

```
import pandas as pd

df = pd.read_csv('creditcard.csv.zip')
df.head()

{"type": "dataframe", "variable_name": "df"}

print("\n Dataset Info:\n")
print(df.info())

print("\n Missing Values:\n")
print(df.isnull().sum())

print("\n Class Distribution:\n")
print(df['Class'].value_counts())
```

Dataset Info:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 284807 entries, 0 to 284806
Data columns (total 31 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Time        284807 non-null  float64
1   V1          284807 non-null  float64
2   V2          284807 non-null  float64
3   V3          284807 non-null  float64
4   V4          284807 non-null  float64
5   V5          284807 non-null  float64
6   V6          284807 non-null  float64
7   V7          284807 non-null  float64
8   V8          284807 non-null  float64
9   V9          284807 non-null  float64
10  V10         284807 non-null  float64
11  V11         284807 non-null  float64
12  V12         284807 non-null  float64
13  V13         284807 non-null  float64
14  V14         284807 non-null  float64
15  V15         284807 non-null  float64
16  V16         284807 non-null  float64
17  V17         284807 non-null  float64
18  V18         284807 non-null  float64
19  V19         284807 non-null  float64
20  V20         284807 non-null  float64
21  V21         284807 non-null  float64
22  V22         284807 non-null  float64
23  V23         284807 non-null  float64
24  V24         284807 non-null  float64
25  V25         284807 non-null  float64
```

```
26 V26      284807 non-null float64
27 V27      284807 non-null float64
28 V28      284807 non-null float64
29 Amount    284807 non-null float64
30 Class     284807 non-null int64
dtypes: float64(30), int64(1)
memory usage: 67.4 MB
None
```

□ Missing Values:

```
Time      0
V1        0
V2        0
V3        0
V4        0
V5        0
V6        0
V7        0
V8        0
V9        0
V10       0
V11       0
V12       0
V13       0
V14       0
V15       0
V16       0
V17       0
V18       0
V19       0
V20       0
V21       0
V22       0
V23       0
V24       0
V25       0
V26       0
V27       0
V28       0
Amount    0
Class     0
dtype: int64
```

□ Class Distribution:

```
Class
0      284315
1        492
Name: count, dtype: int64
```

```

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from imblearn.over_sampling import SMOTE

X = df.drop('Class', axis=1)
y = df['Class']

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

smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X_scaled, y)

print("\n Class distribution before SMOTE:")
print(y.value_counts())

print("\n Class distribution after SMOTE:")
print(pd.Series(y_resampled).value_counts())

Class distribution before SMOTE:
Class
0    284315
1      492
Name: count, dtype: int64

Class distribution after SMOTE:
Class
0    284315
1    284315
Name: count, dtype: int64

from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X_resampled,
y_resampled,
                                                    test_size=0.2,
                                                    random_state=42)

lr = LogisticRegression(max_iter=1000)
lr.fit(X_train, y_train)
lr_pred = lr.predict(X_test)

print("\n Logistic Regression Report:\n")

```

```
print(confusion_matrix(y_test, lr_pred))
print(classification_report(y_test, lr_pred))
```

□ Logistic Regression Report:

```
[[55361 1389]
 [ 4289 52687]]
```

	precision	recall	f1-score	support
0	0.93	0.98	0.95	56750
1	0.97	0.92	0.95	56976
accuracy			0.95	113726
macro avg	0.95	0.95	0.95	113726
weighted avg	0.95	0.95	0.95	113726

```
dt = DecisionTreeClassifier(random_state=42)
dt.fit(X_train, y_train)
dt_pred = dt.predict(X_test)
```

```
print("□ Decision Tree Report:\n")
print(confusion_matrix(y_test, dt_pred))
print(classification_report(y_test, dt_pred))
```

□ Decision Tree Report:

```
[[56605 145]
 [ 58 56918]]
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	56750
1	1.00	1.00	1.00	56976
accuracy			1.00	113726
macro avg	1.00	1.00	1.00	113726
weighted avg	1.00	1.00	1.00	113726

```
mlp = MLPClassifier(hidden_layer_sizes=(100,), max_iter=300,
random_state=42)
mlp.fit(X_train, y_train)
mlp_pred = mlp.predict(X_test)
```

```
print("□ Neural Network Report:\n")
print(confusion_matrix(y_test, mlp_pred))
print(classification_report(y_test, mlp_pred))
```

□ Neural Network Report:

```
[[56721 29]
```

```
[ 0 56976]]
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	56750
1	1.00	1.00	1.00	56976
accuracy			1.00	113726
macro avg	1.00	1.00	1.00	113726
weighted avg	1.00	1.00	1.00	113726

```
from sklearn.ensemble import IsolationForest
from sklearn.metrics import classification_report, confusion_matrix
```

```
iso_forest = IsolationForest(contamination=0.01, random_state=42)
y_pred_iforest = iso_forest.fit_predict(X_scaled)
```

```
y_pred_iforest = [1 if x == -1 else 0 for x in y_pred_iforest]
```

```
print("\n Isolation Forest Results:")
print(confusion_matrix(y, y_pred_iforest))
print(classification_report(y, y_pred_iforest))
```

```
\n Isolation Forest Results:
```

```
[[281755  2560]
 [   203    289]]
```

	precision	recall	f1-score	support
0	1.00	0.99	1.00	284315
1	0.10	0.59	0.17	492
accuracy			0.99	284807
macro avg	0.55	0.79	0.58	284807
weighted avg	1.00	0.99	0.99	284807

```
!pip install tensorflow --upgrade
```

```
Requirement already satisfied: tensorflow in
/usr/local/lib/python3.11/dist-packages (2.18.0)
Collecting tensorflow
  Downloading tensorflow-2.19.0-cp311-cp311-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (4.1 kB)
Requirement already satisfied: absl-py>=1.0.0 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=24.3.25 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (25.2.10)
```

```

Requirement already satisfied: gast!=0.5.0,!0.5.1,!0.5.2,>=0.2.1
in /usr/local/lib/python3.11/dist-packages (from tensorflow) (0.6.0)
Requirement already satisfied: google-pasta>=0.1.1 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: libclang>=13.0.0 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (18.1.1)
Requirement already satisfied: opt-einsum>=2.3.2 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (3.4.0)
Requirement already satisfied: packaging in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (24.2)
Requirement already satisfied: protobuf!=4.21.0,!4.21.1,!4.21.2,!
=4.21.3,!4.21.4,!4.21.5,<6.0.0dev,>=3.20.3 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (5.29.5)
Requirement already satisfied: requests<3,>=2.21.0 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (2.32.3)
Requirement already satisfied: setuptools in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (75.2.0)
Requirement already satisfied: six>=1.12.0 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (1.17.0)
Requirement already satisfied: termcolor>=1.1.0 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (3.1.0)
Requirement already satisfied: typing-extensions>=3.6.6 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (4.14.0)
Requirement already satisfied: wrapt>=1.11.0 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (1.17.2)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (1.72.1)
Collecting tensorboard~=2.19.0 (from tensorflow)
  Downloading tensorboard-2.19.0-py3-none-any.whl.metadata (1.8 kB)
Requirement already satisfied: keras>=3.5.0 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (3.8.0)
Requirement already satisfied: numpy<2.2.0,>=1.26.0 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (2.0.2)
Requirement already satisfied: h5py>=3.11.0 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (3.13.0)
Collecting ml-dtypes<1.0.0,>=0.5.1 (from tensorflow)
  Downloading ml_dtypes-0.5.1-cp311-cp311-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (21 kB)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (0.37.1)
Requirement already satisfied: wheel<1.0,>=0.23.0 in
/usr/local/lib/python3.11/dist-packages (from astunparse>=1.6.0-
>tensorflow) (0.45.1)
Requirement already satisfied: rich in /usr/local/lib/python3.11/dist-
packages (from keras>=3.5.0->tensorflow) (13.9.4)
Requirement already satisfied: namex in
/usr/local/lib/python3.11/dist-packages (from keras>=3.5.0-
>tensorflow) (0.1.0)
Requirement already satisfied: optree in

```

```

/usr/local/lib/python3.11/dist-packages (from keras>=3.5.0-
>tensorflow) (0.16.0)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0-
>tensorflow) (3.4.2)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0-
>tensorflow) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0-
>tensorflow) (2.4.0)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0-
>tensorflow) (2025.4.26)
Requirement already satisfied: markdown>=2.6.8 in
/usr/local/lib/python3.11/dist-packages (from tensorboard~=2.19.0-
>tensorflow) (3.8)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0
in /usr/local/lib/python3.11/dist-packages (from tensorboard~=2.19.0-
>tensorflow) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in
/usr/local/lib/python3.11/dist-packages (from tensorboard~=2.19.0-
>tensorflow) (3.1.3)
Requirement already satisfied: MarkupSafe>=2.1.1 in
/usr/local/lib/python3.11/dist-packages (from werkzeug>=1.0.1-
>tensorboard~=2.19.0->tensorflow) (3.0.2)
Requirement already satisfied: markdown-it-py>=2.2.0 in
/usr/local/lib/python3.11/dist-packages (from rich->keras>=3.5.0-
>tensorflow) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in
/usr/local/lib/python3.11/dist-packages (from rich->keras>=3.5.0-
>tensorflow) (2.19.1)
Requirement already satisfied: mdurl~=0.1 in
/usr/local/lib/python3.11/dist-packages (from markdown-it-py>=2.2.0-
>rich->keras>=3.5.0->tensorflow) (0.1.2)
Downloading tensorflow-2.19.0-cp311-cp311-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl (644.9 MB)
_____ 644.9/644.9 MB 858.0 kB/s eta
0:00:00
l_dtypes-0.5.1-cp311-cp311-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl (4.7 MB)
_____ 4.7/4.7 MB 81.7 MB/s eta
0:00:00
_____ 5.5/5.5 MB 76.7 MB/s eta
0:00:00
l-dtypes, tensorboard, tensorflow
  Attempting uninstall: ml-dtypes
    Found existing installation: ml-dtypes 0.4.1
    Uninstalling ml-dtypes-0.4.1:

```

```
Successfully uninstalled ml-dtypes-0.4.1
Attempting uninstall: tensorboard
Found existing installation: tensorboard 2.18.0
Uninstalling tensorboard-2.18.0:
Successfully uninstalled tensorboard-2.18.0
Attempting uninstall: tensorflow
Found existing installation: tensorflow 2.18.0
Uninstalling tensorflow-2.18.0:
Successfully uninstalled tensorflow-2.18.0
ERROR: pip's dependency resolver does not currently take into account
all the packages that are installed. This behaviour is the source of
the following dependency conflicts.
tensorflow-decision-forests 1.11.0 requires tensorflow==2.18.0, but
you have tensorflow 2.19.0 which is incompatible.
tensorflow-text 2.18.1 requires tensorflow<2.19,>=2.18.0, but you have
tensorflow 2.19.0 which is incompatible.
tf-keras 2.18.0 requires tensorflow<2.19,>=2.18, but you have
tensorflow 2.19.0 which is incompatible.
Successfully installed ml-dtypes-0.5.1 tensorboard-2.19.0 tensorflow-
2.19.0
```

```
import tensorflow as tf
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input, Dense

input_dim = X_scaled.shape[1]
input_layer = Input(shape=(input_dim,))
encoded = Dense(16, activation='relu')(input_layer)
encoded = Dense(8, activation='relu')(encoded)
decoded = Dense(16, activation='relu')(encoded)
output_layer = Dense(input_dim, activation='linear')(decoded)

autoencoder = Model(inputs=input_layer, outputs=output_layer)
autoencoder.compile(optimizer='adam', loss='mse')

autoencoder.fit(X_scaled, X_scaled, epochs=10, batch_size=256,
shuffle=True, validation_split=0.2)

reconstructions = autoencoder.predict(X_scaled)
mse = tf.keras.losses.mse(X_scaled, reconstructions).numpy()
threshold = mse.mean() + 3 * mse.std()

y_pred_auto = [1 if e > threshold else 0 for e in mse]

print("\n Autoencoder Results:")
print(confusion_matrix(y, y_pred_auto))
print(classification_report(y, y_pred_auto))
```



```

Epoch 1/10
891/891 _____ 4s 3ms/step - loss: 0.9119 - val_loss: 0.7111
Epoch 2/10
891/891 _____ 2s 3ms/step - loss: 0.6220 - val_loss: 0.6301
Epoch 3/10
891/891 _____ 3s 3ms/step - loss: 0.5770 - val_loss: 0.5951
Epoch 4/10
891/891 _____ 4s 2ms/step - loss: 0.5423 - val_loss: 0.5685
Epoch 5/10
891/891 _____ 3s 2ms/step - loss: 0.5248 - val_loss: 0.5583
Epoch 6/10
891/891 _____ 3s 2ms/step - loss: 0.5129 - val_loss: 0.5446
Epoch 7/10
891/891 _____ 3s 3ms/step - loss: 0.4974 - val_loss: 0.5390
Epoch 8/10
891/891 _____ 5s 3ms/step - loss: 0.4919 - val_loss: 0.5359
Epoch 9/10
891/891 _____ 2s 2ms/step - loss: 0.4845 - val_loss: 0.5332
Epoch 10/10
891/891 _____ 2s 3ms/step - loss: 0.4830 - val_loss: 0.5312
8901/8901 _____ 10s 1ms/step

```

□ Autoencoder Results:

```

[[283675    640]
 [   469     23]]

```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	284315
1	0.03	0.05	0.04	492
accuracy			1.00	284807
macro avg	0.52	0.52	0.52	284807
weighted avg	1.00	1.00	1.00	284807

```

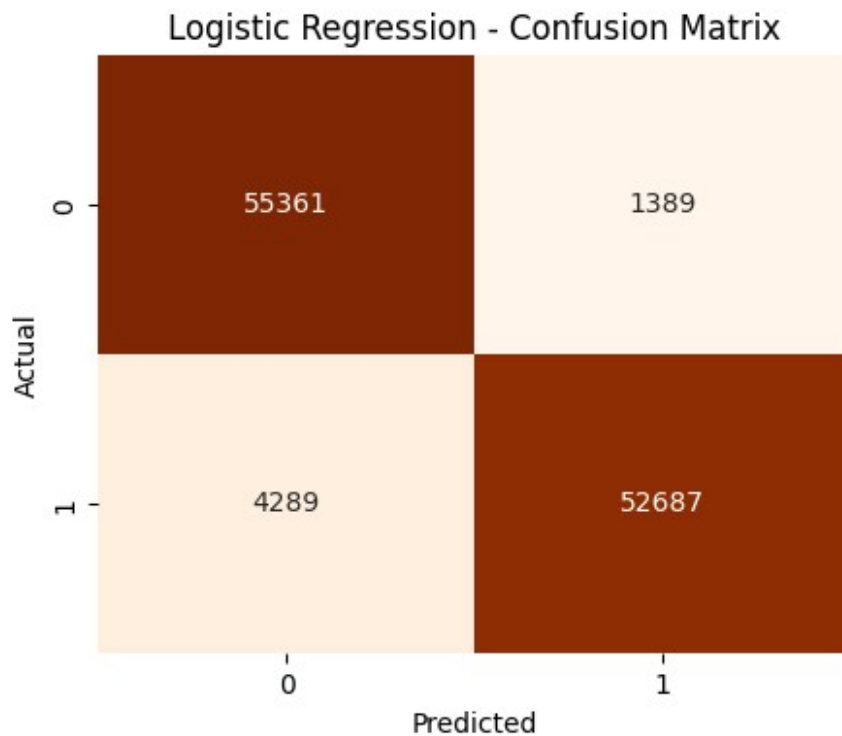
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix

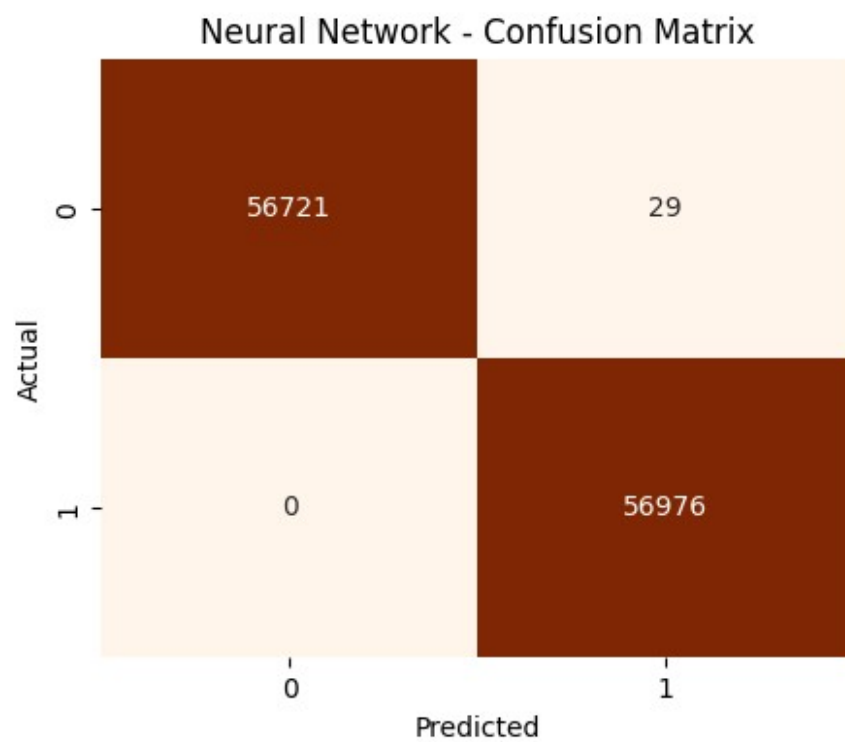
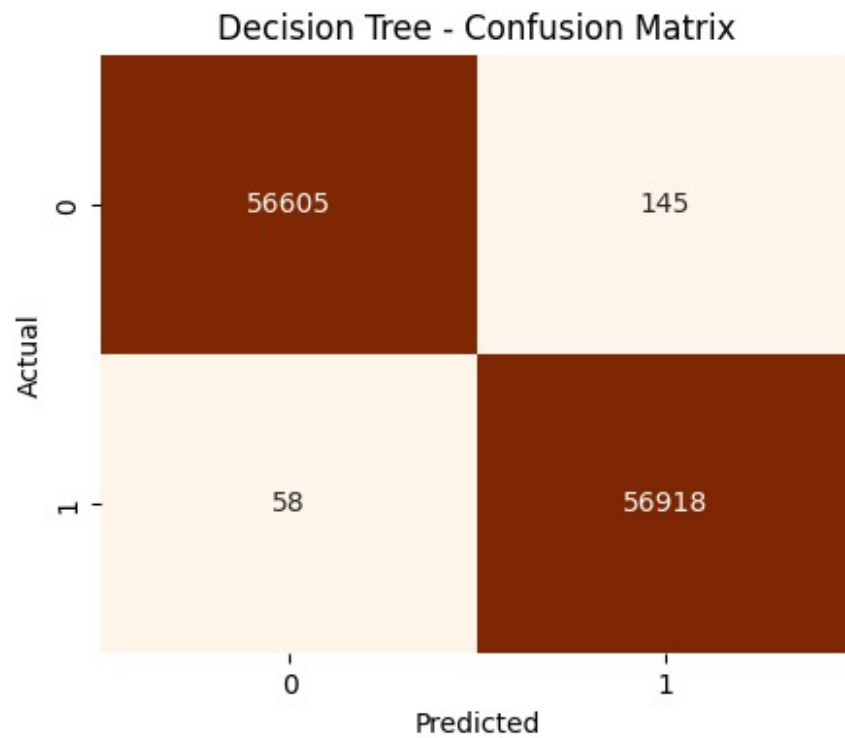
def plot_confusion(model_name, y_true, y_pred):
    cm = confusion_matrix(y_true, y_pred)

```

```
plt.figure(figsize=(5, 4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Oranges', cbar=False)
plt.title(f"{model_name} - Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

```
plot_confusion("Logistic Regression", y_test, lr_pred)
plot_confusion("Decision Tree", y_test, dt_pred)
plot_confusion("Neural Network", y_test, mlp_pred)
```





```
from sklearn.metrics import roc_curve, roc_auc_score  
import matplotlib.pyplot as plt
```

```

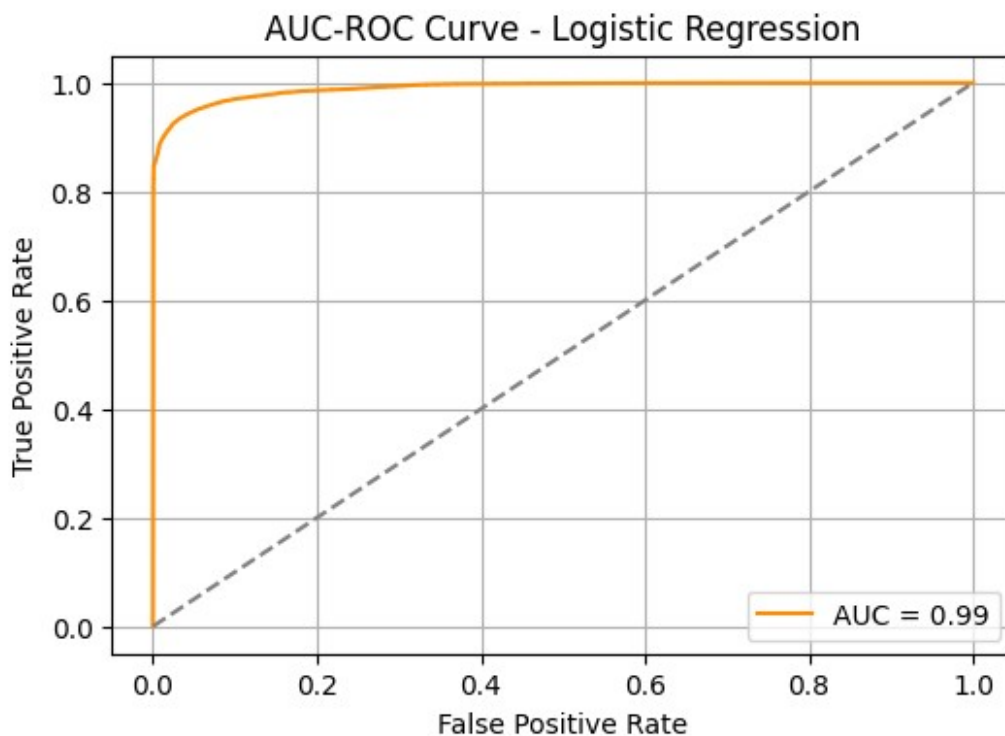
y_prob = lr.predict_proba(X_test)[: , 1]

auc_score = roc_auc_score(y_test, y_prob)

fpr, tpr, thresholds = roc_curve(y_test, y_prob)

plt.figure(figsize=(6, 4))
plt.plot(fpr, tpr, label=f'AUC = {auc_score:.2f}', color='darkorange')
plt.plot([0, 1], [0, 1], linestyle='--', color='gray')
plt.title('AUC-ROC Curve - Logistic Regression')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.legend()
plt.grid(True)
plt.show()

```



```

from sklearn.metrics import accuracy_score
import seaborn as sns

models = ['Logistic Regression', 'Decision Tree', 'Neural Network']
accuracies = [
    accuracy_score(y_test, lr_pred),
    accuracy_score(y_test, dt_pred),

```

```

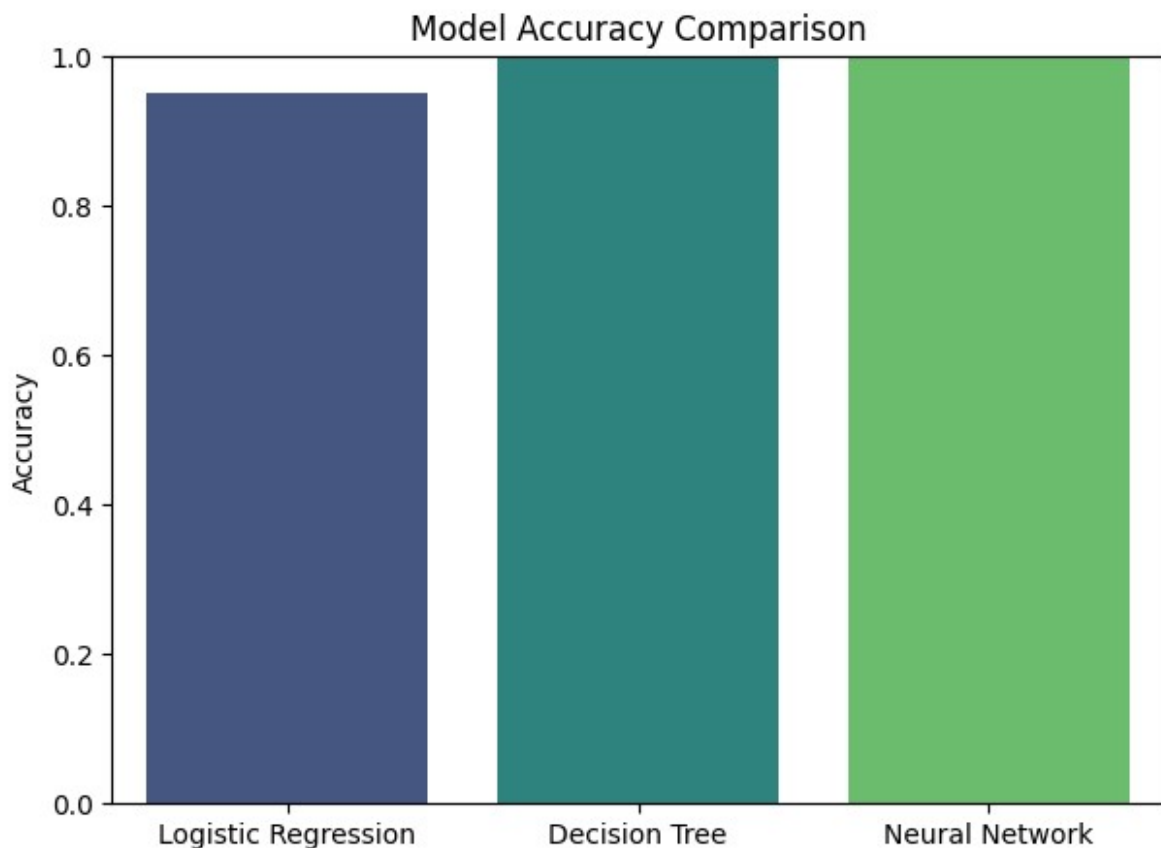
    accuracy_score(y_test, mlp_pred)
]

plt.figure(figsize=(7, 5))
sns.barplot(x=models, y=accuracies, palette="viridis")
plt.title("Model Accuracy Comparison")
plt.ylabel("Accuracy")
plt.ylim(0, 1)
plt.show()

<ipython-input-24-2395171501>:12: FutureWarning:
Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

sns.barplot(x=models, y=accuracies, palette="viridis")

```



```

import matplotlib.pyplot as plt

labels = ['Normal', 'Fraud']
sizes = y.value_counts()

```

```
plt.figure(figsize=(5, 5))
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=90,
        colors=['#ff7043', '#fff3e0'])
plt.title("Original Class Distribution")
plt.axis('equal')
plt.show()
```

