

Anomaly Detection using Autoencoders

This will be conducted on the same healthcare dataset that we feature engineered. I focus on numeric features tied to payments and peer comparison, then standardize before modeling. I train two autoencoder algorithms from PyOD and compare their results.

```
In [1]: # import necessary libraries
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from IPython.display import display
from sklearn.preprocessing import StandardScaler

from pyod.models.auto_encoder import AutoEncoder
from pyod.models.vae import VAE
from pyod.models.auto_encoder import AutoEncoder as AEModel

%matplotlib inline
```

```
In [2]: # Load and clean data

df = pd.read_csv("/users/tiffanytruong/Documents/APAN5420/healthcare_data.csv")
df.head()
```

Out [2]:

	DRG	Provider_Id	Provider_Name	Provider_StreetAddress	Provider_City
0	039 - EXTRACRANIAL PROCEDURES W/O CC/MCC	10001	SOUTHEAST ALABAMA MEDICAL CENTER	1108 ROSS CLARK CIRCLE	DOTHAN
1	039 - EXTRACRANIAL PROCEDURES W/O CC/MCC	10005	MARSHALL MEDICAL CENTER SOUTH	2505 U S HIGHWAY 431 NORTH	BOAZ
2	039 - EXTRACRANIAL PROCEDURES W/O CC/MCC	10006	ELIZA COFFEE MEMORIAL HOSPITAL	205 MARENGO STREET	FLORENCE
3	039 - EXTRACRANIAL PROCEDURES W/O CC/MCC	10011	ST VINCENT'S EAST	50 MEDICAL PARK EAST DRIVE	BIRMINGHAM
4	039 - EXTRACRANIAL PROCEDURES W/O CC/MCC	10016	SHELBY BAPTIST MEDICAL CENTER	1000 FIRST STREET NORTH	ALABASTER



1. Prep the data

```
In [3]: # select numeric features used for anomaly detection
numeric_cols = [
    "Total_Discharges",
    "Average_Total_Payments",
    "Average_Medicare_Payment",
    "Pct_Medicare_Payment",
    "Peer_Avg_Payment",
    "Deviation_From_Peer_Avg_Pct"
]

# validate required columns exist
missing_cols = [c for c in numeric_cols if c not in df.columns]
if missing_cols:
    raise ValueError(f"Missing expected columns: {missing_cols}")

# prepare numeric matrix and impute with medians
X = df[numeric_cols].copy()
X = X.fillna(X.median(numeric_only=True))

# scale features since autoencoders are sensitive to magnitude
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

```
In [4]: # select 4 key numeric features for visualization
selected_cols = [
```

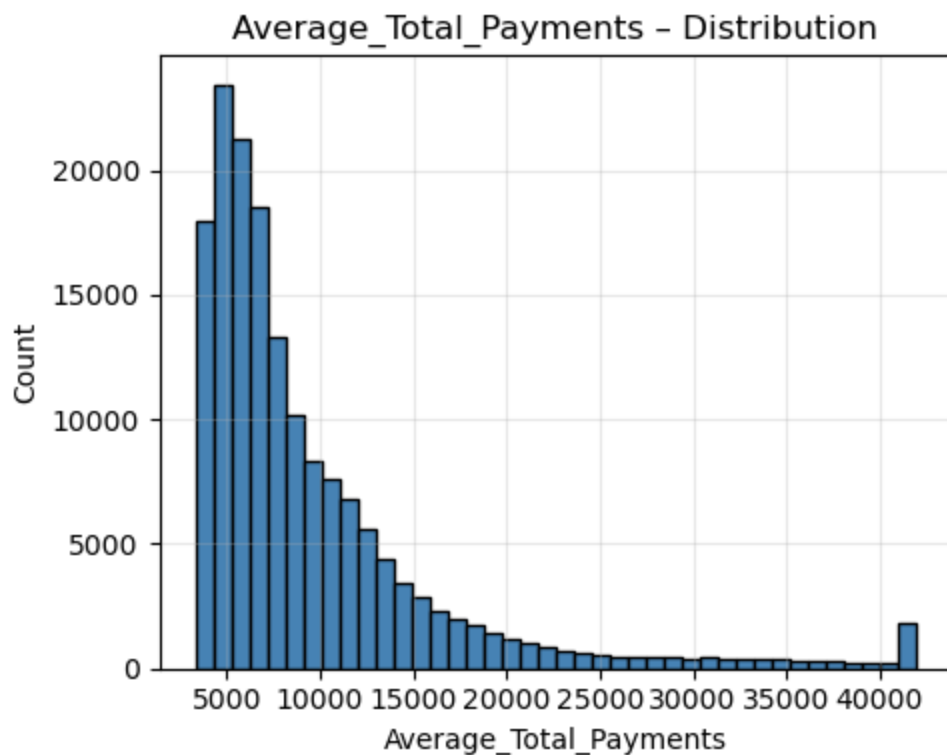
```

    "Average_Total_Payments",
    "Average_Medicare_Payment",
    "Pct_Medicare_Payment",
    "Deviation_From_Peer_Avg_Pct"
]

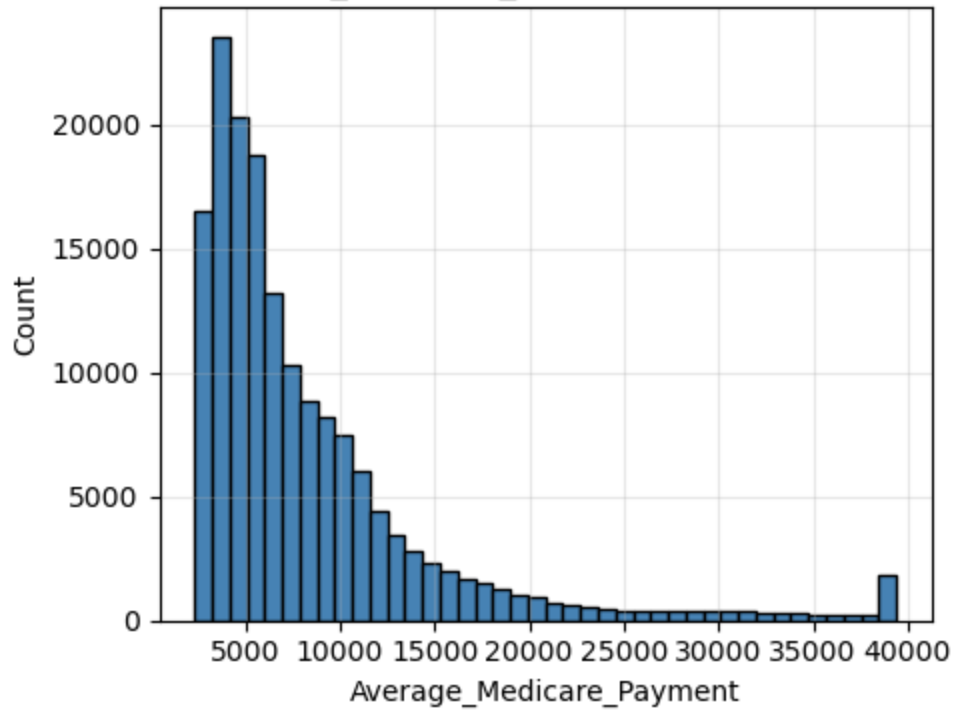
# Clean and plot distributions for selected features
plot_df = df[selected_cols].replace([np.inf, -np.inf], np.nan).dropna()

for col in selected_cols:
    plt.figure(figsize=(5, 4))
    # Clip extreme outliers for better visualization
    series = plot_df[col].clip(
        lower=plot_df[col].quantile(0.01),
        upper=plot_df[col].quantile(0.99)
    )
    plt.hist(series, bins=40, color='steelblue', edgecolor='black')
    plt.title(f"{col} - Distribution")
    plt.xlabel(col)
    plt.ylabel("Count")
    plt.grid(True, alpha=0.3)
    plt.tight_layout()
    plt.show()

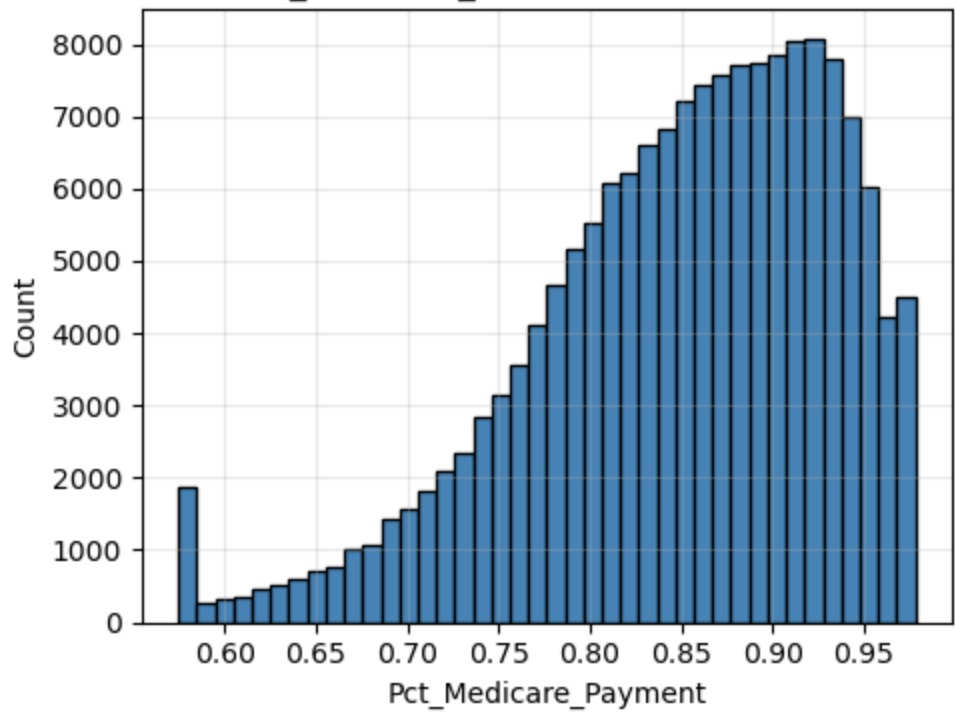
```

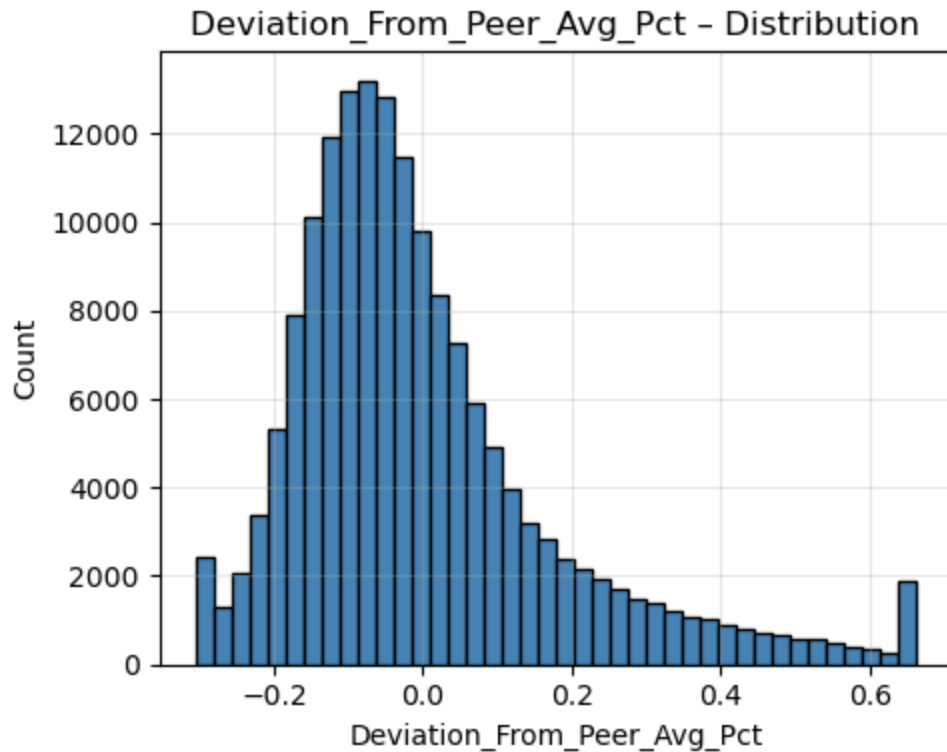


Average_Medicare_Payment - Distribution



Pct_Medicare_Payment - Distribution





2. AutoEncoder (PyOD)

An autoencoder learns to compress each record into a small latent vector and then reconstruct it. If a record reconstructs poorly, the reconstruction error is high and we mark it as anomalous. I use a compact symmetric network to encourage useful compression.

```
In [5]: # fit AutoEncoder
ae = AutoEncoder(
    hidden_neuron_list=[6, 3, 3, 6], # simple symmetric architecture
    contamination=0.05,               # target about 5% anomalies
    verbose=0,
    random_state=42
)
ae.fit(X_scaled)
```

```
In [6]: # labels and scores
ae_labels = ae.labels_           # 0 normal, 1 anomaly
ae_scores = ae.decision_scores_  # higher means more anomalous

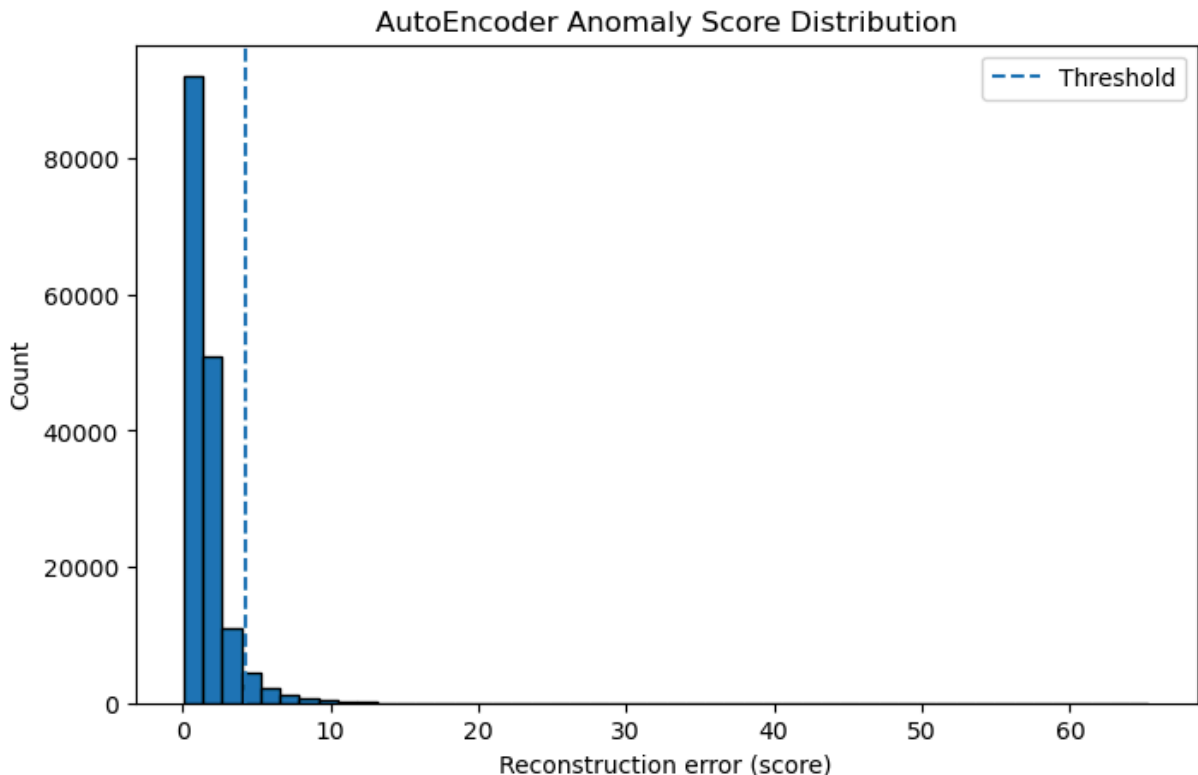
# attach results
df["AE_Anomaly_Flag"] = ae_labels
df["AE_Anomaly_Score"] = ae_scores

# size in count and %
n_total = len(df)
n_ae = int(df["AE_Anomaly_Flag"].sum())
pct_ae = 100.0 * n_ae / n_total if n_total else 0.0
```

```
print(f"[AE] anomalies: {n_ae:,} of {n_total:,} ({pct_ae:.2f}%)")
print("AutoEncoder learned threshold:", getattr(ae, "threshold_", None))
```

[AE] anomalies: 8,154 of 163,065 (5.00%)
AutoEncoder learned threshold: 4.240203285217285

```
In [7]: # Histogram of AE scores
plt.figure(figsize=(8,5))
plt.hist(ae_scores, bins=50, edgecolor="black")
thr = getattr(ae, "threshold_", None)
if thr is not None:
    plt.axvline(thr, linestyle="--", label="Threshold")
plt.title("AutoEncoder Anomaly Score Distribution")
plt.xlabel("Reconstruction error (score)")
plt.ylabel("Count")
if thr is not None:
    plt.legend()
plt.show()
```



Looking at the AutoEncoder anomaly score histogram, I notice that most hospitals have relatively low scores, while a small group extends far to the right. This tail represents cases that the model found harder to reconstruct, meaning their patterns differ from the majority of the data. The clear right skew suggests that the model is able to separate normal payment behavior from unusual or extreme cases effectively.

```
In [8]: # top flagged rows for AE
id_cols = [c for c in ["DRG", "Provider_Id", "Provider_Name", "Provider_State",
ae_cols_show = id_cols + numeric_cols + ["AE_Anomaly_Score"]

top_ae = (
```

```
df[df["AE_Anomaly_Flag"] == 1]
  .sort_values("AE_Anomaly_Score", ascending=False)
)

print("Top 10 AE anomalies")
display(top_ae[ae_cols_show].head(10))
```

Top 10 AE anomalies

	DRG	Provider_Id	Provider_Name	Provider_State	Hospital_referral_
	470 - MAJOR JOINT REPLACEMENT OR REATTACHMENT ...	330270	HOSPITAL FOR SPECIAL SURGERY	NY	N'
112969					
20388	189 - PULMONARY EDEMA & RESPIRATORY FAILURE	390096	ST JOSEPH MEDICAL CENTER	PA	
38923	203 - BRONCHITIS & ASTHMA W/O CC/MCC	220008	STURDY MEMORIAL HOSPITAL	MA	R
157581	885 - PSYCHOSES	50024	PARADISE VALLEY HOSPITAL	CA	C
158518	897 - ALCOHOL/DRUG ABUSE OR DEPENDENCE W/O REH...	220062	ADCARE HOSPITAL OF WORCESTER INC	MA	M
39562	207 - RESPIRATORY SYSTEM DIAGNOSIS W VENTILATO...	50441	STANFORD HOSPITAL	CA	CA - San
109462	460 - SPINAL FUSION EXCEPT CERVICAL W/O MCC	150166	PINNACLE HOSPITAL	IN	
155572	885 - PSYCHOSES	140180	ST MARY & ELIZABETH MED CTR- DIVISION CAMPUS	IL	
112450	470 - MAJOR JOINT REPLACEMENT OR REATTACHMENT ...	220088	NEW ENGLAND BAPTIST HOSPITAL	MA	
150793	853 - INFECTIOUS & PARASITIC DISEASES W O.R. P...	330234	WESTCHESTER MEDICAL CENTER	NY	NY

To check face validity, I compare summary statistics between the normal and anomalous groups for the key attributes. I expect anomalies to show higher payments and larger deviations from peer averages.

```
In [9]: # Summary stats by group for AE
print("Summary stats for normal vs anomalous groups (AE)")

desc_normal_ae = X[ae_labels==0].describe()
desc_anom_ae    = X[ae_labels==1].describe()

print("\nNormal group:")
display(desc_normal_ae)

print("\nAnomalous group:")
display(desc_anom_ae)
```

Summary stats for normal vs anomalous groups (AE)

Normal group:

	Total_Discharges	Average_Total_Payments	Average_Medicare_Payment	Pct_Mec
count	154911.000000	154911.000000	154911.000000	
mean	40.155567	8580.705081	7448.509032	
std	36.211927	4818.750293	4615.480525	
min	11.000000	2673.000000	1594.500000	
25%	17.000000	5151.945000	4114.930000	
50%	27.000000	7002.610000	5960.610000	
75%	49.000000	10609.265000	9498.380000	
max	261.000000	34067.000000	31152.850000	

Anomalous group:

	Total_Discharges	Average_Total_Payments	Average_Medicare_Payment	Pct_Mec
count	8154.000000	8154.000000	8154.000000	
mean	92.565489	31114.006745	28366.223494	
std	157.190094	15854.003910	15751.566496	
min	11.000000	3006.010000	1148.900000	
25%	16.000000	16661.355000	13275.227500	
50%	25.000000	32835.400000	30521.600000	
75%	57.000000	39915.095000	37277.345000	
max	3383.000000	156158.180000	154620.810000	

The AE anomalous group shows higher total payments and larger positive deviations from peer averages. That matches the behavior I expect from unusual cost patterns.

3. VAE

A VAE learns a probabilistic latent space and reconstructs from a distribution. This can surface subtle irregularities because it models both mean and variance in the latent factors.

```
In [10]: # fit VAE
vae = VAE(
    contamination=0.05,
    verbose=0,
    random_state=42
)
vae.fit(X_scaled)
```

```
In [11]: # labels and scores
vae_labels = vae.labels_
vae_scores = vae.decision_scores_

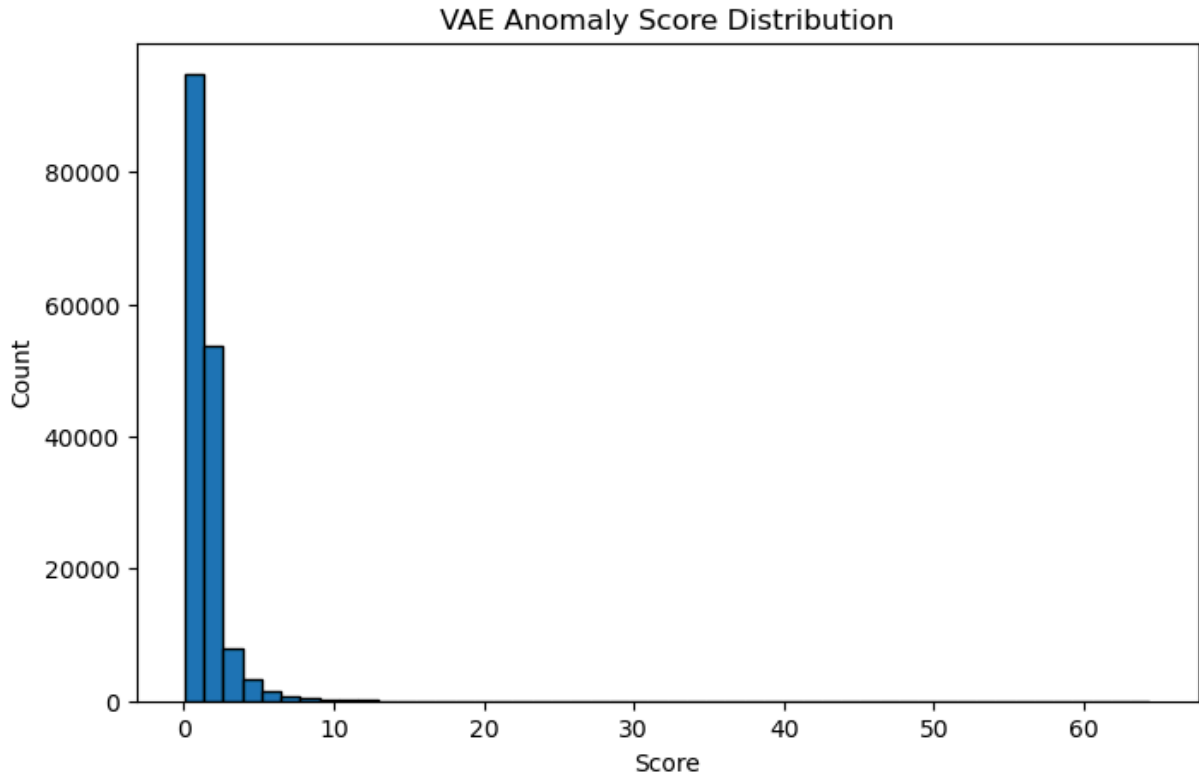
df["VAE_Anomaly_Flag"] = vae_labels
df["VAE_Anomaly_Score"] = vae_scores

n_vae = int(df["VAE_Anomaly_Flag"].sum())
pct_vae = 100.0 * n_vae / n_total if n_total else 0.0
print(f"[VAE] anomalies: {n_vae:,} of {n_total:,} ({pct_vae:.2f}%)")
```

[VAE] anomalies: 8,154 of 163,065 (5.00%)

```
In [12]: # histogram of VAE scores
plt.figure(figsize=(8,5))
plt.hist(vae_scores, bins=50, edgecolor="black")
plt.title("VAE Anomaly Score Distribution")
plt.xlabel("Score")
```

```
plt.ylabel("Count")
plt.show()
```



The VAE score distribution shows a similar pattern but appears smoother than the standard AutoEncoder. Most records have low anomaly scores, with a small portion forming a long right tail. This indicates that both models are identifying roughly the same high-cost or irregular cases. The similarity between them gives me more confidence that the anomalies are meaningful rather than random.

```
In [13]: # top flagged rows for VAE
vae_cols_show = id_cols + numeric_cols + ["VAE_Anomaly_Score"]

top_vae = (
    df[df["VAE_Anomaly_Flag"] == 1]
    .sort_values("VAE_Anomaly_Score", ascending=False)
)

print("Top 10 VAE anomalies")
display(top_vae[vae_cols_show].head(10))
```

Top 10 VAE anomalies

	DRG	Provider_Id	Provider_Name	Provider_State	Hospital_referral_
112969	470 - MAJOR JOINT REPLACEMENT OR REATTACHMENT ...	330270	HOSPITAL FOR SPECIAL SURGERY	NY	N'
20388	189 - PULMONARY EDEMA & RESPIRATORY FAILURE	390096	ST JOSEPH MEDICAL CENTER	PA	
38923	203 - BRONCHITIS & ASTHMA W/O CC/MCC	220008	STURDY MEMORIAL HOSPITAL	MA	R
157581	885 - PSYCHOSES	50024	PARADISE VALLEY HOSPITAL	CA	C
158518	897 - ALCOHOL/DRUG ABUSE OR DEPENDENCE W/O REH...	220062	ADCARE HOSPITAL OF WORCESTER INC	MA	M
39562	207 - RESPIRATORY SYSTEM DIAGNOSIS W VENTILATO...	50441	STANFORD HOSPITAL	CA	CA - San
155572	885 - PSYCHOSES	140180	ST MARY & ELIZABETH MED CTR- DIVISION CAMPUS	IL	
109462	460 - SPINAL FUSION EXCEPT CERVICAL W/O MCC	150166	PINNACLE HOSPITAL	IN	
112450	470 - MAJOR JOINT REPLACEMENT OR REATTACHMENT ...	220088	NEW ENGLAND BAPTIST HOSPITAL	MA	
102239	392 - ESOPHAGITIS, GASTROENT & MISC DIGEST DIS...	100007	FLORIDA HOSPITAL	FL	

```
In [14]: # summary stats by group for VAE
print("\nSummary stats for normal vs anomalous groups (VAE)")
desc_normal_vae = X[vae_labels == 0].describe()
desc_anom_vae    = X[vae_labels == 1].describe()

print("\nNormal group:")
display(desc_normal_vae)

print("\nAnomalous group:")
display(desc_anom_vae)
```

Summary stats for normal vs anomalous groups (VAE)

Normal group:

	Total_Discharges	Average_Total_Payments	Average_Medicare_Payment	Pct_Mec
count	154911.000000	154911.000000	154911.000000	
mean	40.456927	8660.172871	7532.618322	
std	36.905674	4945.811109	4731.250043	
min	11.000000	2673.000000	1594.500000	
25%	17.000000	5167.195000	4143.970000	
50%	27.000000	7032.730000	5998.060000	
75%	49.000000	10686.815000	9569.310000	
max	272.000000	35648.890000	33030.020000	

Anomalous group:

	Total_Discharges	Average_Total_Payments	Average_Medicare_Payment	Pct_Mec
count	8154.000000	8154.000000	8154.000000	
mean	86.840201	29604.264935	26768.301710	
std	155.918627	17129.608980	17079.872494	
min	11.000000	2868.580000	1148.900000	
25%	15.000000	13117.045000	10283.387500	
50%	24.000000	32705.995000	30372.055000	
75%	48.000000	39915.095000	37277.345000	
max	3383.000000	156158.180000	154620.810000	

4. Agreement and union

Here, I will compare agreement between AE and VAE. I will also create a union set that flags a row if either model marks it as anomalous. This supports a conservative review list for audit.

```
In [15]: agreement = (ae_labels == vae_labels).mean() * 100.0
union_flags = ((ae_labels == 1) | (vae_labels == 1)).astype(int)

df["Anomaly_Flag_Union"] = union_flags
df["Anomaly_Score_Max"] = np.maximum(ae_scores, vae_scores)

n_union = int(df["Anomaly_Flag_Union"].sum())
pct_union = 100.0 * n_union / n_total if n_total else 0.0
```

```
In [16]: print(f"AE vs VAE agreement: {agreement:.2f}%")
print(f"[Union] anomalies: {n_union:}, of {n_total:}, ({pct_union:.2f}%")
```

```
AE vs VAE agreement: 99.04%
[Union] anomalies: 8,934 of 163,065 (5.48%)
```

```
In [17]: print("\nTop 10 union anomalies by max score")
union_cols_show = id_cols + numeric_cols + ["Anomaly_Score_Max"]
display(
    df[df["Anomaly_Flag_Union"] == 1]
      .sort_values("Anomaly_Score_Max", ascending=False)[union_cols_show]
      .head(10)
)
```

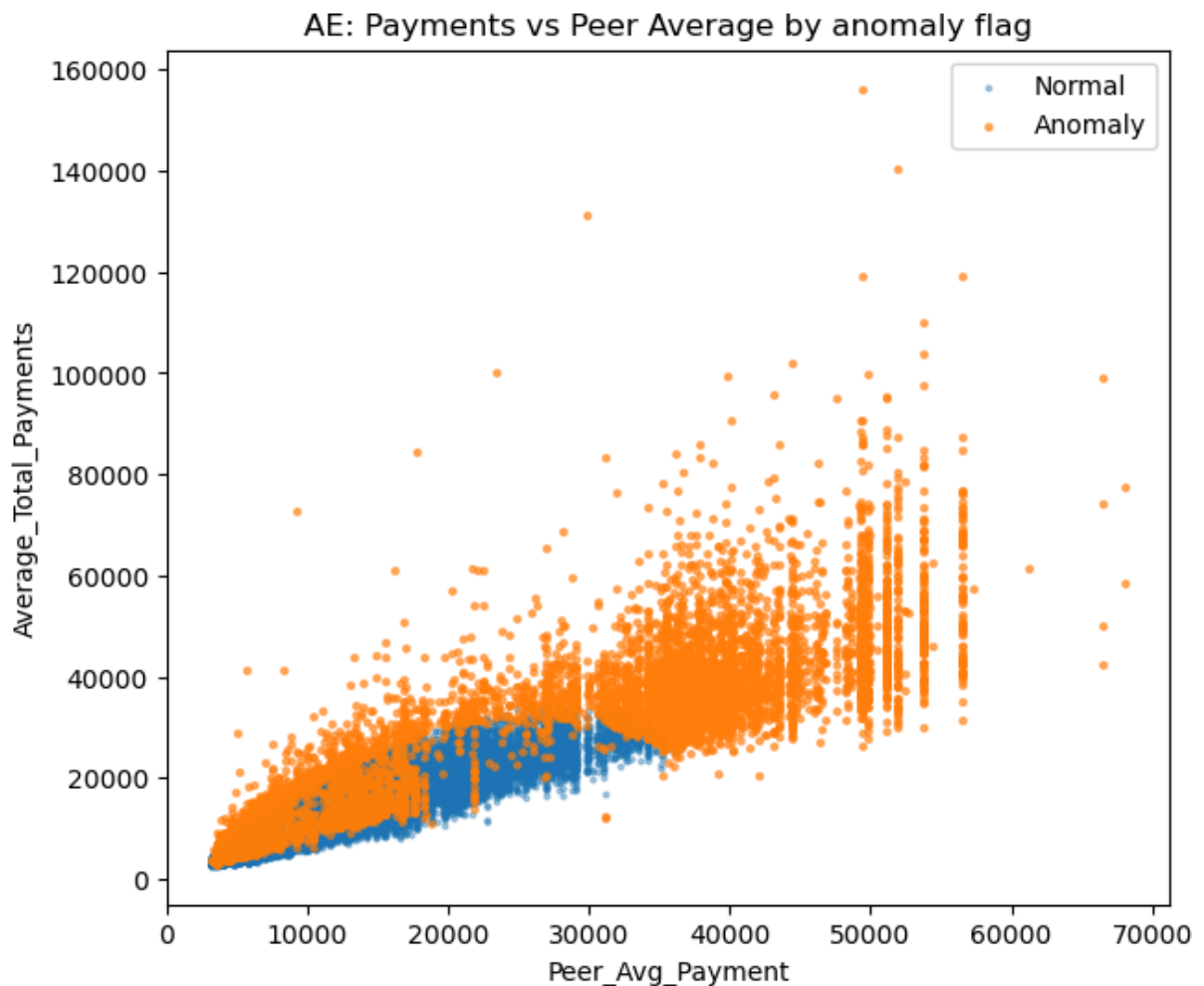
Top 10 union anomalies by max score

	DRG	Provider_Id	Provider_Name	Provider_State	Hospital_referral
112969	470 - MAJOR JOINT REPLACEMENT OR REATTACHMENT ...	330270	HOSPITAL FOR SPECIAL SURGERY	NY	N'
20388	189 - PULMONARY EDEMA & RESPIRATORY FAILURE	390096	ST JOSEPH MEDICAL CENTER	PA	
38923	203 - BRONCHITIS & ASTHMA W/O CC/MCC	220008	STURDY MEMORIAL HOSPITAL	MA	R
157581	885 - PSYCHOSES	50024	PARADISE VALLEY HOSPITAL	CA	C
158518	897 - ALCOHOL/DRUG ABUSE OR DEPENDENCE W/O REH...	220062	ADCARE HOSPITAL OF WORCESTER INC	MA	M
39562	207 - RESPIRATORY SYSTEM DIAGNOSIS W VENTILATO...	50441	STANFORD HOSPITAL	CA	CA - San
109462	460 - SPINAL FUSION EXCEPT CERVICAL W/O MCC	150166	PINNACLE HOSPITAL	IN	
155572	885 - PSYCHOSES	140180	ST MARY & ELIZABETH MED CTR-DIVISION CAMPUS	IL	
112450	470 - MAJOR JOINT REPLACEMENT OR REATTACHMENT ...	220088	NEW ENGLAND BAPTIST HOSPITAL	MA	
150793	853 - INFECTIOUS & PARASITIC DISEASES W O.R. P...	330234	WESTCHESTER MEDICAL CENTER	NY	NY

5. Where are the outliers?

To justify the detections, I will plot payments versus peer averages and compare group distributions. I will also compute mean tables for key attributes.

```
In [18]: # scatter: payments vs peer average, colored by AE anomaly flag
plt.figure(figsize=(7,6))
plt.scatter(
    df.loc[df["AE_Anomaly_Flag"]==0, "Peer_Avg_Payment"],
    df.loc[df["AE_Anomaly_Flag"]==0, "Average_Total_Payments"],
    s=4, alpha=0.35, label="Normal"
)
plt.scatter(
    df.loc[df["AE_Anomaly_Flag"]==1, "Peer_Avg_Payment"],
    df.loc[df["AE_Anomaly_Flag"]==1, "Average_Total_Payments"],
    s=6, alpha=0.6, label="Anomaly"
)
plt.xlabel("Peer_Avg_Payment")
plt.ylabel("Average_Total_Payments")
plt.title("AE: Payments vs Peer Average by anomaly flag")
plt.legend()
plt.show()
```



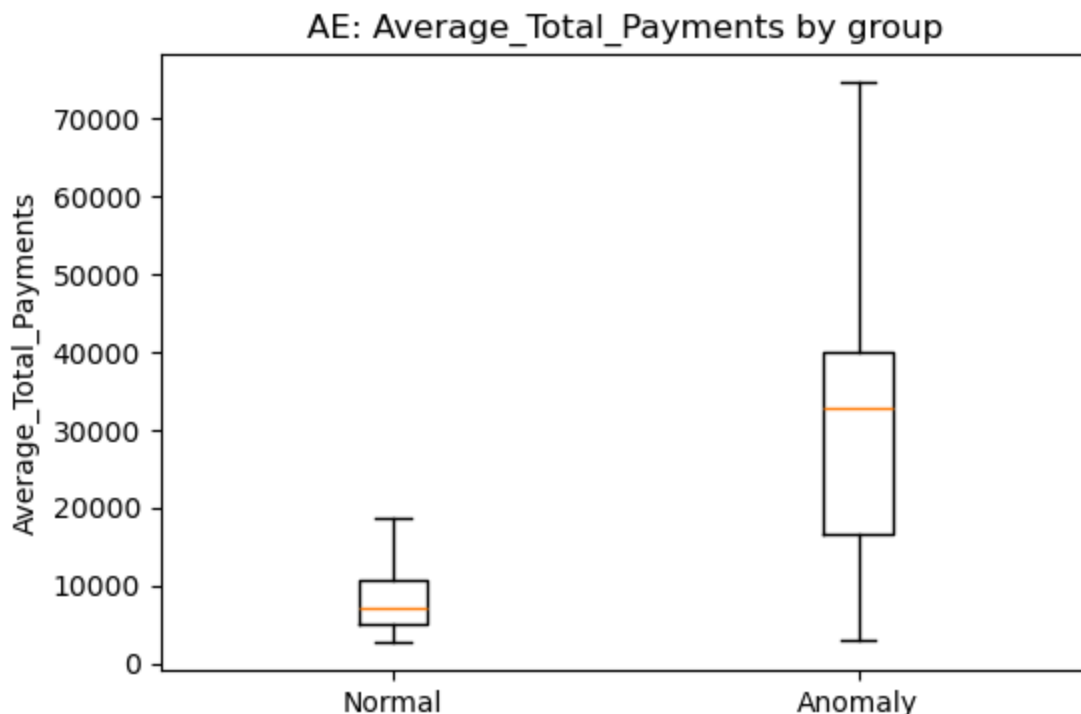
In the scatter plot, I can see that most hospitals fall along a diagonal line where average total payments align closely with peer averages. The anomalous points, however, appear well above this line, meaning their total payments are much higher compared to what their peers typically charge. These outliers might represent hospitals with unusually high treatment costs or possible billing inconsistencies that should be reviewed more closely.

```
In [19]: # Boxplots to visualize shifts
def box_by_flag(col, flag_col, title):
    plt.figure(figsize=(6,4))
    data_normal = df.loc[df[flag_col]==0, col].dropna()
    data_anomaly = df.loc[df[flag_col]==1, col].dropna()
    plt.boxplot([data_normal, data_anomaly], labels=["Normal","Anomaly"], sh
    plt.title(title)
    plt.ylabel(col)
    plt.show()

box_by_flag("Average_Total_Payments", "AE_Anomaly_Flag", "AE: Average_Total_
```

```
/var/folders/xl/nbpgryh9421_0f0f6ks04j000000gn/T/ipykernel_77498/367450754.p
y:6: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() has b
een renamed 'tick_labels' since Matplotlib 3.9; support for the old name wil
l be dropped in 3.11.
```

```
plt.boxplot([data_normal, data_anomaly], labels=["Normal","Anomaly"], show
fliers=False)
```

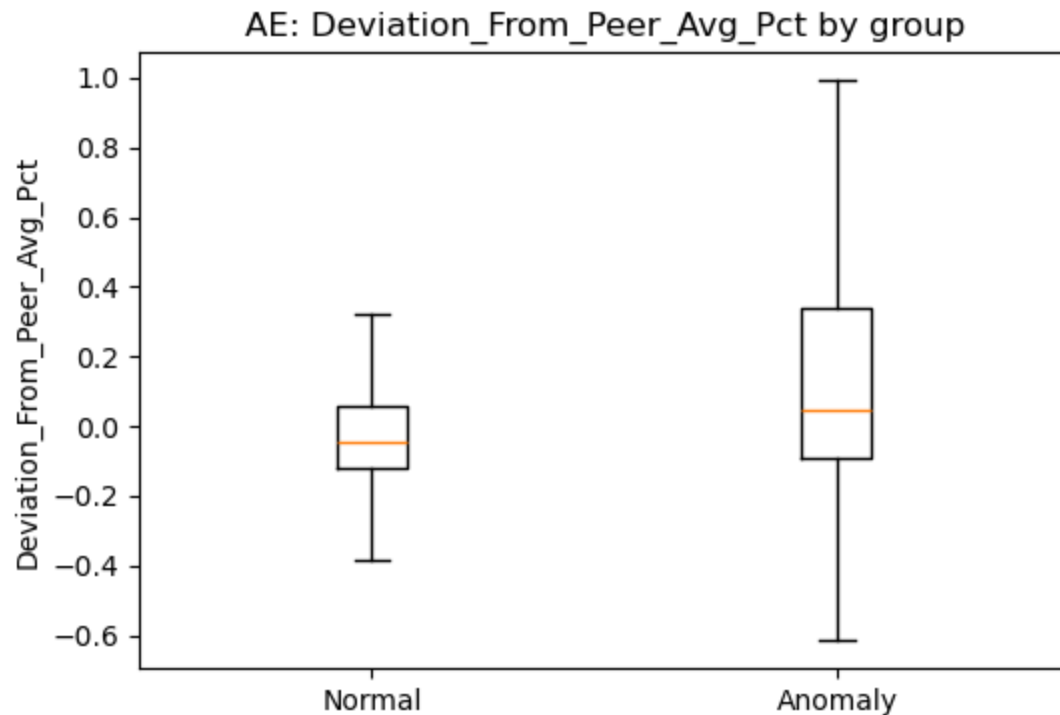


The boxplot comparing normal and anomalous groups shows that the anomaly group has a much higher median total payment and a wider spread. This tells me that the flagged hospitals consistently pay or charge more than the rest. It supports the model's detection logic since unusually high payments are a common indicator of potential inefficiency or overpricing.

```
In [20]: box_by_flag("Deviation_From_Peer_Avg_Pct", "AE_Anomaly_Flag", "AE: Deviation
```

```
/var/folders/xl/nbpgryh9421_0f0f6ks04j000000gn/T/ipykernel_77498/367450754.p  
y:6: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() has b  
een renamed 'tick_labels' since Matplotlib 3.9; support for the old name wil  
l be dropped in 3.11.
```

```
plt.boxplot([data_normal, data_anomaly], labels=["Normal","Anomaly"], show  
fliers=False)
```

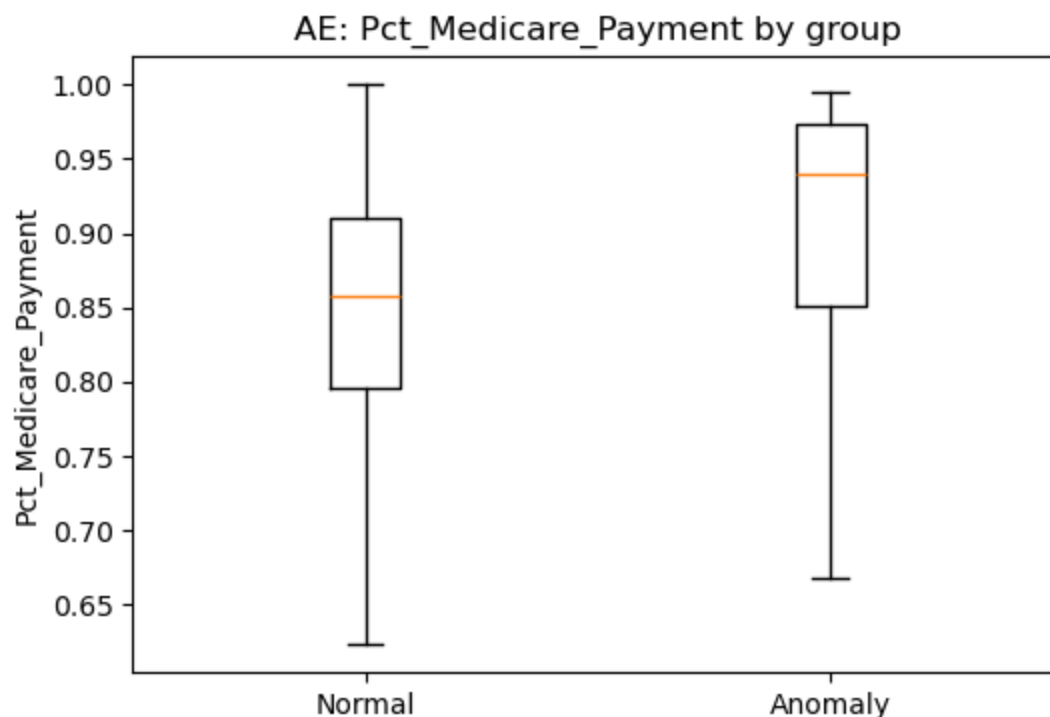


From this boxplot, I can see that anomalies tend to have higher deviations from peer averages. Their values are more spread out and extend further into positive territory. This means the flagged hospitals are not just slightly different but significantly outside the normal payment range for their peers. This gives a clear signal that their cost structures might not align with industry norms.

```
In [21]: box_by_flag("Pct_Medicare_Payment", "AE_Anomaly_Flag", "AE: Pct_Medicare_Pay
```

```
/var/folders/xl/nbpgryh9421_0f0f6ks04j000000gn/T/ipykernel_77498/367450754.p  
y:6: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() has b  
een renamed 'tick_labels' since Matplotlib 3.9; support for the old name wil  
l be dropped in 3.11.
```

```
plt.boxplot([data_normal, data_anomaly], labels=["Normal","Anomaly"], show  
fliers=False)
```



The percentage of Medicare payments varies more in the anomaly group. Some providers have almost full Medicare coverage, while others are much lower than average. These extremes could point to unique patient demographics or specialized treatment programs, but they could also suggest uneven dependency on Medicare reimbursements. This variation makes these hospitals worth examining further.

To quantify the differences, I will compare group means for attributes that are most relevant to operations and finance decisions.

```
In [22]: # Attribute-level justification tables
def mean_table(flag_col, cols):
    rows = []
    for c in cols:
        grp = df.groupby(flag_col)[c].mean()
        rows.append(pd.Series({"Normal_mean": grp.get(0, np.nan), "Anomaly_mean": grp.get(1, np.nan)}))
    tab = pd.DataFrame(rows)
    tab["Diff_Anomaly_minus_Normal"] = tab["Anomaly_mean"] - tab["Normal_mean"]
    return tab

key_cols = [
    "Average_Total_Payments",
    "Average_Medicare_Payment",
    "Deviation_From_Peer_Avg_Pct",
    "Pct_Medicare_Payment",
    "Total_Discharges"
]
```

```
In [23]: print("AE: mean comparison by group")
display(mean_table("AE_Anomaly_Flag", key_cols).style.format("{:,.2f}"))
```

```
print("VAE: mean comparison by group")
display(mean_table("VAE_Anomaly_Flag", key_cols).style.format("{:,.2f}"))
```

AE: mean comparison by group

	Normal_mean	Anomaly_mean	Diff_Anomaly_minus_Norm
Average_Total_Payments	8,580.71	31,114.01	22,533.
Average_Medicare_Payment	7,448.51	28,366.22	20,917.
Deviation_From_Peer_Avg_Pct	-0.01	0.18	0.
Pct_Medicare_Payment	0.84	0.88	0.
Total_Discharges	40.16	92.57	52.

VAE: mean comparison by group

	Normal_mean	Anomaly_mean	Diff_Anomaly_minus_Norm
Average_Total_Payments	8,660.17	29,604.26	20,944.
Average_Medicare_Payment	7,532.62	26,768.30	19,235.
Deviation_From_Peer_Avg_Pct	-0.01	0.16	0.
Pct_Medicare_Payment	0.85	0.85	-0.
Total_Discharges	40.46	86.84	46.

6. Hot spots by DRG and by state

By looking at where anomalies concentrate, it helps prioritize reviews and audits where they can have the biggest impact.

```
In [24]: # DRG anomaly share using AE
drg_counts = df.groupby("DRG", dropna=False).size().rename("total")
drg_anom    = df.groupby("DRG", dropna=False)["AE_Anomaly_Flag"].sum().rename("anom")
drg_share   = pd.concat([drg_counts, drg_anom], axis=1)
drg_share["anom_pct"] = 100.0 * drg_share["anom"] / drg_share["total"]
drg_share = drg_share.sort_values(["anom_pct", "anom"], ascending=[False, False])
print("Top DRGs by AE anomaly percentage (min 50 rows)")
display(drg_share[drg_share["total"]>=50].head(10))
```

Top DRGs by AE anomaly percentage (min 50 rows)

	total	anom	anom_pct
DRG			
870 - SEPTICEMIA OR SEVERE SEPSIS W MV 96+ HOURS	939	939	100.000000
853 - INFECTIOUS & PARASITIC DISEASES W O.R. PROCEDURE W MCC	1376	1344	97.674419
329 - MAJOR SMALL & LARGE BOWEL PROCEDURES W MCC	1476	1352	91.598916
207 - RESPIRATORY SYSTEM DIAGNOSIS W VENTILATOR SUPPORT 96+ HOURS	1163	1034	88.907997
460 - SPINAL FUSION EXCEPT CERVICAL W/O MCC	1332	279	20.945946
470 - MAJOR JOINT REPLACEMENT OR REATTACHMENT OF LOWER EXTREMITY W/O MCC	2750	549	19.963636
885 - PSYCHOSES	613	121	19.738989
871 - SEPTICEMIA OR SEVERE SEPSIS W/O MV 96+ HOURS W MCC	2812	324	11.522048
252 - OTHER VASCULAR PROCEDURES W MCC	1151	99	8.601216
246 - PERC CARDIOVASC PROC W DRUG-ELUTING STENT W MCC OR 4+ VESSELS/STENTS	917	71	7.742639

```
In [25]: # State anomaly share using AE
st_counts = df.groupby("Provider_State", dropna=False).size().rename("total")
st_anom    = df.groupby("Provider_State", dropna=False)["AE_Anomaly_Flag"].sum()
state_share = pd.concat([st_counts, st_anom], axis=1)
state_share["anom_pct"] = 100.0 * state_share["anom"] / state_share["total"]
state_share = state_share.sort_values(["anom_pct", "anom"], ascending=False)
print("Top States by AE anomaly percentage (min 200 rows)")
display(state_share[state_share["total"]>=200].head(10))
```

Top States by AE anomaly percentage (min 200 rows)

	total	anom	anom_pct
Provider_State			
MD	3330	290	8.708709
DE	394	31	7.868020
DC	462	33	7.142857
NY	9178	640	6.973197
CA	13064	881	6.743723
AK	231	15	6.493506
IL	7909	504	6.372487
NJ	4826	285	5.905512
MI	5419	316	5.831334
VA	4332	241	5.563250

The DRG-level results show that certain procedures, like joint replacements and respiratory treatments, appear frequently among anomalies. These are typically high-cost categories, so it makes sense that they stand out. On the state level, I notice that a few regions have higher anomaly percentages than others, which could reflect local cost structures or policy differences. Focusing audits on these specific DRGs and states might reveal valuable cost insights.

7. Hyperparameter tuning

To check model sensitivity, I vary the contamination hyperparameter for the AutoEncoder. Contamination controls the expected outlier fraction. Smaller values create a tighter threshold while larger values flag more points for review. I compare the resulting anomaly sizes to pick a balanced setting.

```
In [26]: def run_ae_with_contamination(contam):
    model = AEModel(
        hidden_neuron_list=[6,3,3,6],
        contamination=contam,
        verbose=0,
        random_state=42
    )
    model.fit(X_scaled)
    labels = model.labels_
    count = int(labels.sum())
    pct = 100.0 * count / n_total if n_total else 0.0
    return {"contamination": contam, "count": count, "pct": pct}

sens = pd.DataFrame([run_ae_with_contamination(c) for c in [0.03, 0.05, 0.07]
```

```
print("Sensitivity of anomaly size to contamination")
display(sens.style.format({"contamination":"{:0.2f}", "count":"{:,}", "pct":
```

Sensitivity of anomaly size to contamination

	contamination	count	pct
0	0.03	4,892	3.00%
1	0.05	8,154	5.00%
2	0.07	11,415	7.00%
3	0.10	16,307	10.00%

When I adjusted the contamination rate, the number of anomalies increased gradually, which makes sense since a higher contamination value tells the model to expect more outliers. The 5 percent setting seems like a good balance because it captures the most extreme cases without overwhelming the analysis with borderline ones. This shows that the chosen parameter is reasonable for this dataset.

8. Summary Stats

```
In [27]: def pct(n, d):
          return 100.0 * n / d if d else 0.0

agreement = (ae_labels == vae_labels).mean() * 100.0
union_count = int(df["Anomaly_Flag_Union"].sum())

print("Outlier sizes")
print(f"- AE:    {n_ae:,} of {n_total:,} ({pct_ae:.2f}%)")
print(f"- VAE:    {n_vae:,} of {n_total:,} ({pct_vae:.2f}%)")
print(f"- Union: {union_count:,} of {n_total:,} ({pct(union_count, n_total):.2f}%)")
print(f"AE vs VAE agreement: {agreement:.2f}%")
```

Outlier sizes

- AE: 8,154 of 163,065 (5.00%)
- VAE: 8,154 of 163,065 (5.00%)
- Union: 8,934 of 163,065 (5.48%)

AE vs VAE agreement: 99.04%

The AutoEncoder and VAE models agree on most of the flagged cases, which shows that the anomalies are consistent across different detection methods. The union of both models slightly increases the total anomalies, providing a conservative list that includes any case identified by either model

9. Business insight

Both detectors identify a small subset of records that behave differently from peer patterns. These records show high average total payments, large positive deviations from peer averages, and unusual distributions of Medicare share in some cases. I recommend prioritizing the DRGs and states with the highest anomaly share, reviewing

contract terms at high volume outlier facilities, and validating fields that drive alerts. Follow up should include chart reviews and coder audits for the top facilities and DRGs to determine if the differences reflect legitimate case mix, negotiated rates, or data quality issues.