ml3-1

July 4, 2024

[1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
data = pd.read_csv("/content/Health1.csv")
data.head()
```

[1]:
```
      index  National Provider Identifier  \
0  8774979                     1891106191
1  3354385                     1346202256
2  3001884                     1306820956
3  7594822                     1770523540
4   746159                     1073627758

  Last Name/Organization Name of the Provider First Name of the Provider  \
0                                UPADHYAYULA                   SATYASREE
1                                      JONES                       WENDY
2                                   DUROCHER                     RICHARD
3                                    FULLARD                      JASPER
4                                   PERROTTI                     ANTHONY

  Middle Initial of the Provider Credentials of the Provider  \
0                            NaN                       M.D.
1                              P                       M.D.
2                              W                        DPM
3                            NaN                         MD
4                              E                         DO

  Gender of the Provider Entity Type of the Provider  \
0                      F                           I
1                      F                           I
2                      M                           I
3                      M                           I
4                      M                           I

  Street Address 1 of the Provider Street Address 2 of the Provider  …  \
```

```
0                1402 S GRAND BLVD                    FDT 14TH FLOOR  …
1                  2950 VILLAGE DR                               NaN  …
2                20 WASHINGTON AVE                         STE 212  …
3              5746 N BROADWAY ST                               NaN  …
4                875 MILITARY TRL                        SUITE 200  …

  HCPCS Code                                  HCPCS Description  \
0      99223  Initial hospital inpatient care, typically 70 …
1      G0202  Screening mammography, bilateral (2-view study…
2      99348  Established patient home visit, typically 25 m…
3      81002                            Urinalysis, manual test
4      96372  Injection beneath the skin or into muscle for …

  HCPCS Drug Indicator Number of Services Number of Medicare Beneficiaries  \
0                    N                27                                24
1                    N               175                               175
2                    N                32                                13
3                    N                20                                18
4                    N                33                                24

  Number of Distinct Medicare Beneficiary/Per Day Services  \
0                                                 27
1                                                175
2                                                 32
3                                                 20
4                                                 31

  Average Medicare Allowed Amount Average Submitted Charge Amount  \
0                    200.58777778                     305.21111111
1                          123.73                            548.8
2                           90.65                              155
3                             3.5                                5
4                           26.52                               40

  Average Medicare Payment Amount Average Medicare Standardized Amount
0                    157.26222222                          160.90888889
1                          118.83                          135.31525714
2                       64.4396875                           60.5959375
3                            3.43                                3.43
4                    19.539393939                          19.057575758

[5 rows x 27 columns]
```

[ ]:
```python
from sklearn.preprocessing import StandardScaler, LabelEncoder

# Identify numerical and categorical columns
numerical_cols = data.select_dtypes(include=['int64', 'float64']).columns
```

```
categorical_cols = data.select_dtypes(include=['object', 'category']).columns

# Encode categorical variables
label_encoders = {}
for col in categorical_cols:
    le = LabelEncoder()
    data[col] = le.fit_transform(data[col])
    label_encoders[col] = le

# Scale numerical features
scaler = StandardScaler()
data[numerical_cols] = scaler.fit_transform(data[numerical_cols])

# Display preprocessed data
print(data.head())
```

```
      index  National Provider Identifier  \
0  1.361920                      1.366960
1 -0.546996                     -0.528945
2 -0.671133                     -0.665966
3  0.946316                      0.947412
4 -1.465509                     -1.477323

   Last Name/Organization Name of the Provider  First Name of the Provider  \
0                                        39468                      10215
1                                        18353                      12477
2                                        10622                       9592
3                                        12995                       5036
4                                        29475                        773

   Middle Initial of the Provider  Credentials of the Provider  \
0                              29                          667
1                              18                          667
2                              25                          438
3                              29                          956
4                               7                          410

   Gender of the Provider  Entity Type of the Provider  \
0                       0                            0
1                       0                            0
2                       1                            0
3                       1                            0
4                       1                            0

   Street Address 1 of the Provider  Street Address 2 of the Provider  …  \
0                              9763                              3452  …
1                             25436                             10024  …
```

```
2                              16839                              6156  …
3                              39461                             10024  …
4                              48417                              7639  …

   HCPCS Code  HCPCS Description  HCPCS Drug Indicator  Number of Services  \
0        2251                967                     0                1414
1        2374               2054                     0                 943
2        2295                665                     0                1656
3        1329               2330                     0                1288
4        2163                973                     0                1669

   Number of Medicare Beneficiaries  \
0                               427
1                               304
2                               245
3                               309
4                               427

   Number of Distinct Medicare Beneficiary/Per Day Services  \
0                                               1015
1                                                729
2                                               1140
3                                                930
4                                               1129

   Average Medicare Allowed Amount  Average Submitted Charge Amount  \
0                            18375                             20594
1                             8016                             29310
2                            47539                              8716
3                            25406                             27866
4                            23077                             24583

   Average Medicare Payment Amount  Average Medicare Standardized Amount
0                            16676                                 15861
1                             8422                                 11560
2                            64550                                 56850
3                            35619                                 31586
4                            22947                                 20127

[5 rows x 27 columns]
```

[ ]:
```python
import pandas as pd

# Load the dataset
file_path = '/content/Health1.csv'
data = pd.read_csv(file_path)
```

```python
# Display the first few rows of the dataset
print(data.head())

# Display summary statistics
print(data.describe())

# Display column information
print(data.info())
```

```
     index  National Provider Identifier  \
0  8774979                     1891106191
1  3354385                     1346202256
2  3001884                     1306820956
3  7594822                     1770523540
4   746159                     1073627758

  Last Name/Organization Name of the Provider First Name of the Provider  \
0                                UPADHYAYULA                   SATYASREE
1                                      JONES                       WENDY
2                                   DUROCHER                     RICHARD
3                                    FULLARD                      JASPER
4                                   PERROTTI                     ANTHONY

  Middle Initial of the Provider Credentials of the Provider  \
0                            NaN                        M.D.
1                              P                        M.D.
2                              W                         DPM
3                            NaN                          MD
4                              E                          DO

  Gender of the Provider Entity Type of the Provider  \
0                      F                           I
1                      F                           I
2                      M                           I
3                      M                           I
4                      M                           I

  Street Address 1 of the Provider Street Address 2 of the Provider  …  \
0                1402 S GRAND BLVD                   FDT 14TH FLOOR  …
1                  2950 VILLAGE DR                              NaN  …
2                20 WASHINGTON AVE                          STE 212  …
3               5746 N BROADWAY ST                              NaN  …
4                 875 MILITARY TRL                       SUITE 200  …

  HCPCS Code                                HCPCS Description  \
0      99223  Initial hospital inpatient care, typically 70 …
1      G0202  Screening mammography, bilateral (2-view study…
```

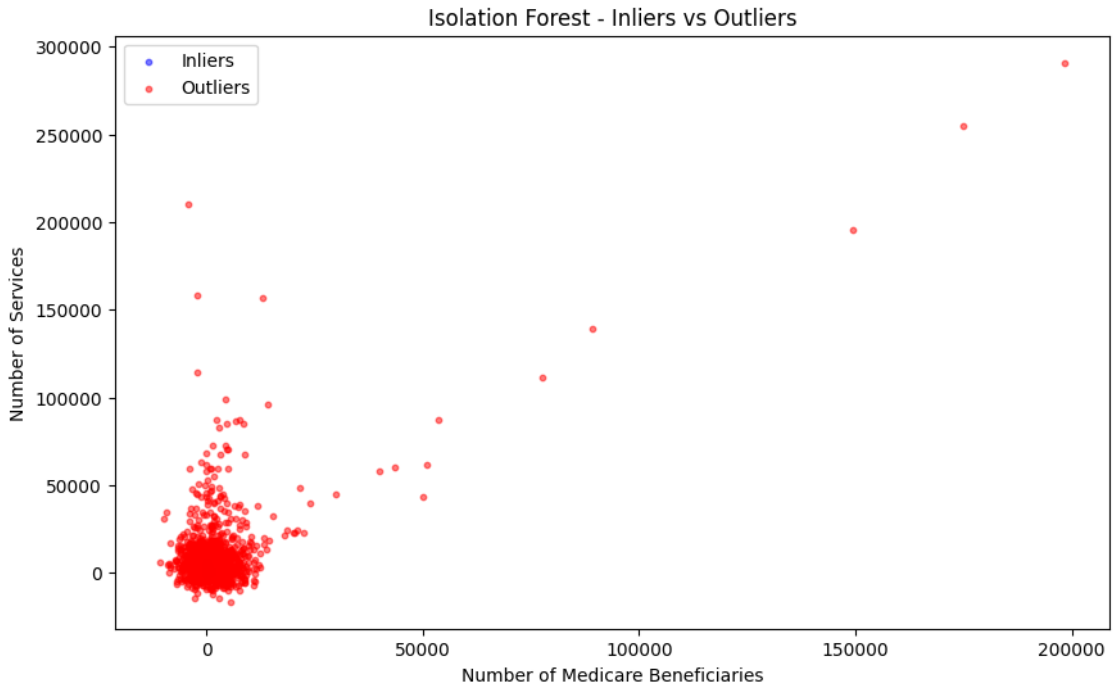
```
2      99348  Established patient home visit, typically 25 m…
3      81002                             Urinalysis, manual test
4      96372  Injection beneath the skin or into muscle for …

  HCPCS Drug Indicator Number of Services Number of Medicare Beneficiaries  \
0                    N                 27                                24
1                    N                175                               175
2                    N                 32                                13
3                    N                 20                                18
4                    N                 33                                24

  Number of Distinct Medicare Beneficiary/Per Day Services  \
0                                                 27
1                                                175
2                                                 32
3                                                 20
4                                                 31

  Average Medicare Allowed Amount Average Submitted Charge Amount  \
0                    200.58777778                     305.21111111
1                          123.73                            548.8
2                           90.65                              155
3                             3.5                                5
4                           26.52                               40

  Average Medicare Payment Amount Average Medicare Standardized Amount
0                    157.26222222                          160.90888889
1                          118.83                          135.31525714
2                       64.4396875                            60.5959375
3                            3.43                                  3.43
4                    19.539393939                          19.057575758

[5 rows x 27 columns]
              index  National Provider Identifier  Zip Code of the Provider
count  8.530000e+04                  8.530000e+04              8.529900e+04
mean   4.910808e+06                  1.498548e+09              4.164880e+08
std    2.839993e+06                  2.874512e+08              3.082592e+08
min    2.090000e+02                  1.003001e+09              6.010000e+02
25%    2.460293e+06                  1.245738e+09              1.426300e+08
50%    4.908030e+06                  1.497875e+09              3.653220e+08
75%    7.351746e+06                  1.740383e+09              6.834213e+08
max    9.847440e+06                  1.993000e+09              9.990166e+08
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 85300 entries, 0 to 85299
Data columns (total 27 columns):
 #   Column                                                Non-Null Count
Dtype
---  ------                                                --------------
```

```
-----
 0   index                                                 85300 non-null
int64
 1   National Provider Identifier                          85300 non-null
int64
 2   Last Name/Organization Name of the Provider           85300 non-null
object
 3   First Name of the Provider                            81701 non-null
object
 4   Middle Initial of the Provider                        60332 non-null
object
 5   Credentials of the Provider                           79155 non-null
object
 6   Gender of the Provider                                81702 non-null
object
 7   Entity Type of the Provider                           85299 non-null
object
 8   Street Address 1 of the Provider                      85299 non-null
object
 9   Street Address 2 of the Provider                      34680 non-null
object
 10  City of the Provider                                  85299 non-null
object
 11  Zip Code of the Provider                              85299 non-null
float64
 12  State Code of the Provider                            85299 non-null
object
 13  Country Code of the Provider                          85299 non-null
object
 14  Provider Type                                         85299 non-null
object
 15  Medicare Participation Indicator                      85299 non-null
object
 16  Place of Service                                      85299 non-null
object
 17  HCPCS Code                                            85299 non-null
object
 18  HCPCS Description                                     85299 non-null
object
 19  HCPCS Drug Indicator                                  85299 non-null
object
 20  Number of Services                                    85299 non-null
object
 21  Number of Medicare Beneficiaries                      85299 non-null
object
 22  Number of Distinct Medicare Beneficiary/Per Day Services  85299 non-null
object
 23  Average Medicare Allowed Amount                       85299 non-null
```

```
object
 24  Average Submitted Charge Amount                    85299 non-null
object
 25  Average Medicare Payment Amount                    85299 non-null
object
 26  Average Medicare Standardized Amount               85299 non-null
object
dtypes: float64(1), int64(2), object(24)
memory usage: 17.6+ MB
None
```

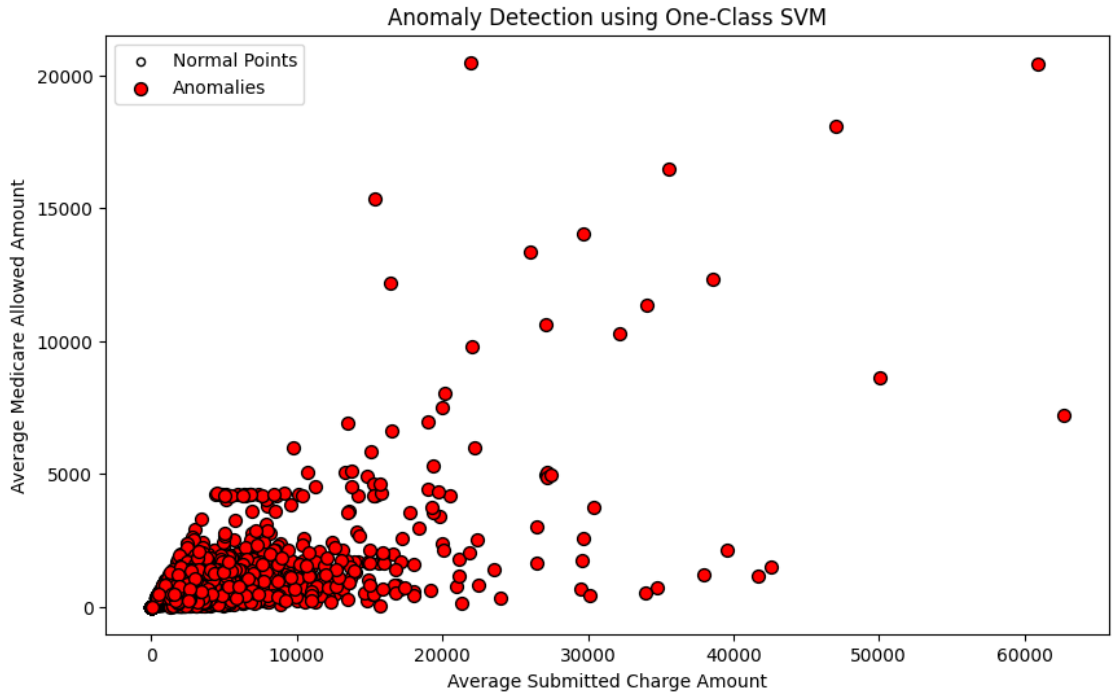
[3]:
```python
#Isolation Forest
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import IsolationForest

# Load the dataset
file_path = '/content/Health1.csv'
data = pd.read_csv(file_path)

# Select the first 10000 rows
data = data.head(100000)

# Select the two numerical features for visualization
selected_features = ['Number of Medicare Beneficiaries', 'Number of Services']

# Remove commas and convert columns to numeric
for col in selected_features:
    data[col] = data[col].astype(str).str.replace(',', '')
    data[col] = pd.to_numeric(data[col], errors='coerce')

# Handle missing values
X = data[selected_features]
X = X.dropna()

# Standardize the data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Train the Isolation Forest on the dataset
clf = IsolationForest(contamination=0.01, random_state=42)
clf.fit(X_scaled)

# Predict anomalies in the dataset
y_pred_full = clf.predict(X_scaled)
```

```
# Add predictions to the original data
data['Prediction'] = y_pred_full

# Separate inliers and outliers
inliers = data[data['Prediction'] == 1]
outliers = data[data['Prediction'] == -1]

# Function to add jitter
def add_jitter(arr, jitter_strength=0.02):
    stdev = jitter_strength * (max(arr) - min(arr))
    return arr + np.random.randn(len(arr)) * stdev

# Visualize the data using the two selected features
plt.figure(figsize=(10, 6))
plt.scatter(add_jitter(inliers[selected_features[0]]),␣
 ↪add_jitter(inliers[selected_features[1]]),
            c='blue', label='Inliers', s=10, alpha=0.5)
plt.scatter(add_jitter(outliers[selected_features[0]]),␣
 ↪add_jitter(outliers[selected_features[1]]),
            c='red', label='Outliers', s=10, alpha=0.5)
plt.xlabel(selected_features[0])
plt.ylabel(selected_features[1])
plt.legend()
plt.title('Isolation Forest - Inliers vs Outliers')
plt.show()
```



[4]:
```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn import svm

# Load the dataset
file_path = '/content/Health1.csv'
data = pd.read_csv(file_path)

# Select the two numerical features for visualization
selected_features = [
    'Average Submitted Charge Amount', 'Average Medicare Allowed Amount'
]

# Remove commas and convert columns to numeric
for col in selected_features:
    data[col] = data[col].astype(str).str.replace(',', '')
    data[col] = pd.to_numeric(data[col], errors='coerce')

# Handle missing values
X = data[selected_features]
X = X.dropna()

# Standardize the data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Set contamination level
contamination = 0.1  # Adjust contamination level as needed

# Train the One-Class SVM on the full dataset
clf = svm.OneClassSVM(nu=contamination, kernel="rbf", gamma=0.1)
clf.fit(X_scaled)

# Predict anomalies in the full dataset
y_pred_full = clf.predict(X_scaled)

# Add the predicted labels to the original dataset
data['Anomaly'] = y_pred_full

# Visualize the results
plt.figure(figsize=(10, 6))

# Plot the normal points
```

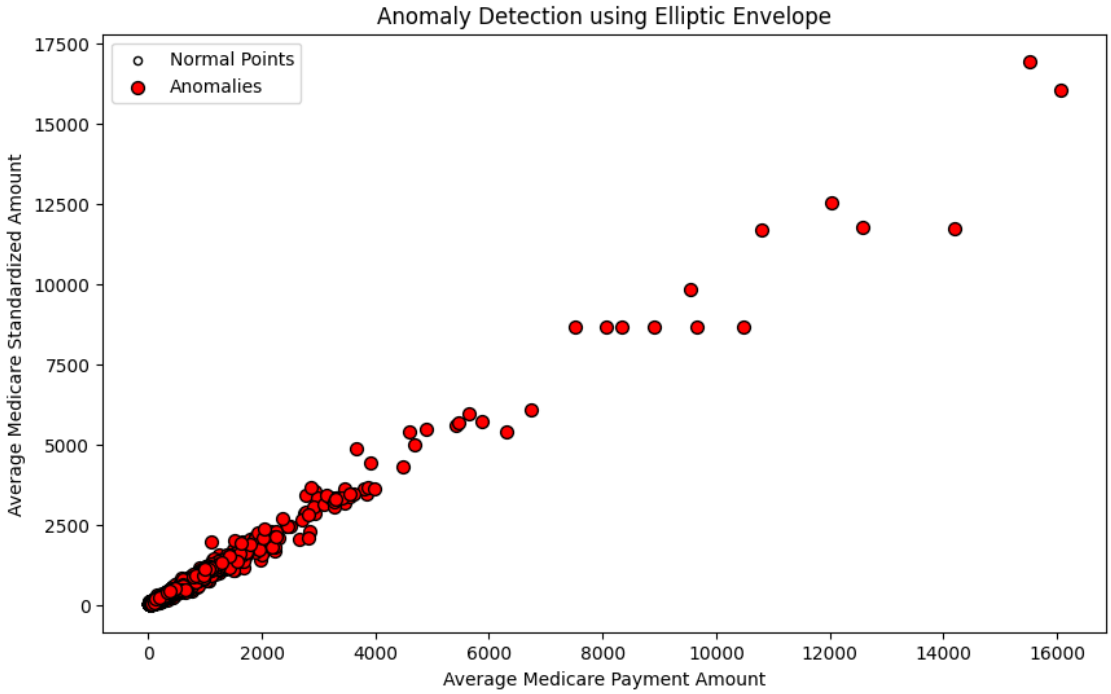
```
plt.scatter(X['Average Submitted Charge Amount'], X['Average Medicare Allowed␣
 ↪Amount'], c='white', edgecolors='k', s=20, label='Normal Points')

# Plot the anomalies
plt.scatter(X[data['Anomaly'] == -1]['Average Submitted Charge Amount'],␣
 ↪X[data['Anomaly'] == -1]['Average Medicare Allowed Amount'], c='red',␣
 ↪edgecolors='k', s=50, label='Anomalies')

plt.title('Anomaly Detection using One-Class SVM')
plt.xlabel('Average Submitted Charge Amount')
plt.ylabel('Average Medicare Allowed Amount')
plt.legend()
plt.show()
```



[49]:
```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.covariance import EllipticEnvelope

# Load the dataset
file_path = '/content/Health1.csv'
data = pd.read_csv(file_path)

# Select the two numerical features for visualization
```

```
selected_features = [
    'Average Medicare Payment Amount', 'Average Medicare Standardized Amount'
]

# Remove commas and convert columns to numeric
for col in selected_features:
    data[col] = data[col].astype(str).str.replace(',', '')
    data[col] = pd.to_numeric(data[col], errors='coerce')

# Handle missing values
X = data[selected_features]
X = X.dropna()

# Standardize the data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Set contamination level
contamination = 0.1  # Adjust contamination level as needed

# Train the Elliptic Envelope on the full dataset
elliptic_env = EllipticEnvelope(contamination=contamination)
elliptic_env.fit(X_scaled)

# Predict anomalies in the full dataset
y_pred_full = elliptic_env.predict(X_scaled)

# Add the predicted labels to the original dataset
data['Anomaly'] = y_pred_full

# Visualize the results
plt.figure(figsize=(10, 6))

# Plot the normal points
plt.scatter(X['Average Medicare Payment Amount'], X['Average Medicare␣
 ↪Standardized Amount'], c='white', edgecolors='k', s=20, label='Normal␣
 ↪Points')

# Plot the anomalies
plt.scatter(X[data['Anomaly'] == -1]['Average Medicare Payment Amount'],␣
 ↪X[data['Anomaly'] == -1]['Average Medicare Standardized Amount'], c='red',␣
 ↪edgecolors='k', s=50, label='Anomalies')

plt.title('Anomaly Detection using Elliptic Envelope')
plt.xlabel('Average Medicare Payment Amount')
plt.ylabel('Average Medicare Standardized Amount')
plt.legend()
```

```
plt.show()
```

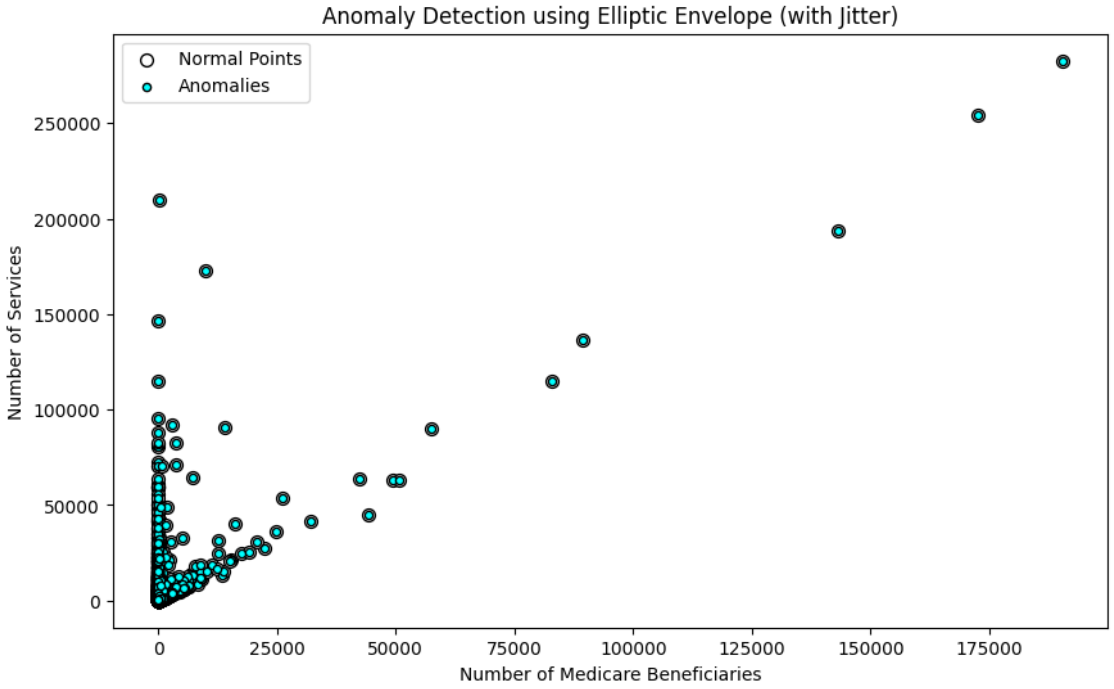


[6]:
```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.covariance import EllipticEnvelope
import numpy as np

# Load the dataset
file_path = '/content/Health1.csv'
data = pd.read_csv(file_path)

# Select the two numerical features for visualization
selected_features = [
    'Number of Medicare Beneficiaries', 'Number of Services'
]

# Remove commas and convert columns to numeric
for col in selected_features:
    data[col] = data[col].astype(str).str.replace(',', '')
    data[col] = pd.to_numeric(data[col], errors='coerce')

# Handle missing values
X = data[selected_features]
```

```
X = X.dropna()

# Standardize the data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Set contamination level
contamination = 0.1  # Adjust contamination level as needed

# Train the Elliptic Envelope on the full dataset
elliptic_env = EllipticEnvelope(contamination=contamination)
elliptic_env.fit(X_scaled)

# Predict anomalies in the full dataset
y_pred_full = elliptic_env.predict(X_scaled)

# Add the predicted labels to the original dataset
data['Anomaly'] = y_pred_full

# Count the number of anomalies and normal points
num_anomalies = (data['Anomaly'] == -1).sum()
num_normals = (data['Anomaly'] == 1).sum()
print(f"Number of anomalies in the dataset: {num_anomalies}")
print(f"Number of normal points in the dataset: {num_normals}")

# Define a jitter function
def add_jitter(arr, jitter_amount=0.01):
    return arr + np.random.uniform(-jitter_amount, jitter_amount, arr.shape)

# Apply jitter to the data
X_jittered = X.copy()
X_jittered['Number of Medicare Beneficiaries'] = add_jitter(X_jittered['Number␣
 ↪of Medicare Beneficiaries'])
X_jittered['Number of Services'] = add_jitter(X_jittered['Number of Services'])

# Visualize the results with jitter
plt.figure(figsize=(10, 6))

# Plot the normal points with jitter
plt.scatter(X_jittered['Number of Medicare Beneficiaries'], X_jittered['Number␣
 ↪of Services'],
            c='white', edgecolors='k', s=50, label='Normal Points')

# Plot the anomalies with jitter
plt.scatter(X_jittered[data['Anomaly'] == -1]['Number of Medicare␣
 ↪Beneficiaries'],
            X_jittered[data['Anomaly'] == -1]['Number of Services'],
```

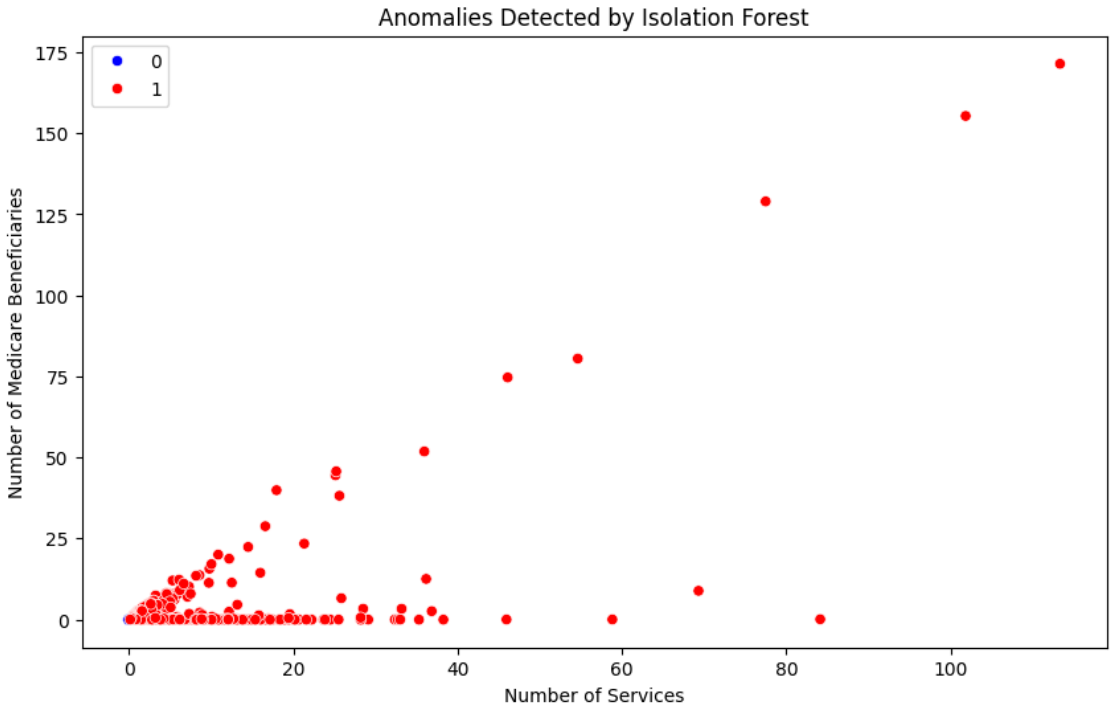
```
            c='cyan', edgecolors='k', s=20, label='Anomalies')

plt.title('Anomaly Detection using Elliptic Envelope (with Jitter)')
plt.xlabel('Number of Medicare Beneficiaries')
plt.ylabel('Number of Services')
plt.legend()
plt.show()
```

```
Number of anomalies in the dataset: 10000
Number of normal points in the dataset: 90000
```



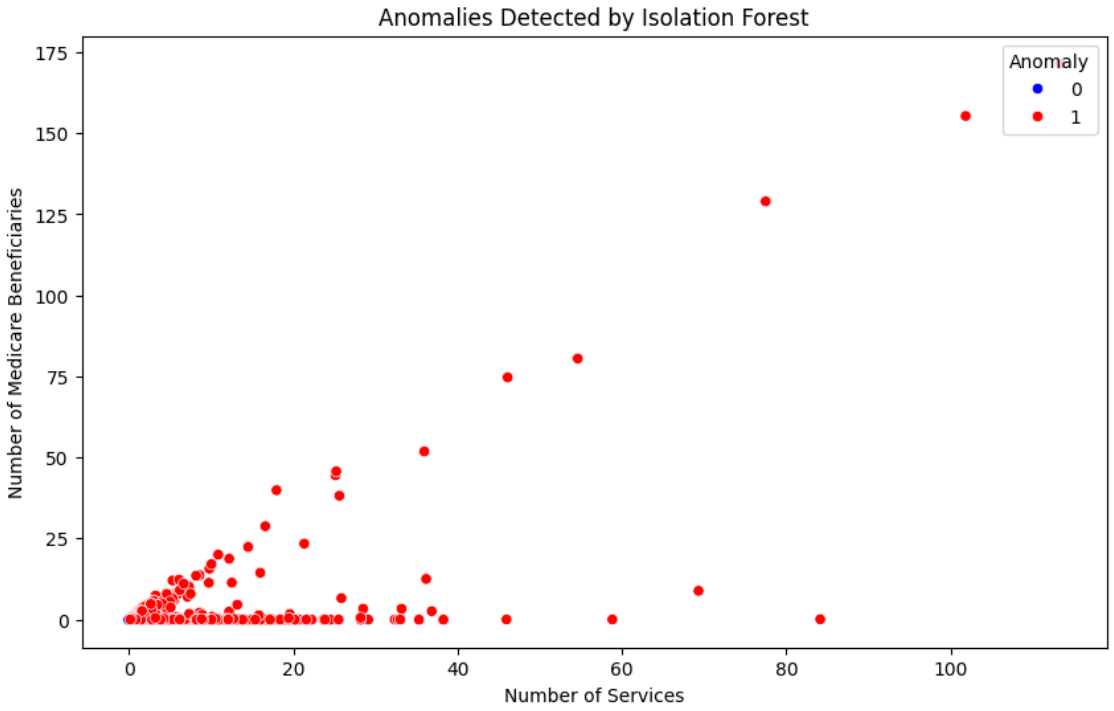
[48]:
```
# Plot the scatter plot for Isolation Forest
plt.figure(figsize=(10, 6))
sns.scatterplot(
    x='Number of Services',
    y='Number of Medicare Beneficiaries',
    hue='anomaly_iso',
    data=data,
    palette={0: 'blue', 1: 'red'},  # Assigning colors to 0 and 1
    hue_order=[0, 1]  # Specify the order of hue values for correct labeling
)
plt.xlabel('Number of Services')
plt.ylabel('Number of Medicare Beneficiaries')
plt.title('Anomalies Detected by Isolation Forest')
```

```
# Customize legend labels and colors
plt.legend(title='Anomaly', loc='upper right', labels=['Normal', 'Anomaly'],␣
 ↪markerscale=2, fontsize='large')

# Set legend label colors
legend = plt.legend()
for text in legend.get_texts():
    if text.get_text() == 'Normal':
        text.set_color('blue')
    elif text.get_text() == 'Anomaly':
        text.set_color('red')

plt.show()
```



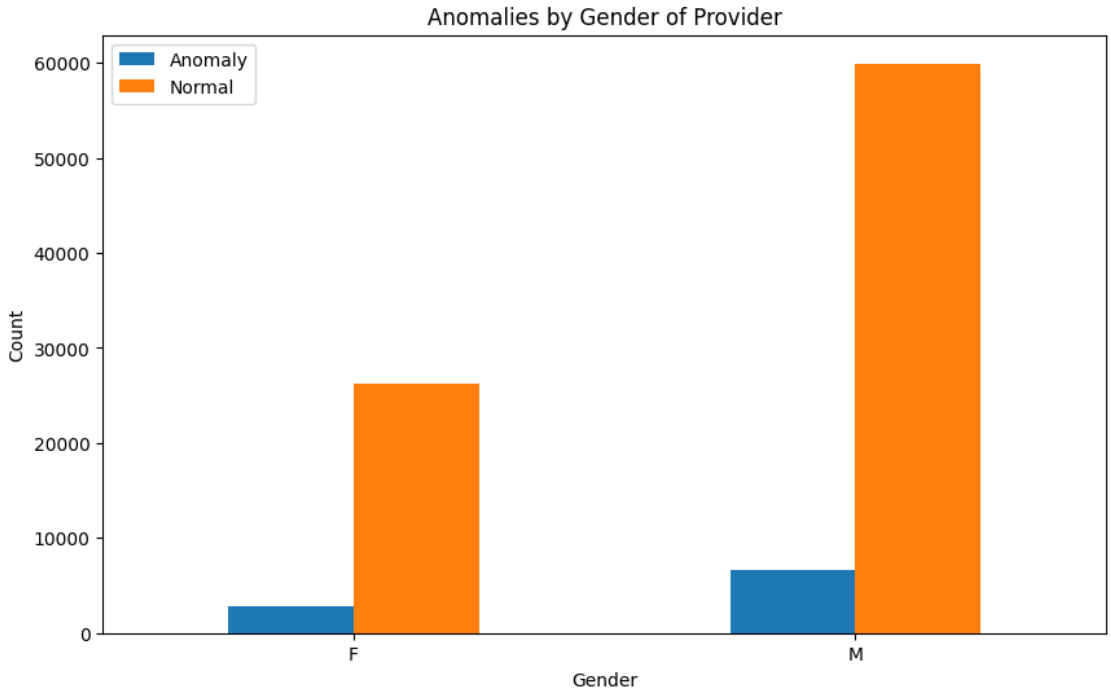
[41]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Plot the scatter plot for Isolation Forest
plt.figure(figsize=(10, 6))
sns.scatterplot(
    x='Number of Services',
    y='Number of Medicare Beneficiaries',
    hue='anomaly_iso',
    data=data,
```

```
    palette={0: 'blue', 1: 'red'}  # corrected the dictionary syntax
)

plt.xlabel('Number of Services')
plt.ylabel('Number of Medicare Beneficiaries')
plt.title('Anomalies Detected by Isolation Forest')
plt.legend(title='Anomaly', loc='upper right')
plt.show()
```



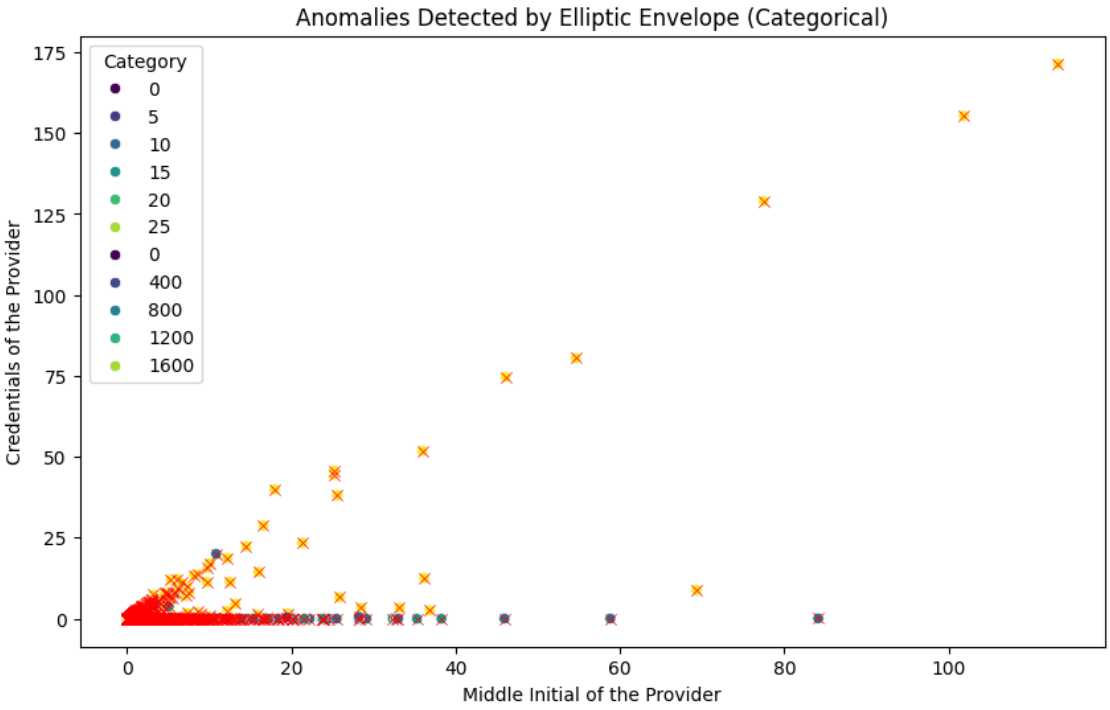
[14]:
```
# Assuming columns 'Gender' and 'Anomaly' exist
gender_anomaly_counts = data.groupby(['Gender of the Provider', 'Anomaly']).
  ↪size().unstack(fill_value=0)
```

[17]:
```
import matplotlib.pyplot as plt

gender_anomaly_counts.plot(kind='bar', figsize=(10, 6))

plt.title('Anomalies by Gender of Provider')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.xticks(rotation=0)
plt.legend(title='Anomaly')
plt.legend(['Anomaly', 'Normal'])
plt.show()
```



[21]:
```
import pandas as pd
import matplotlib.pyplot as plt
# Load the dataset
data = pd.read_csv("/content/Health1.csv")

# Select the categorical columns
categorical_cols = ["Middle Initial of the Provider", "Credentials of the␣
 ↪Provider"]

# Encode the categorical columns
label_encoders = {}
for col in categorical_cols:
    le = LabelEncoder()
    data[col] = le.fit_transform(data[col])
    label_encoders[col] = le

# Select the numerical columns
numerical_cols = ['Number of Services', 'Number of Medicare Beneficiaries']

# Remove commas and convert to numeric
for col in numerical_cols:
    data[col] = data[col].str.replace(',', '').astype(float)

# Scale numerical features
```

```
scaler = StandardScaler()
data[numerical_cols] = scaler.fit_transform(data[numerical_cols])

# Initialize and fit the model
elliptic_env = EllipticEnvelope(contamination=0.1, random_state=42)
elliptic_env.fit(data[numerical_cols])

# Predict anomalies
data['anomaly'] = elliptic_env.predict(data[numerical_cols])
data['anomaly'] = data['anomaly'].map({1: 0, -1: 1})

# Create a scatter plot for Elliptic Envelope, colored by categorical columns
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Number of Services', y='Number of Medicare Beneficiaries',␣
 ↪hue=data['Middle Initial of the Provider'], data=data, palette='viridis')
sns.scatterplot(x='Number of Services', y='Number of Medicare Beneficiaries',␣
 ↪hue=data['Credentials of the Provider'], data=data, palette='viridis')

# Add anomaly points with different marker
sns.scatterplot(x=data[data['anomaly'] == 1]['Number of Services'],␣
 ↪y=data[data['anomaly'] == 1]['Number of Medicare Beneficiaries'],␣
 ↪color='red', marker='x')

plt.xlabel('Middle Initial of the Provider')
plt.ylabel('Credentials of the Provider')
plt.title('Anomalies Detected by Elliptic Envelope (Categorical)')
plt.legend(title='Category')
plt.show()
```


Anomalies Detected by Elliptic Envelope (Categorical)
Category
0
5
10
15
20
25
0
400
800
1200
1600
Credentials of the Provider
Middle Initial of the Provider
175
150
125
100
75
50
25
0
0
20
40
60
80
100