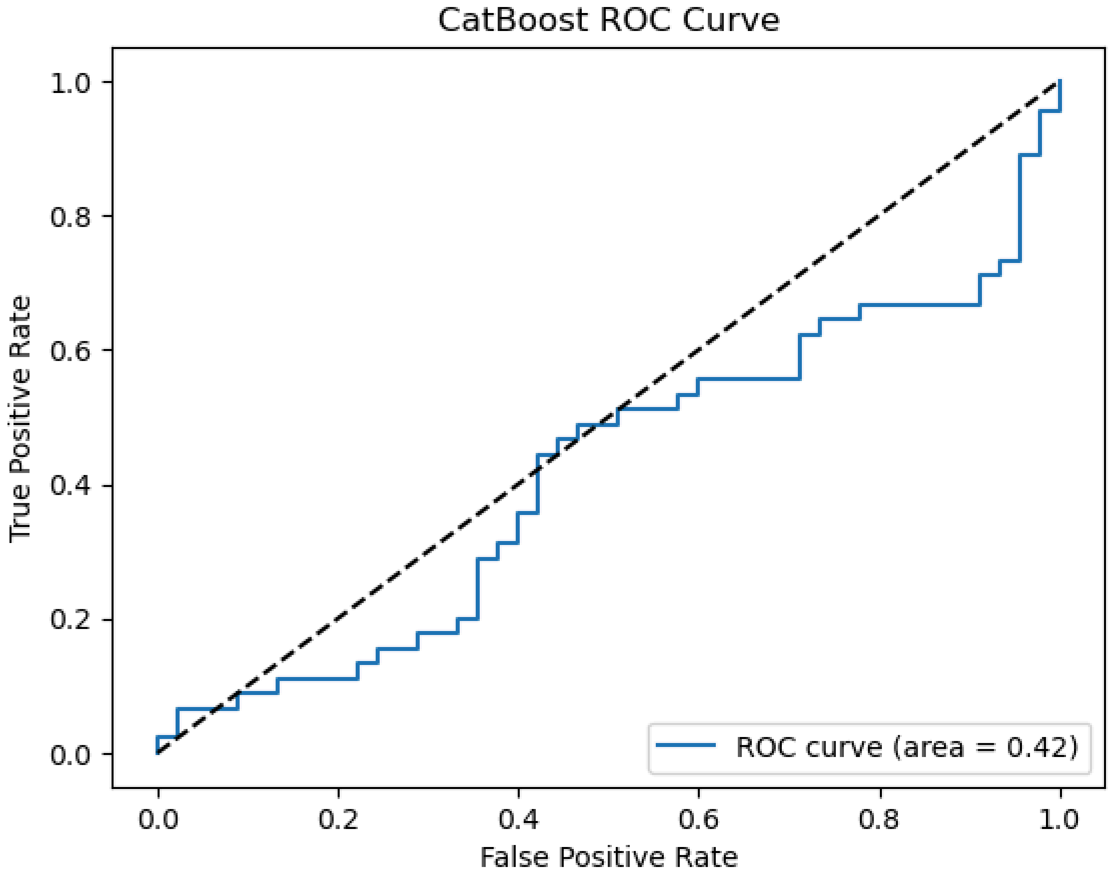
```
XGBoost Classification Metrics:
Accuracy: 0.4444444444444444
Precision: 0.45098039215686275
Recall: 0.5111111111111111
F1 Score: 0.4791666666666667
AUC-ROC: 0.4246913580246914
```





```
CatBoost Classification Metrics:
Accuracy: 0.4666666666666667
Precision: 0.47058823529411764
Recall: 0.5333333333333333
F1 Score: 0.5
AUC-ROC: 0.4162962962962963
```





In [74]:

```
#visualisation
```

In [88]:

```
# Add logic for Expense Tracking
spendings_df = pd.DataFrame(columns=['Date', 'Category', 'Credit Card Spendings', '

# Initialize variables
monthly_target = 0
spending_status = pn.pane.Markdown("Your spending status will be displayed here.")
predicted_spending = pn.pane.Markdown("Your spending prediction will be displayed h

# Widgets for the expense tracking tab
monthly_target_input = pn.widgets.TextInput(name="Monthly Target", placeholder="Ent
update_target_button = pn.widgets.Button(name="Update Target", button_type="primary

category_select = pn.widgets.Select(name="Category", options=['Restaurant', 'Grocer
credit_card_input = pn.widgets.TextInput(name="Credit Card Spendings", placeholder=
debit_card_input = pn.widgets.TextInput(name="Debit Card Spendings", placeholder="E
cash_input = pn.widgets.TextInput(name="Cash Spendings", placeholder="Enter spendin
update_spending_button = pn.widgets.Button(name="Update Spendings", button_type="pr

# Update target logic
def update_target(event):
    global monthly_target
    try:
        monthly_target = float(monthly_target_input.value)
        spending_status.object = f"Monthly target set to €{monthly_target}."
    except ValueError:
        spending_status.object = "Please enter a valid number for the monthly targe

update_target_button.on_click(update_target)

# Update spendings and predict logic
def update_spendings(event):
    global spendings_df, monthly_target
    if not monthly_target:
        spending_status.object = "Please set your monthly target first."
        return
    category = category_select.value
    credit = float(credit_card_input.value) if credit_card_input.value else 0.0
    debit = float(debit_card_input.value) if debit_card_input.value else 0.0
    cash = float(cash_input.value) if cash_input.value else 0.0
    date = pd.Timestamp.today().normalize()

    # Append new spending
    new_data = pd.DataFrame([{
        'Date': date,
        'Category': category,
        'Credit Card Spendings': credit,
        'Debit Card Spendings': debit,
        'Cash Spendings': cash
    }])
    spendings_df = pd.concat([spendings_df, new_data], ignore_index=True)

    # Update status
    monthly_data = spendings_df[spendings_df['Date'].dt.to_period('M') == date.to_p
    total_spent = monthly_data[['Credit Card Spendings', 'Debit Card Spendings', 'C
    remaining = monthly_target - total_spent
    spending_status.object = (
        f"Your monthly target is €{monthly_target}. "
        f"So far, you have spent a total of €{total_spent:.2f}. "
        f"You have €{remaining:.2f} remaining for the month."
    )

    # Predict the spending for the rest of the month
    days_in_month = (date + pd.offsets.MonthEnd(1)).day
    days_spent = date.day
    average_daily_spending = total_spent / days_spent
    predicted_total = average_daily_spending * days_in_month
    predicted_spending.object = (
        f"Based on your current spending, you are projected to spend a total of "
        f"€{predicted_total:.2f} this month. This means you will have spent "
        f"{'more than' if predicted_total > monthly_target else 'less than'} your t
        f"the month. "
    )

update_spending_button.on_click(update_spendings)

# Creation of the Expense Tracking tab
expense_tracking_tab = pn.Column(
    pn.pane.Markdown("## Expense Tracking"),
    monthly_target_input,
    update_target_button,
    category_select,
    credit_card_input,
    debit_card_input,
    cash_input,
    update_spending_button,
    spending_status,
    predicted_spending
)

# Visualization functions
def make_pie_chart(df, year, label):
    df_filtered = df[(df['Date'].dt.year == year) & (df['Expense/Income'] == label)
    return px.pie(df_filtered, names='Description', values='Amount', title=f"{label

def make_bar_chart(df, year, label):
    df_filtered = df[(df['Date'].dt.year == year) & (df['Expense/Income'] == label)
    return px.bar(df_filtered, x='Description', y='Amount', title=f"{label} Totals

def make_income_expense_line_chart(df):
    df_grouped = df.groupby(['Date', 'Expense/Income'])['Amount'].sum().reset_index
    return px.line(df_grouped, x='Date', y='Amount', color='Expense/Income', title=

def make_monthly_expense_income_chart(df, year):
    df_year = df[df['Date'].dt.year == year]
    df_year['Month'] = df_year['Date'].dt.to_period('M').astype(str)
    df_grouped = df_year.groupby(['Month', 'Expense/Income'])['Amount'].sum().reset
    return px.bar(df_grouped, x='Month', y='Amount', color='Expense/Income', barmod

# Dashboard Setup
tabs = pn.Tabs(
    ('2022', pn.Column(
        pn.Row(make_pie_chart(df, 2022, 'Expense'), make_bar_chart(df, 2022, 'Expen
        make_monthly_expense_income_chart(df, 2022)
    )),
    ('2023', pn.Column(
        pn.Row(make_pie_chart(df, 2023, 'Expense'), make_bar_chart(df, 2023, 'Expen
        make_monthly_expense_income_chart(df, 2023)
    )),
    ('Income vs. Expenses', make_income_expense_line_chart(df)),
    ('Expense Tracking', expense_tracking_tab),
    ('Stock Prediction', stock_fig)  # Ensure stock_fig is properly generated
)

dashboard = pn.template.FastListTemplate(
    title='Financial Dashboard',
    main=[tabs]
)

dashboard.servable()
pn.serve(dashboard, show=True, port=8089)
```

```
C:\Users\amuly\AppData\Local\Temp\ipykernel_8380\3124531871.py:105: SettingWithCopyW
arning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy

C:\Users\amuly\AppData\Local\Temp\ipykernel_8380\3124531871.py:105: SettingWithCopyW
arning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
```

```
Launching server at http://localhost:8089
```

```
Out[88]:  <panel.io.server.Server at 0x1d68232a850>
```

```
C:\Users\amuly\AppData\Local\Temp\ipykernel_8380\3124531871.py:50: FutureWarning:

The behavior of DataFrame concatenation with empty or all-NA entries is deprecated.
In a future version, this will no longer exclude empty or all-NA columns when determ
ining the result dtypes. To retain the old behavior, exclude the relevant entries be
fore the concat operation.
```

```
In [ ]:
```

```
In [ ]:
```