xgboost

July 29, 2025

[1]:
```
import shap
import xgboost as xgb
from xgboost import XGBClassifier
from sklearn.metrics import (
    roc_auc_score, roc_curve, accuracy_score,
    precision_score, recall_score, f1_score,
    confusion_matrix, log_loss
)
from sklearn.model_selection import train_test_split
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```
df = pd.read_parquet("/PHI_conf/VaccineUptake/Analysts/Vay/
 ↪vaccinate_uptake_ML_analysis/vaccine_data/master_data/cohort_df_merged.
 ↪parquet")
unique_cohorts = df['cohort_group_ML_analysis'].unique()
unique_cohorts
```

[2]:
```
array(['AGE_50_TO_64', 'AGE_65_TO_74', 'AGE_75_AND_OVER',
       'ALL_HEALTH_CARE_WORKERS', 'ALL_SOCIAL_CARE_WORKERS',
       '18_TO_64_FLU_AT_RISK', 'OLDER_PEOPLE_CARE_HOME',
       'WEAKENED_IMMUNE_SYSTEM'], dtype=object)
```

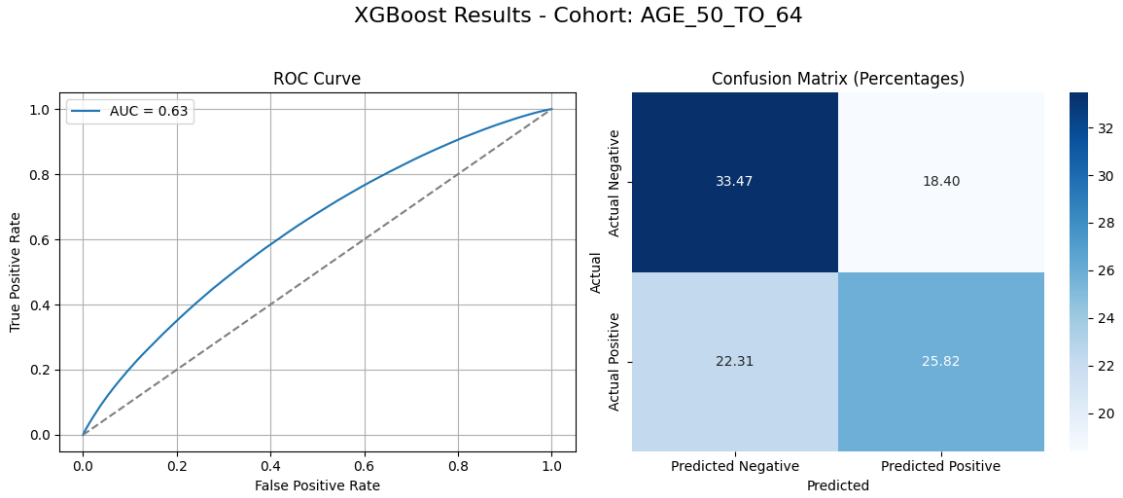
[3]:
```
shap.initjs()

xgb_cross_cohort_summary = {}
xgb_model_scores = {}

for cohort in unique_cohorts:
    print(f"\n\n======== Cohort: {cohort} =========")
    cohort_df = df[df["cohort_group_ML_analysis"] == cohort].copy()

    # Prepare features and target
    X = cohort_df.drop(columns=["cohort_group_ML_analysis",␣
 ↪"attended_vaccination_event"])
```

```
    y = cohort_df["attended_vaccination_event"]

    # Train-test split
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.3, random_state=42, stratify=y
    )
    print("Train size:", len(X_train))
    print("Test size:", len(X_test))

    # Fit XGBoost
    xgb_model = XGBClassifier(
        n_estimators=100,
        max_depth=4,
        learning_rate=0.1,
        use_label_encoder=False,
        eval_metric='logloss',
        random_state=42,
        n_jobs=-1
    )
    xgb_model.fit(X_train, y_train)

    # Predict
    y_prob = xgb_model.predict_proba(X_test)[:, 1]
    y_pred = xgb_model.predict(X_test)

    # ROC & AUC
    fpr, tpr, _ = roc_curve(y_test, y_prob)
    auc_score = roc_auc_score(y_test, y_prob)
    logloss_score = log_loss(y_test, y_prob)

    # Metrics summary
    metrics_summary = pd.DataFrame({
        "Metric": ["Accuracy", "Precision", "Recall", "F1 Score", "AUC Score",␣
↪"Log Loss"],
        "Value": [
            accuracy_score(y_test, y_pred),
            precision_score(y_test, y_pred),
            recall_score(y_test, y_pred),
            f1_score(y_test, y_pred),
            auc_score,
            logloss_score
        ]
    })

    xgb_model_scores[cohort] = {
        "F1 Score": f1_score(y_test, y_pred),
        "AUC Score": auc_score,
```

```
    "Log Loss": logloss_score
}

print("\nModel Performance Metrics:")
print(metrics_summary.to_string(index=False))

# Plot ROC + Confusion Matrix
fig, axes = plt.subplots(1, 2, figsize=(12, 5))

axes[0].plot(fpr, tpr, label=f"AUC = {auc_score:.2f}")
axes[0].plot([0, 1], [0, 1], linestyle="--", color="gray")
axes[0].set_xlabel("False Positive Rate")
axes[0].set_ylabel("True Positive Rate")
axes[0].set_title("ROC Curve")
axes[0].legend()
axes[0].grid(True)

cm = confusion_matrix(y_test, y_pred)
cm_percent = cm.astype('float') / cm.sum() * 100
sns.heatmap(cm_percent, annot=True, fmt='.2f', cmap='Blues',
            xticklabels=['Predicted Negative', 'Predicted Positive'],
            yticklabels=['Actual Negative', 'Actual Positive'],
            ax=axes[1])
axes[1].set_xlabel('Predicted')
axes[1].set_ylabel('Actual')
axes[1].set_title('Confusion Matrix (Percentages)')

fig.suptitle(f"XGBoost Results - Cohort: {cohort}", fontsize=16, y=1.03)
plt.tight_layout()
plt.show()

# Feature Importances
importances = xgb_model.feature_importances_
feature_names = X.columns
importance_df = pd.DataFrame({
    'Feature': feature_names,
    'Importance': importances
}).sort_values(by='Importance', ascending=False)

print("\nTop Feature Importances:")
print(importance_df.round(4).head(10).to_string(index=False))

xgb_cross_cohort_summary[cohort] = dict(zip(
    importance_df['Feature'],
    (importance_df['Importance'] * 100).round(1).astype(str) + "%"
))
```

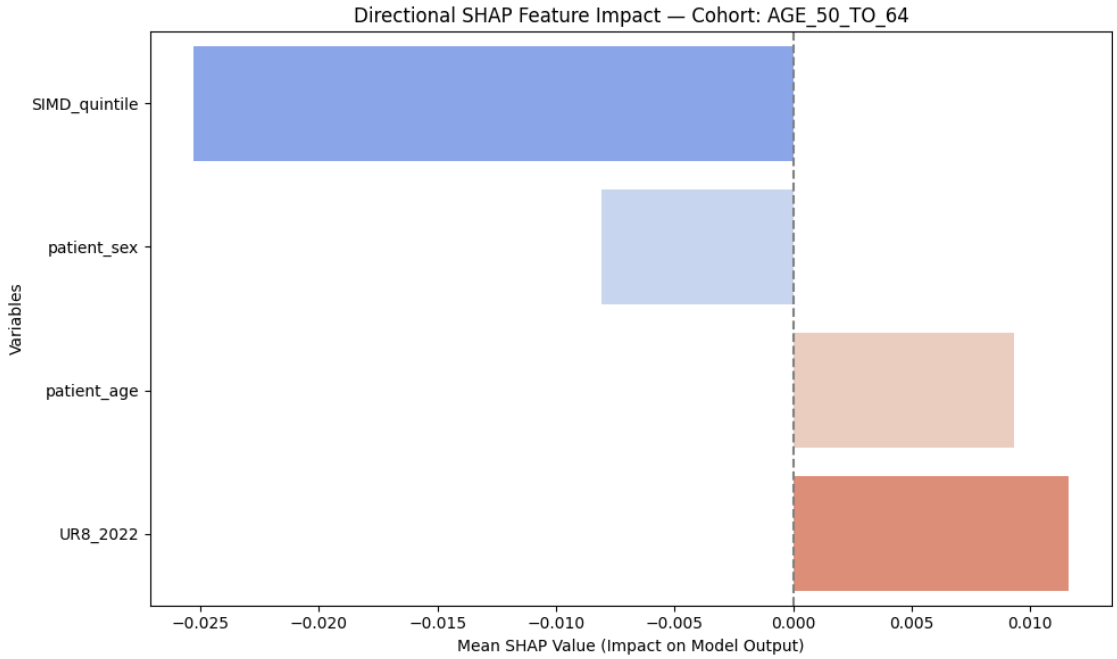
```
    # SHAP for selected cohorts

    cohort_size = len(X_test)
    if cohort_size < 100_000:
        shap_n = min(20000, cohort_size)
    elif cohort_size < 500_000:
        shap_n = 40000
    elif cohort_size < 1_000_000:
        shap_n = 80000
    else:
        shap_n = 160000

    print(f"\nSampling {shap_n} rows for SHAP from test set (size:␣
↪{cohort_size})")

    X_test_sample, _, y_test_sample, _ = train_test_split(
        X_test, y_test, train_size=shap_n, stratify=y_test, random_state=42
    )
    print(f"Actual SHAP sample size: {X_test_sample.shape[0]}")

    # SHAP Explainer for XGBoost
    explainer = shap.Explainer(xgb_model, X_train)
    shap_values = explainer(X_test_sample)

    # Directional SHAP bar plot
    print(f"\nDirectional SHAP Bar Plot - Cohort: {cohort}")
    shap_class1 = shap_values.values
    shap_df = pd.DataFrame(shap_class1, columns=X_test_sample.columns)
    shap_summary_directional = shap_df.mean().sort_values()

    plt.figure(figsize=(10, 6))
    sns.barplot(x=shap_summary_directional.values, y=shap_summary_directional.
↪index, palette="coolwarm")
    plt.axvline(0, color='gray', linestyle='--')
    plt.title(f"Directional SHAP Feature Impact - Cohort: {cohort}")
    plt.xlabel("Mean SHAP Value (Impact on Model Output)")
    plt.ylabel("Variables")
    plt.tight_layout()
    plt.show()

    # SHAP Dependence Plot for Top Features
    top_features = shap_summary_directional.index.tolist()

    for feature in top_features:
        print(f"SHAP Dependence Plot in Cohort {cohort} for: {feature}")
        shap.dependence_plot(feature, shap_values.values, X_test_sample,␣
↪show=True)
```

```
<IPython.core.display.HTML object>


======== Cohort: AGE_50_TO_64 =========
Train size: 2068840
Test size: 886646

/mnt/homes/vayly01/myenv/lib/python3.13/site-packages/xgboost/training.py:183:
UserWarning: [09:26:57] WARNING: /workspace/src/learner.cc:738:
Parameters: { "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)


Model Performance Metrics:
    Metric    Value
  Accuracy 0.592903
 Precision 0.583984
    Recall 0.536452
  F1 Score 0.559210
 AUC Score 0.628661
  Log Loss 0.666542
```

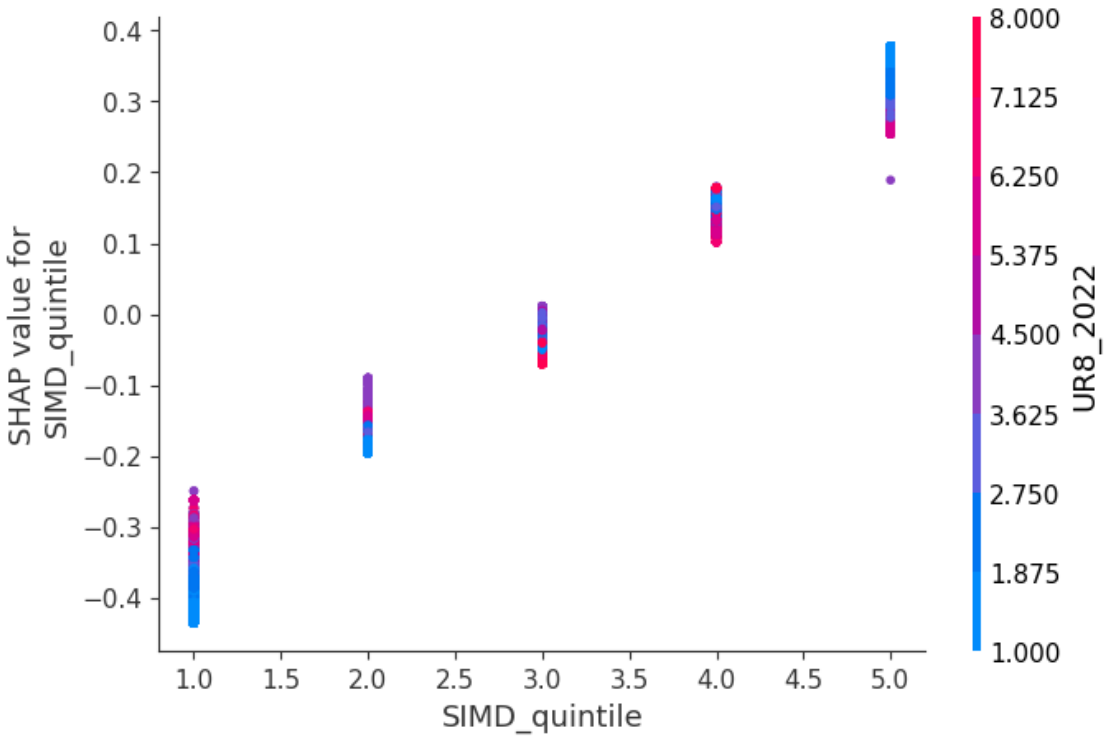



```
Top Feature Importances:
      Feature  Importance
  patient_age      0.4594
SIMD_quintile      0.3092
  patient_sex      0.1935
```

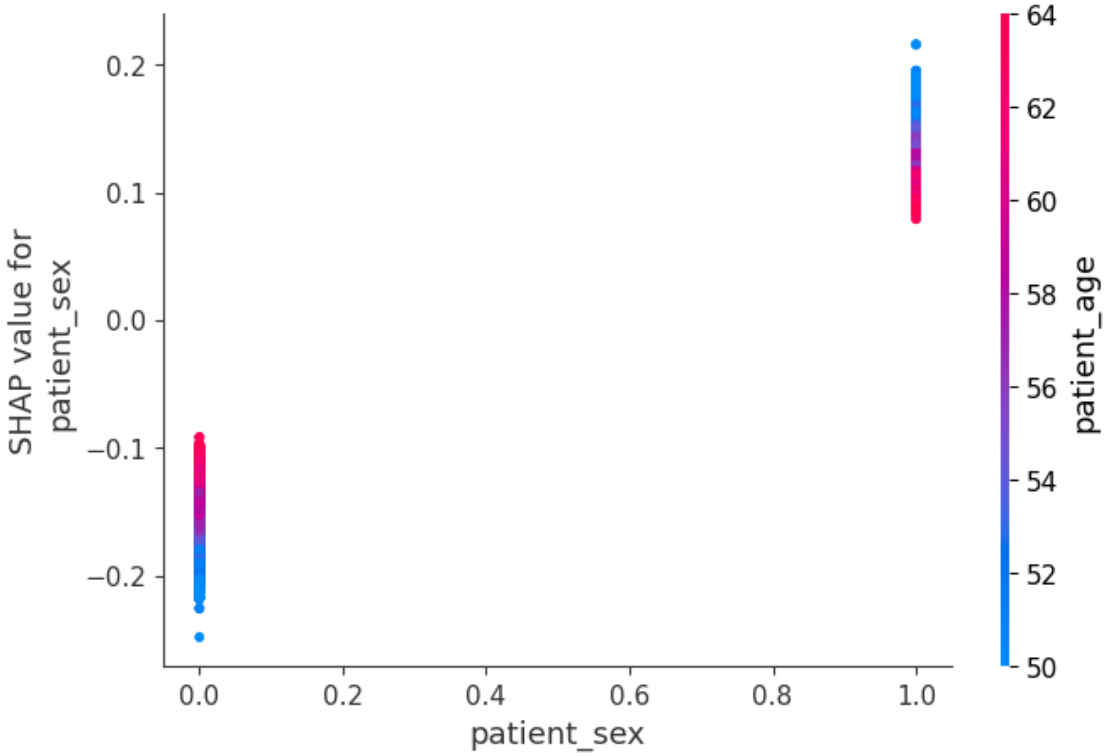
```
     UR8_2022      0.0379

Sampling 80000 rows for SHAP from test set (size: 886646)
Actual SHAP sample size: 80000

100%|==================| 79916/80000 [02:18<00:00]


Directional SHAP Bar Plot - Cohort: AGE_50_TO_64

/tmp/ipykernel_4511/78395491.py:135: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=shap_summary_directional.values,
y=shap_summary_directional.index, palette="coolwarm")
```



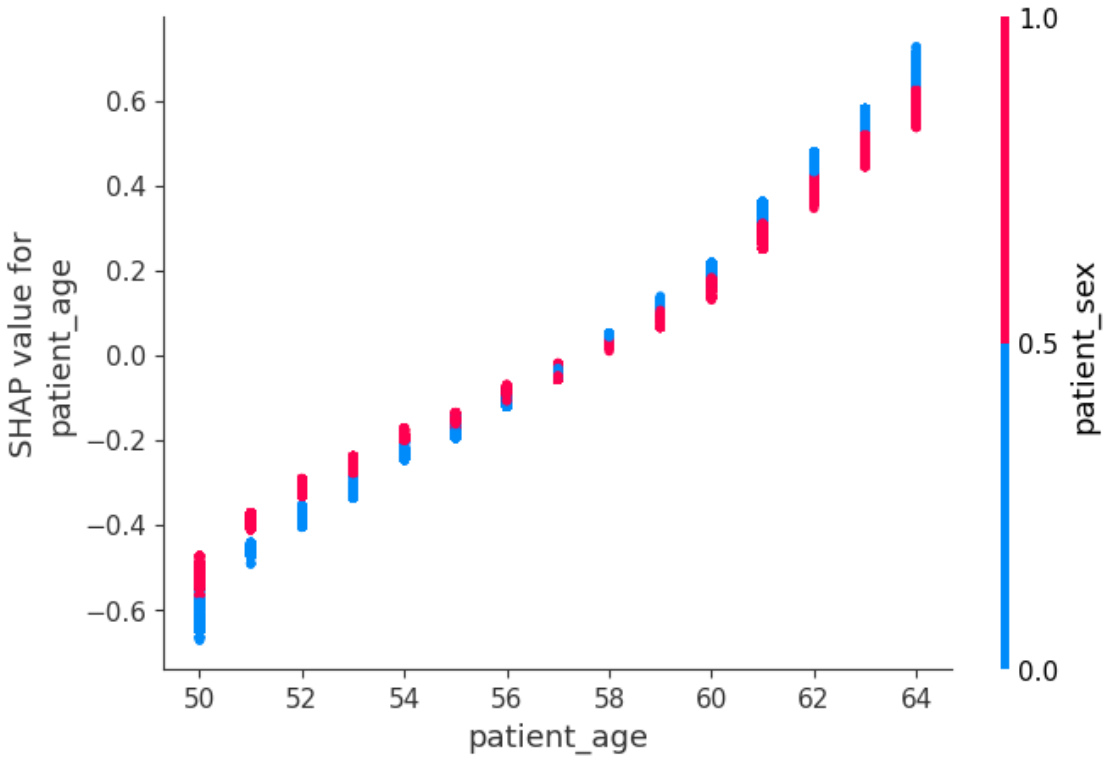

```
SHAP Dependence Plot in Cohort AGE_50_TO_64 for: SIMD_quintile
```



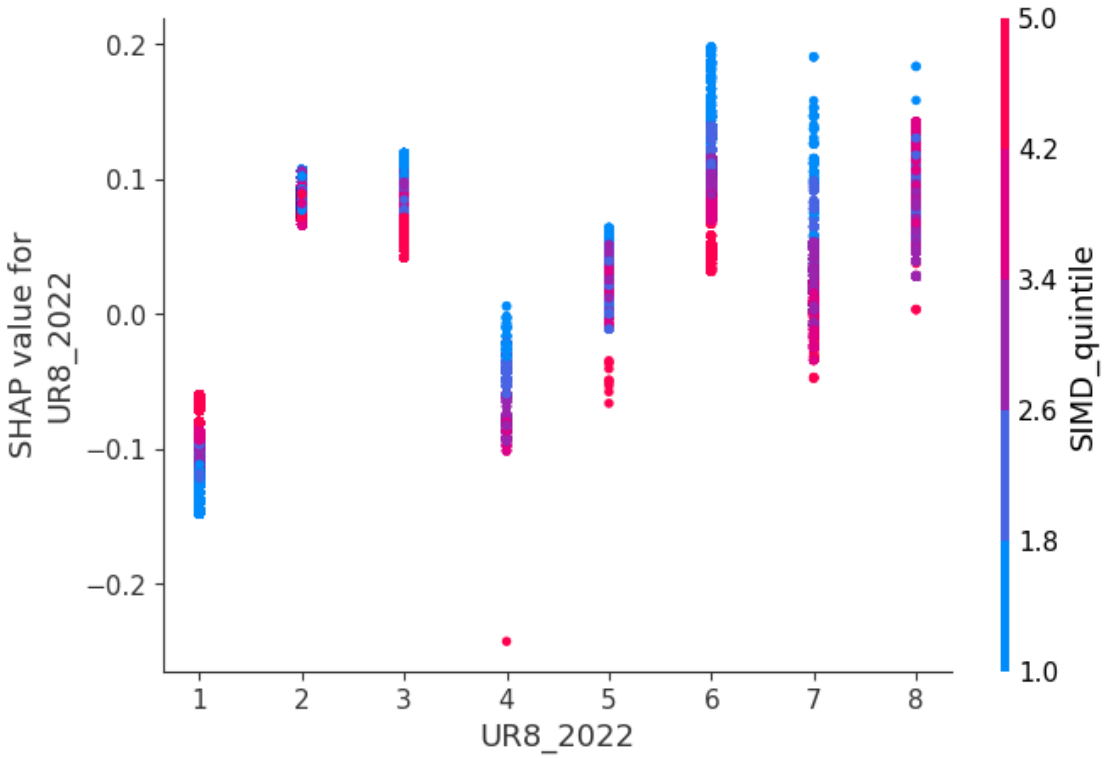
SHAP Dependence Plot in Cohort AGE_50_TO_64 for: patient_sex



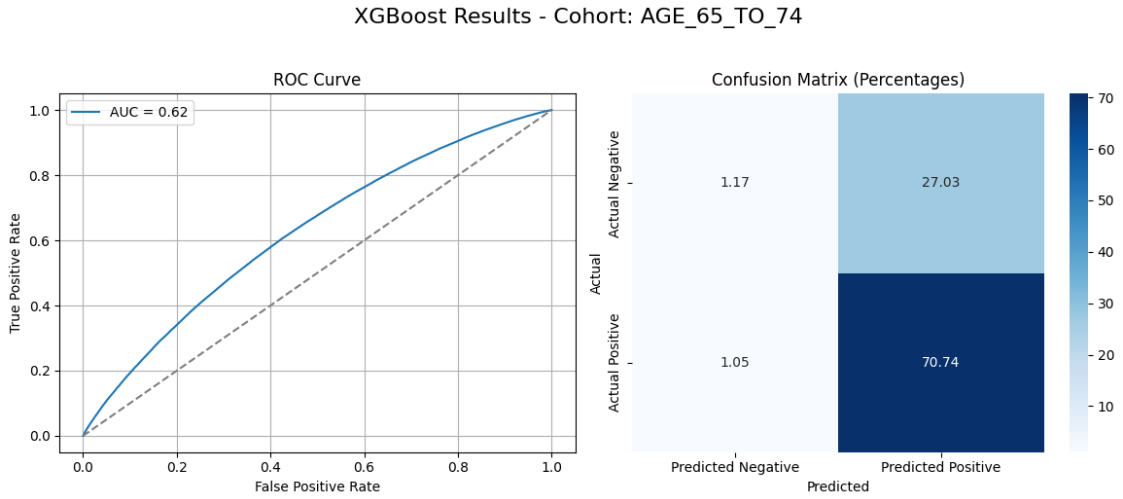
SHAP Dependence Plot in Cohort AGE_50_TO_64 for: patient_age



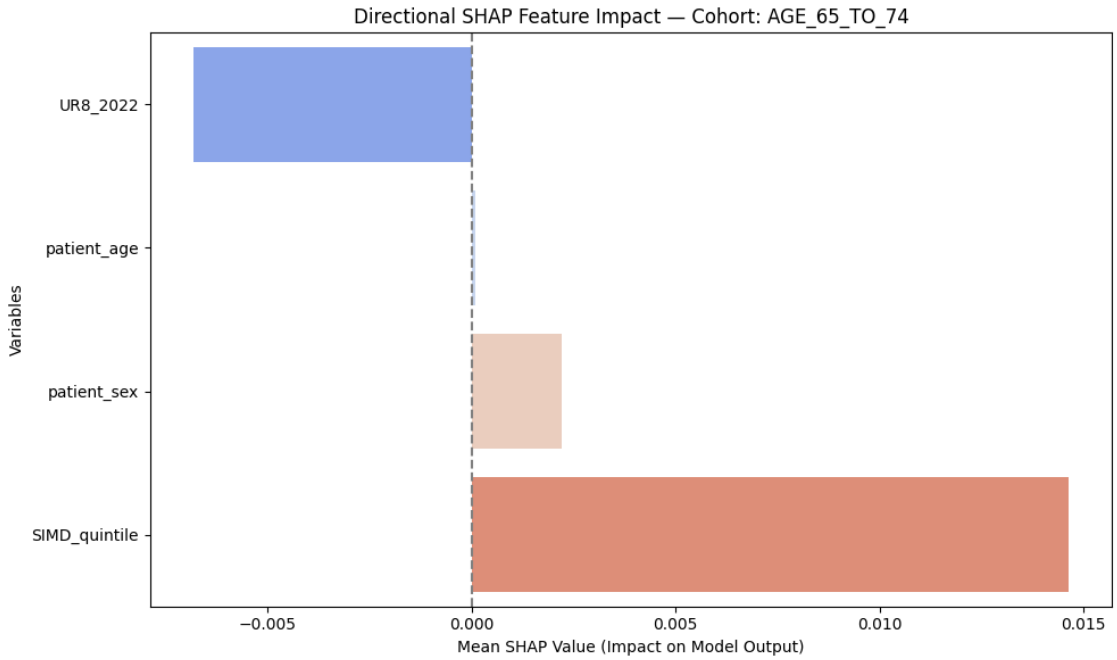
SHAP Dependence Plot in Cohort AGE_50_TO_64 for: UR8_2022



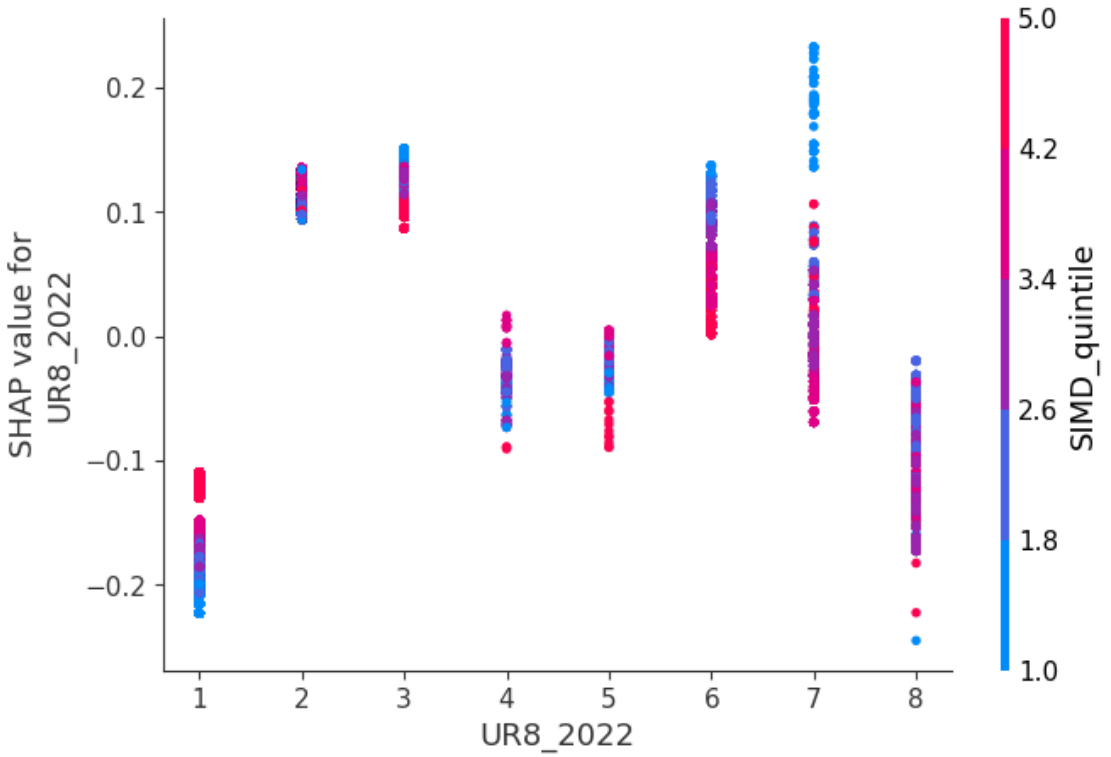

```
======== Cohort: AGE_65_TO_74 =========
Train size: 896325
Test size: 384140

/mnt/homes/vayly01/myenv/lib/python3.13/site-packages/xgboost/training.py:183:
UserWarning: [09:29:27] WARNING: /workspace/src/learner.cc:738:
Parameters: { "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)


Model Performance Metrics:
    Metric    Value
  Accuracy 0.719173
 Precision 0.723513
    Recall 0.985398
  F1 Score 0.834389
 AUC Score 0.624382
  Log Loss 0.575073
```



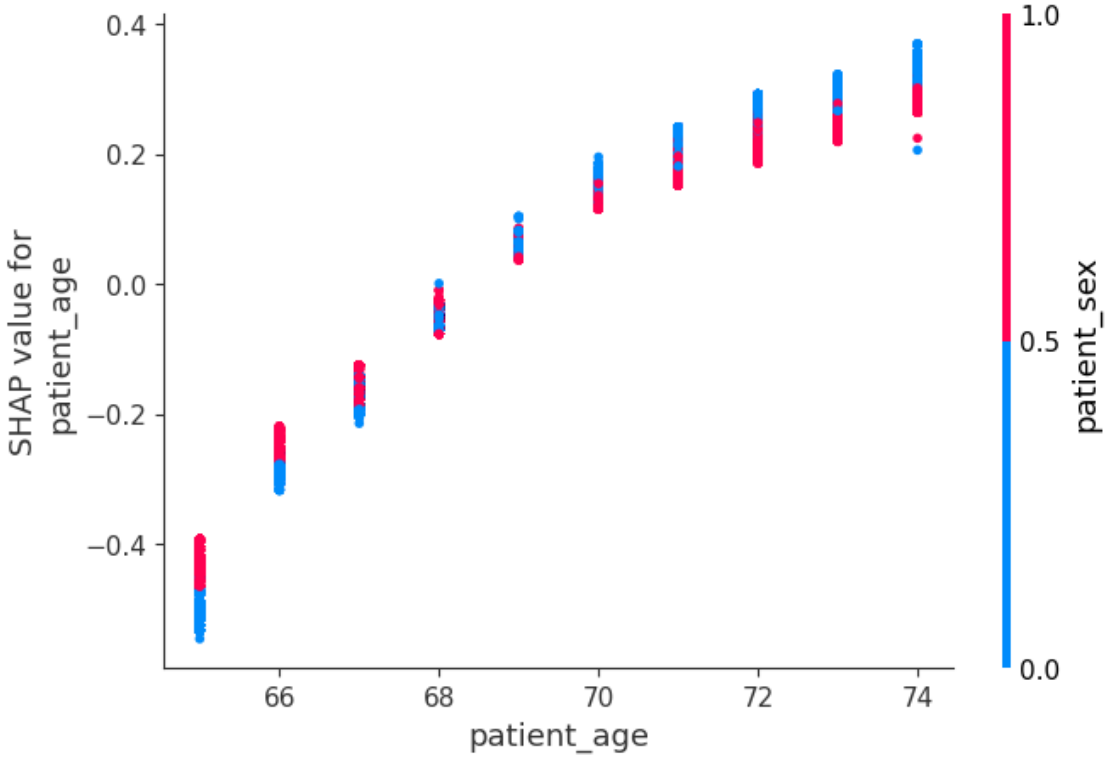

```
Top Feature Importances:
      Feature  Importance
SIMD_quintile      0.5668
  patient_age      0.3398
     UR8_2022      0.0751
  patient_sex      0.0182

Sampling 40000 rows for SHAP from test set (size: 384140)
Actual SHAP sample size: 40000
100%|====================| 39900/40000 [01:04<00:00]


Directional SHAP Bar Plot - Cohort: AGE_65_TO_74
/tmp/ipykernel_4511/78395491.py:135: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=shap_summary_directional.values,
y=shap_summary_directional.index, palette="coolwarm")
```



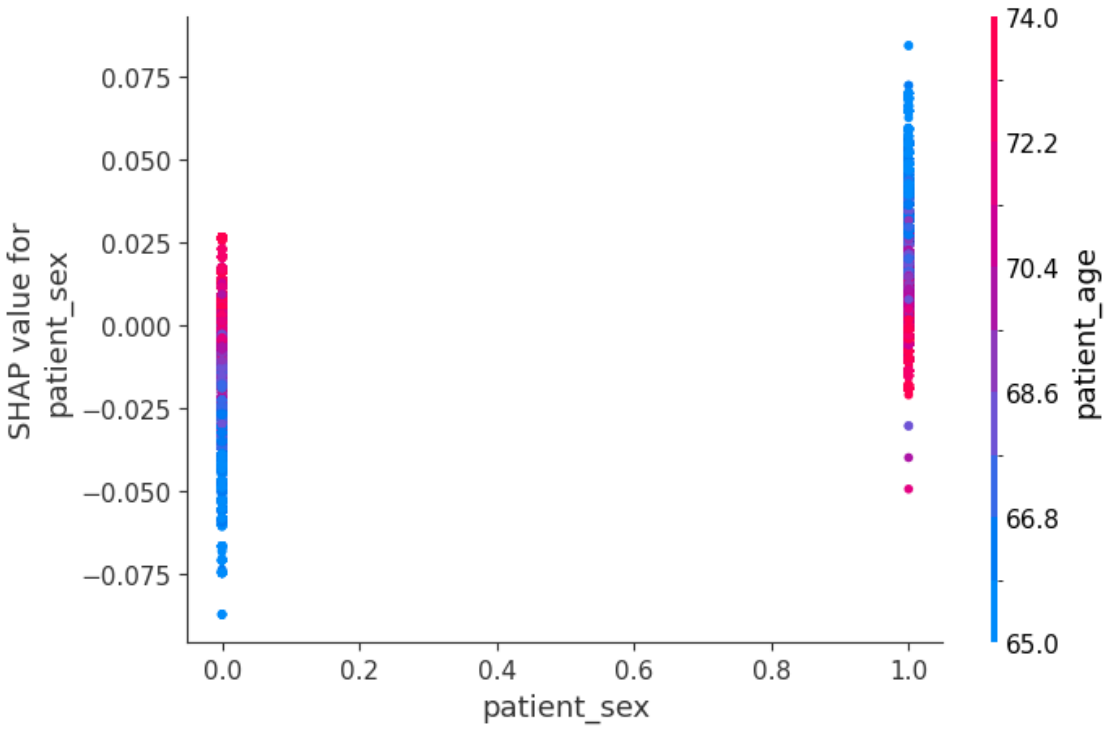
SHAP Dependence Plot in Cohort AGE_65_TO_74 for: UR8_2022



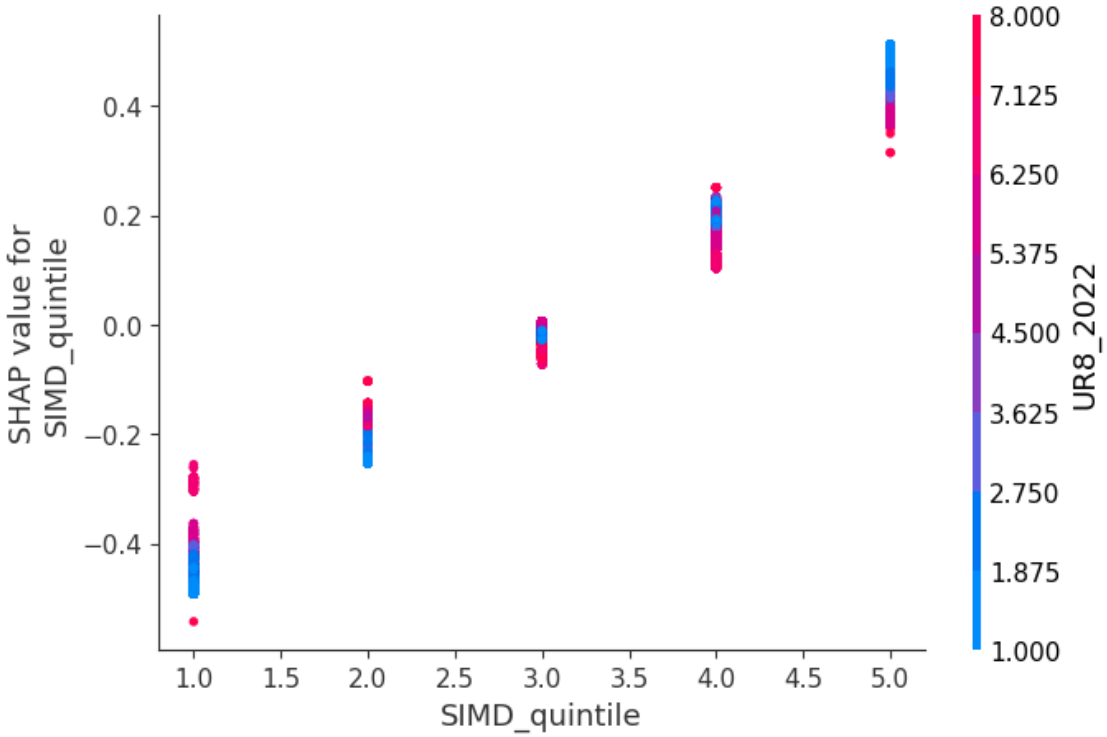
SHAP Dependence Plot in Cohort AGE_65_TO_74 for: patient_age



SHAP Dependence Plot in Cohort AGE_65_TO_74 for: patient_sex



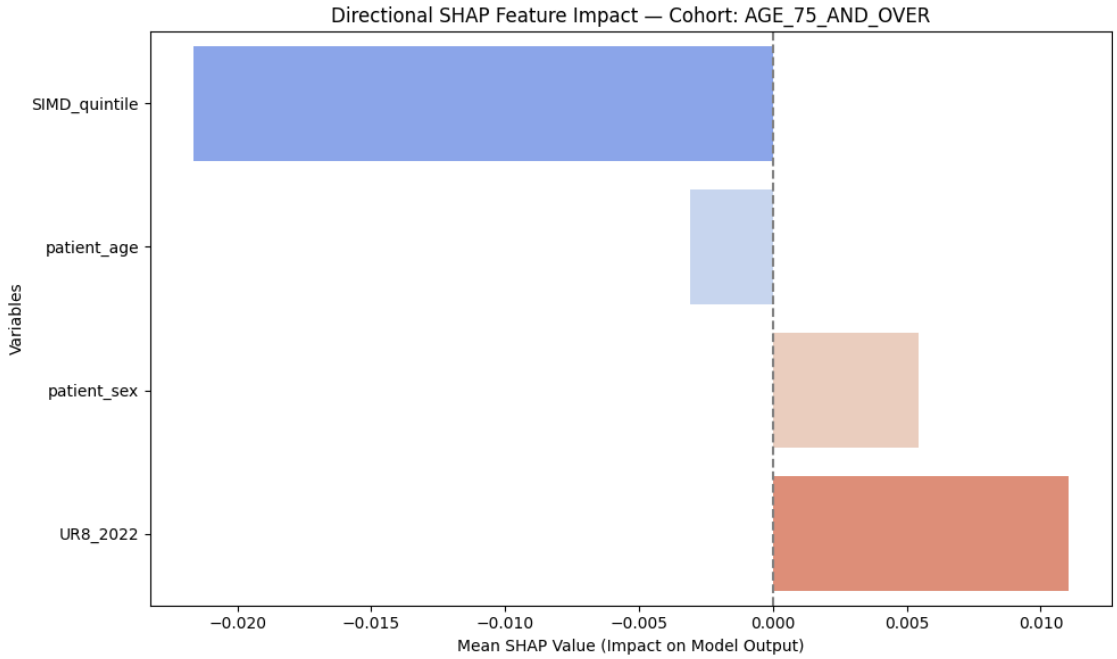
SHAP Dependence Plot in Cohort AGE_65_TO_74 for: SIMD_quintile



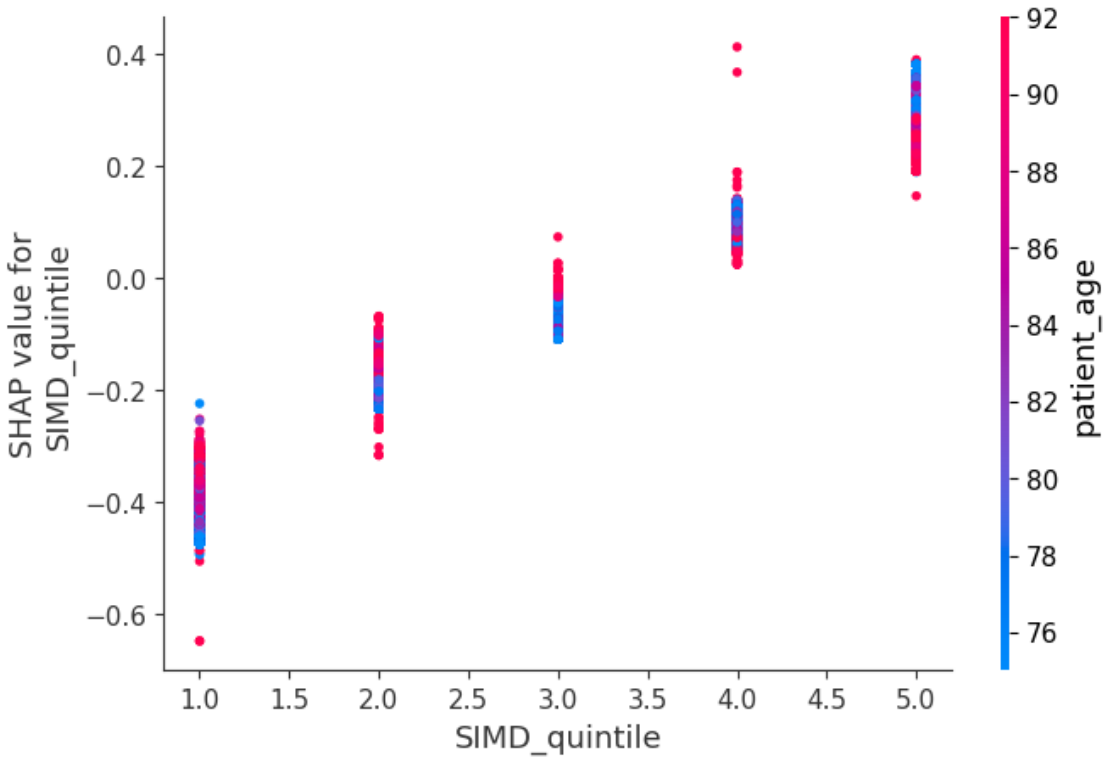
```
======== Cohort: AGE_75_AND_OVER =========
Train size: 794668
Test size: 340572

/mnt/homes/vayly01/myenv/lib/python3.13/site-packages/xgboost/training.py:183:
UserWarning: [09:30:38] WARNING: /workspace/src/learner.cc:738:
Parameters: { "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)


Model Performance Metrics:
   Metric    Value
 Accuracy 0.809098
Precision 0.809115
   Recall 0.999960
 F1 Score 0.894471
AUC Score 0.580348
 Log Loss 0.481286
```


```
Top Feature Importances:
      Feature  Importance
SIMD_quintile      0.8000
     UR8_2022      0.1000
  patient_sex      0.0577
  patient_age      0.0423
```

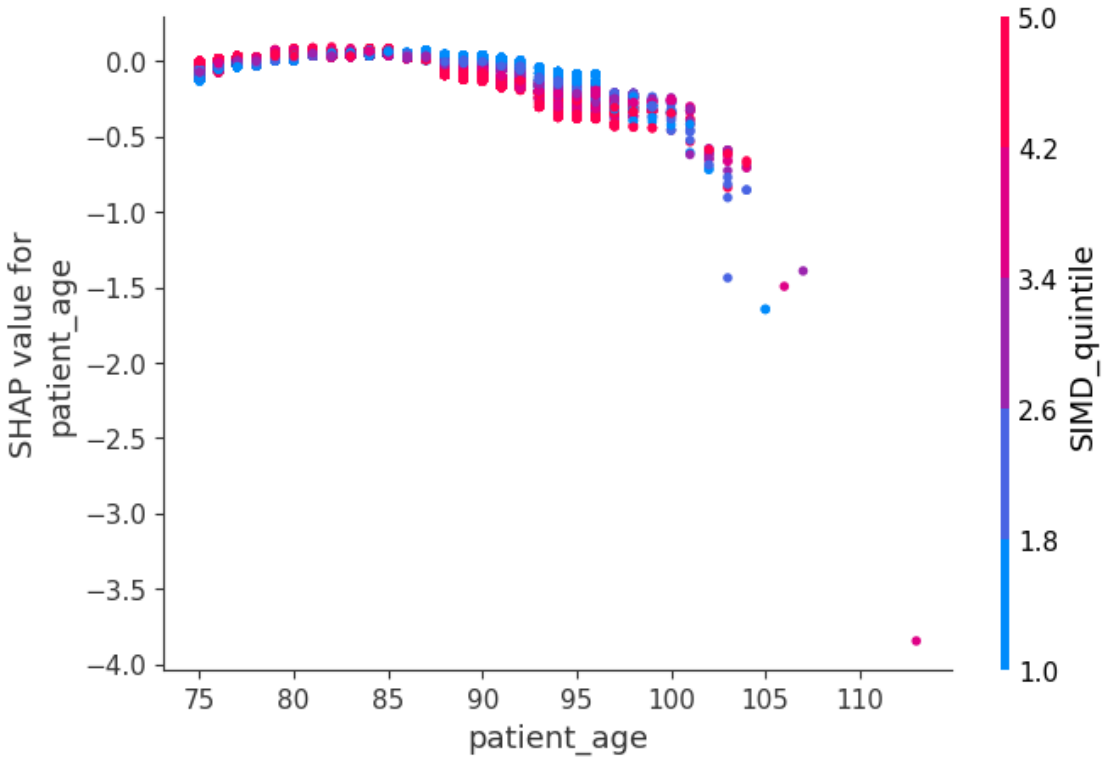
```
Sampling 40000 rows for SHAP from test set (size: 340572)
Actual SHAP sample size: 40000
 99%|===================| 39431/40000 [00:48<00:00]

Directional SHAP Bar Plot - Cohort: AGE_75_AND_OVER
/tmp/ipykernel_4511/78395491.py:135: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=shap_summary_directional.values,
y=shap_summary_directional.index, palette="coolwarm")
```



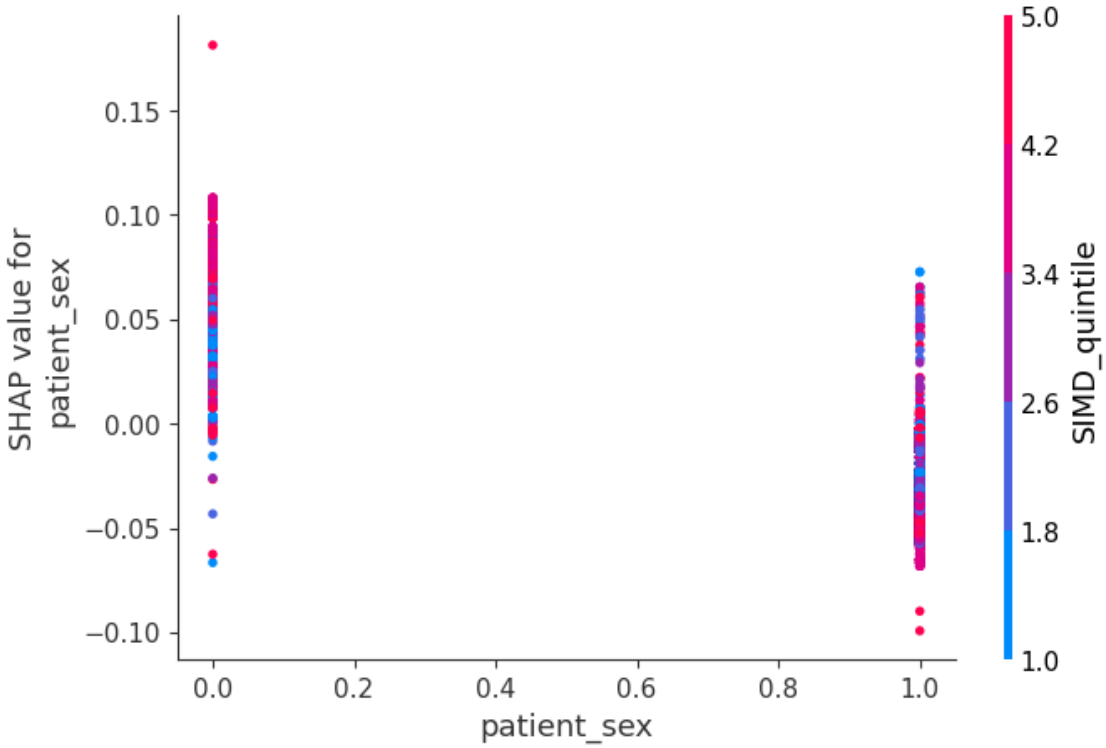

```
SHAP Dependence Plot in Cohort AGE_75_AND_OVER for: SIMD_quintile
```



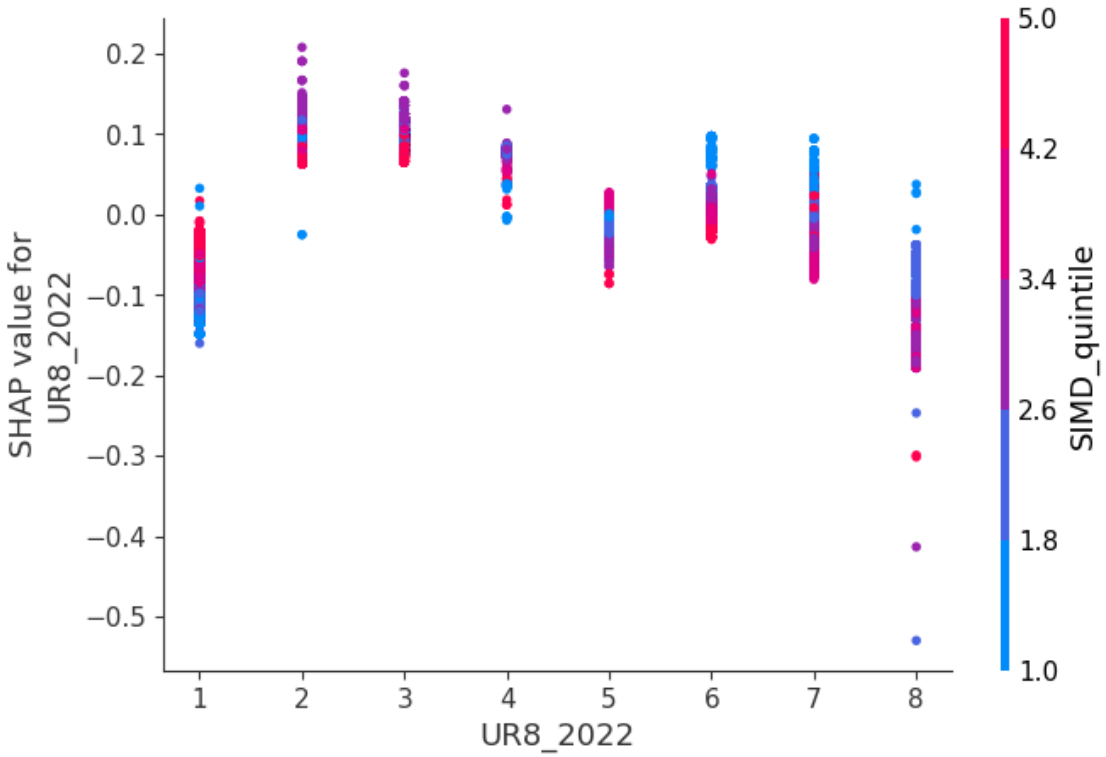
SHAP Dependence Plot in Cohort AGE_75_AND_OVER for: patient_age



SHAP Dependence Plot in Cohort AGE_75_AND_OVER for: patient_sex

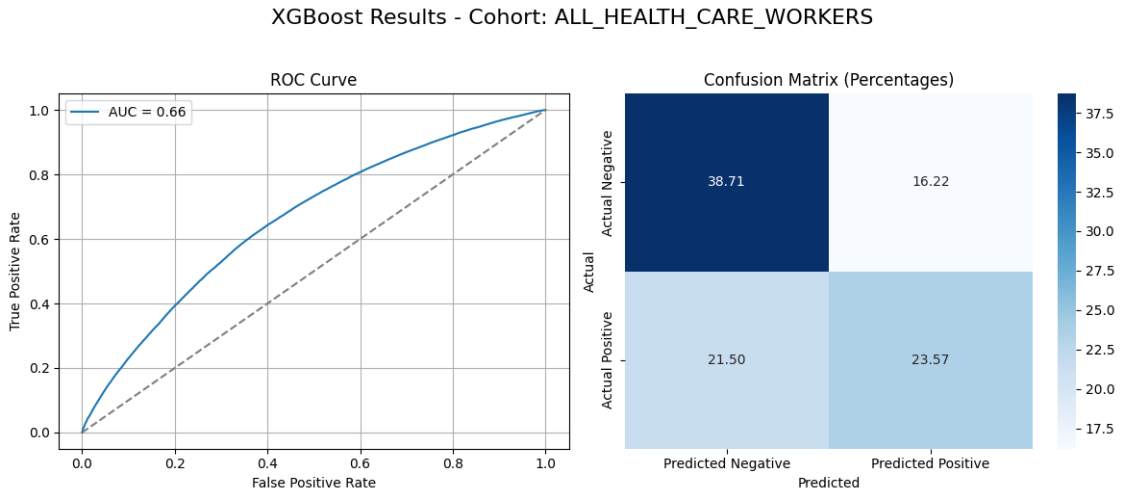


SHAP Dependence Plot in Cohort AGE_75_AND_OVER for: UR8_2022

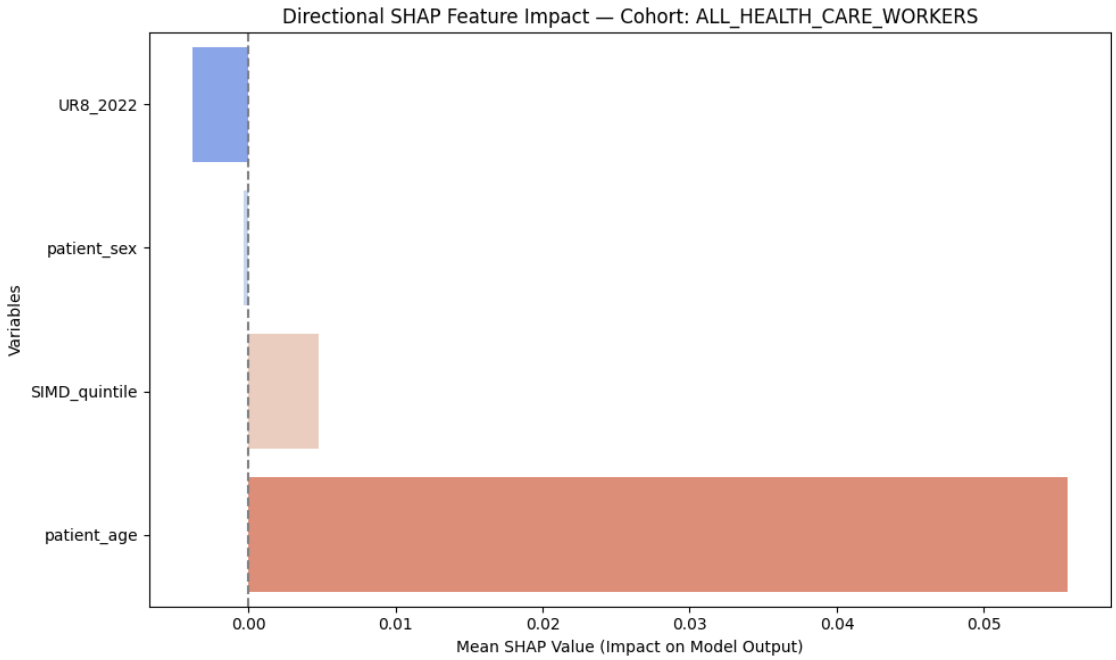



```
======== Cohort: ALL_HEALTH_CARE_WORKERS =========
Train size: 317592
Test size: 136111

/mnt/homes/vayly01/myenv/lib/python3.13/site-packages/xgboost/training.py:183:
UserWarning: [09:31:32] WARNING: /workspace/src/learner.cc:738:
Parameters: { "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)


Model Performance Metrics:
    Metric    Value
  Accuracy 0.622830
 Precision 0.592397
    Recall 0.523033
  F1 Score 0.555558
 AUC Score 0.661593
```

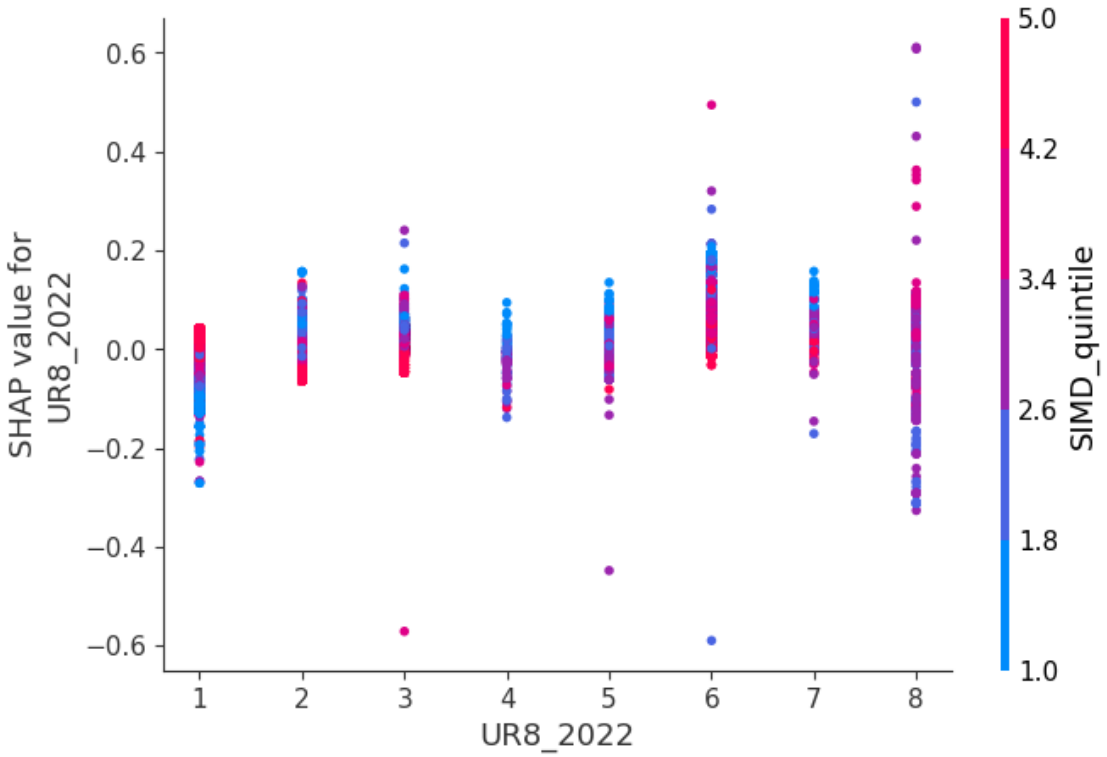
```
Log Loss 0.647826
```



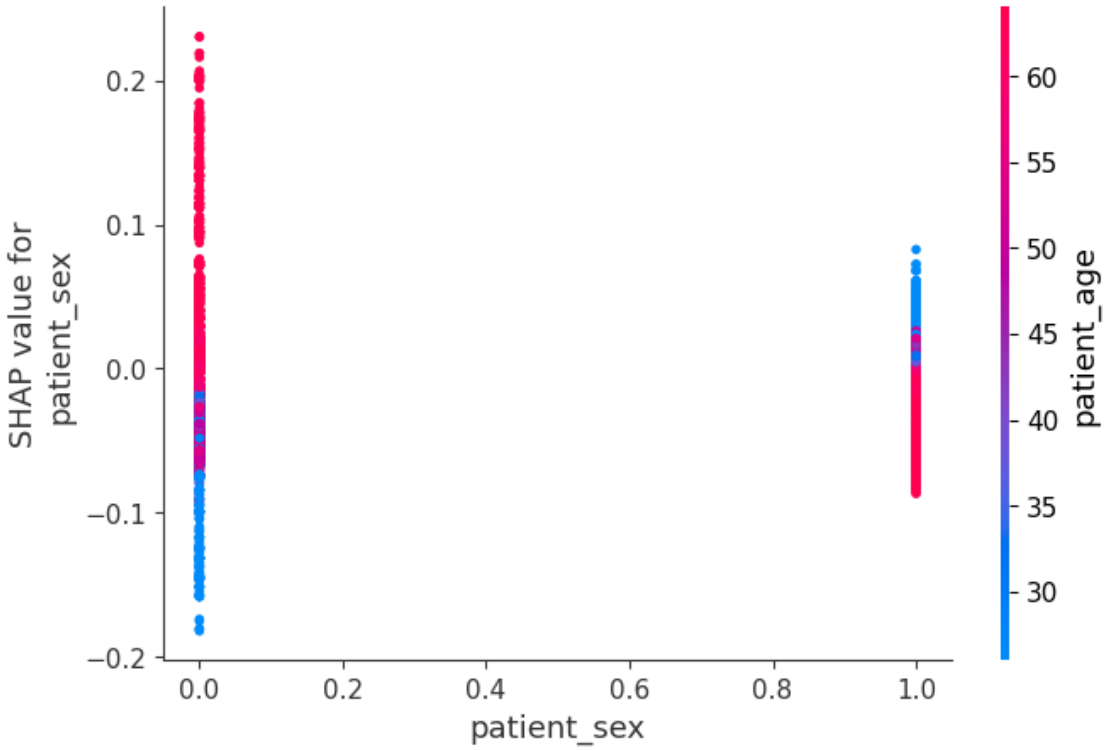

```
Top Feature Importances:
      Feature  Importance
  patient_age      0.7012
SIMD_quintile      0.2305
  patient_sex      0.0344
     UR8_2022      0.0339

Sampling 40000 rows for SHAP from test set (size: 136111)
Actual SHAP sample size: 40000

100%|==================| 39817/40000 [00:51<00:00]


Directional SHAP Bar Plot - Cohort: ALL_HEALTH_CARE_WORKERS

/tmp/ipykernel_4511/78395491.py:135: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=shap_summary_directional.values,
y=shap_summary_directional.index, palette="coolwarm")
```



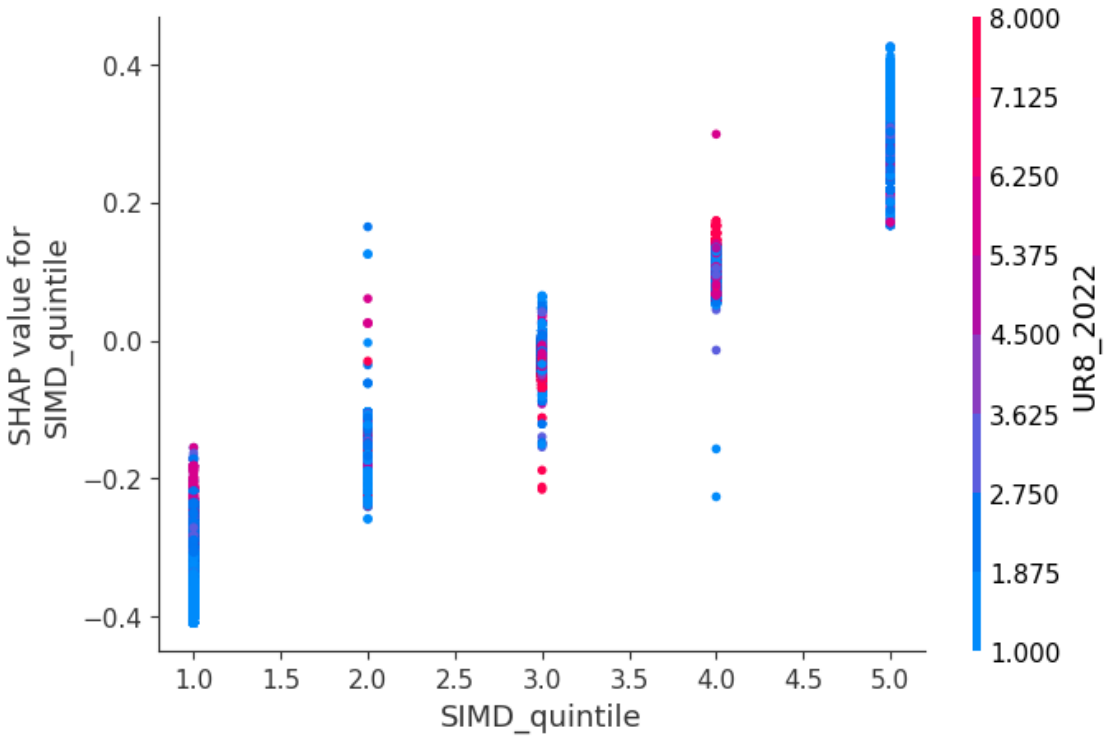
SHAP Dependence Plot in Cohort ALL_HEALTH_CARE_WORKERS for: UR8_2022



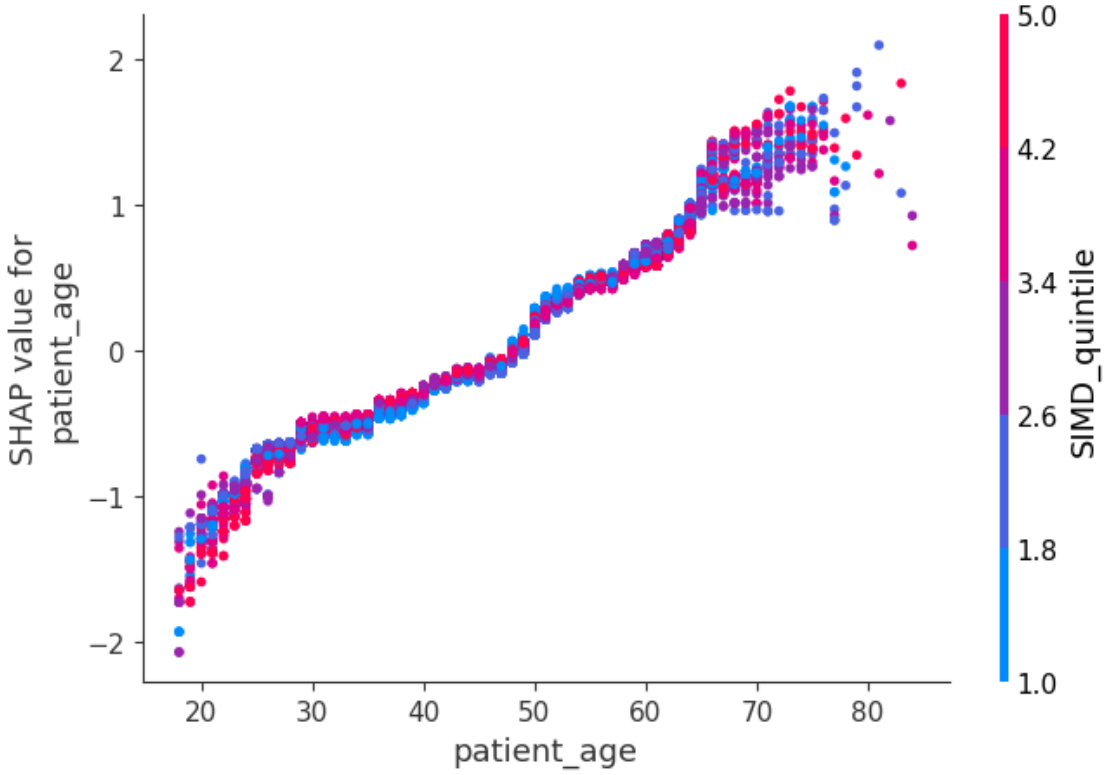
SHAP Dependence Plot in Cohort ALL_HEALTH_CARE_WORKERS for: patient_sex



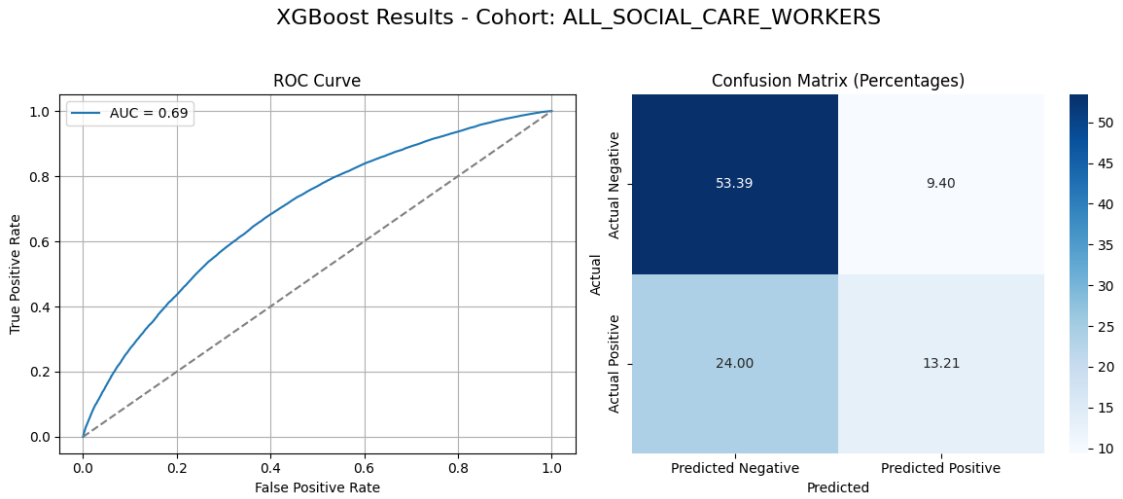
SHAP Dependence Plot in Cohort ALL_HEALTH_CARE_WORKERS for: SIMD_quintile



SHAP Dependence Plot in Cohort ALL_HEALTH_CARE_WORKERS for: patient_age

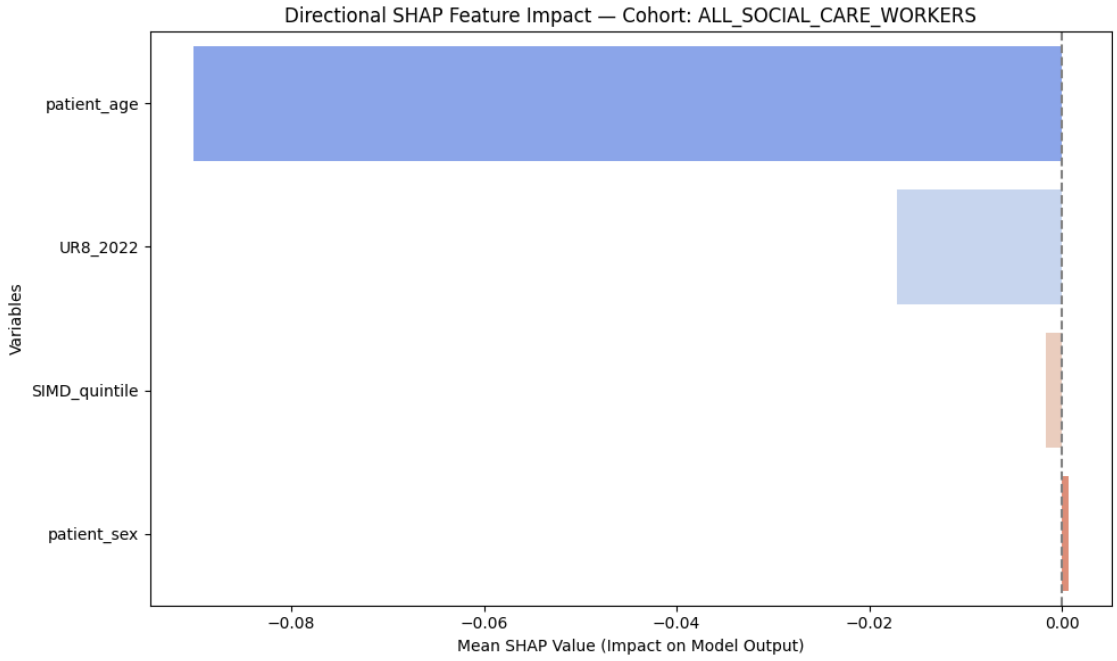


```
======== Cohort: ALL_SOCIAL_CARE_WORKERS =========
Train size: 201854
Test size: 86509

/mnt/homes/vayly01/myenv/lib/python3.13/site-packages/xgboost/training.py:183:
UserWarning: [09:32:26] WARNING: /workspace/src/learner.cc:738:
Parameters: { "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)


Model Performance Metrics:
    Metric    Value
  Accuracy 0.666023
 Precision 0.584301
    Recall 0.354966
  F1 Score 0.441636
 AUC Score 0.690873
  Log Loss 0.606264
```

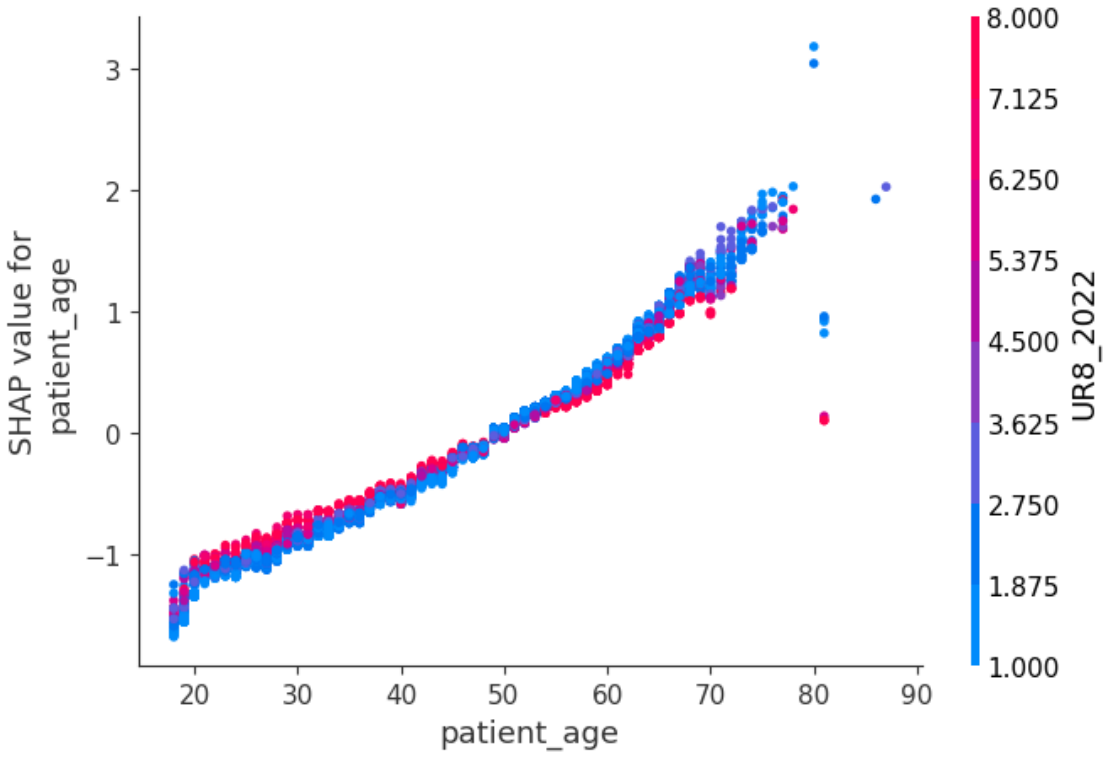



```
Top Feature Importances:
      Feature  Importance
  patient_age      0.8202
SIMD_quintile      0.1036
     UR8_2022      0.0525
  patient_sex      0.0237
```

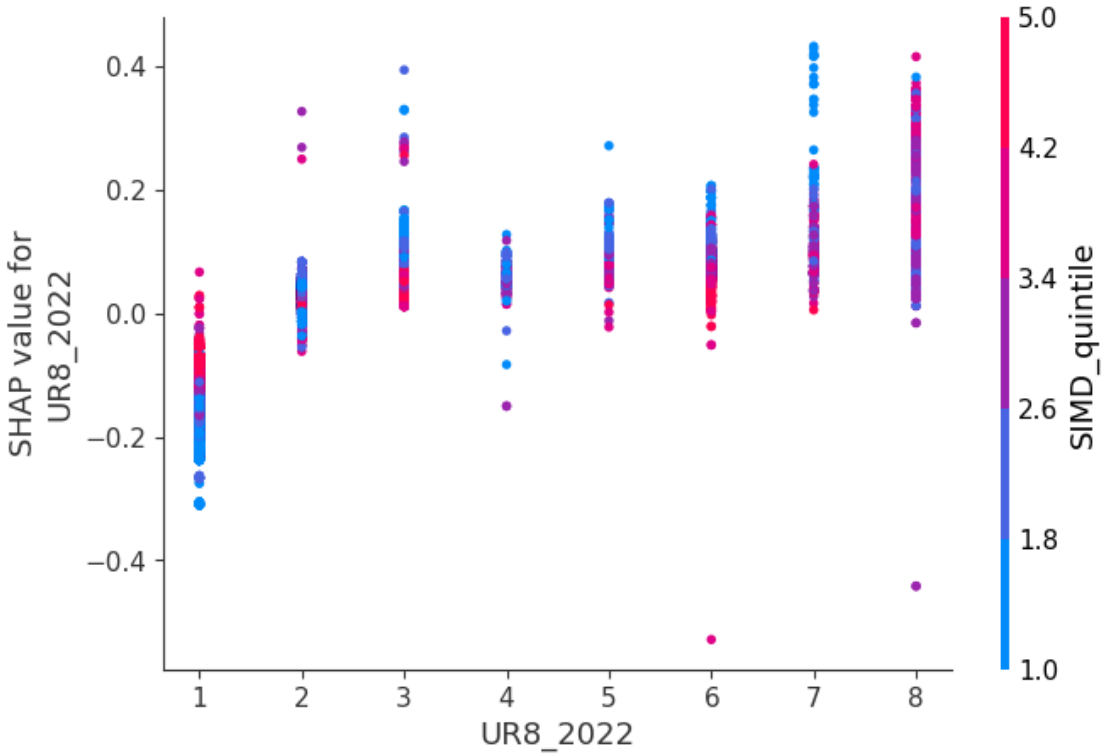
```
Sampling 20000 rows for SHAP from test set (size: 86509)
Actual SHAP sample size: 20000
 97%|=================== | 19466/20000 [00:24<00:00]

Directional SHAP Bar Plot - Cohort: ALL_SOCIAL_CARE_WORKERS
/tmp/ipykernel_4511/78395491.py:135: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=shap_summary_directional.values,
y=shap_summary_directional.index, palette="coolwarm")
```



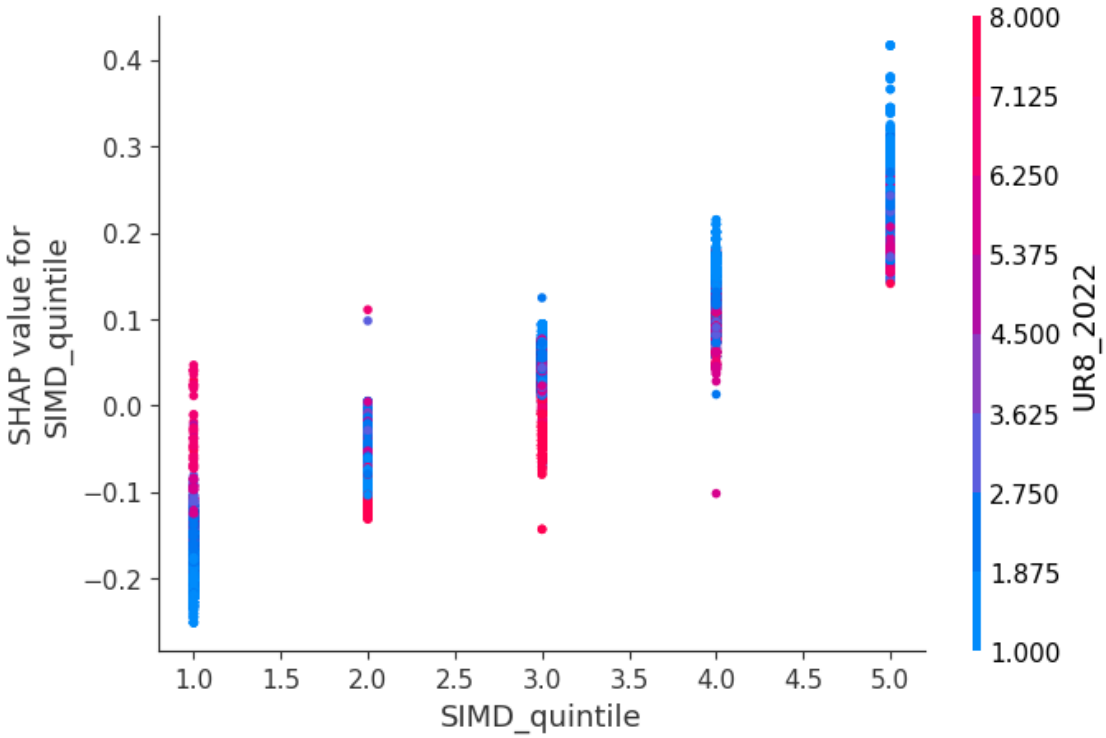

```
SHAP Dependence Plot in Cohort ALL_SOCIAL_CARE_WORKERS for: patient_age
```



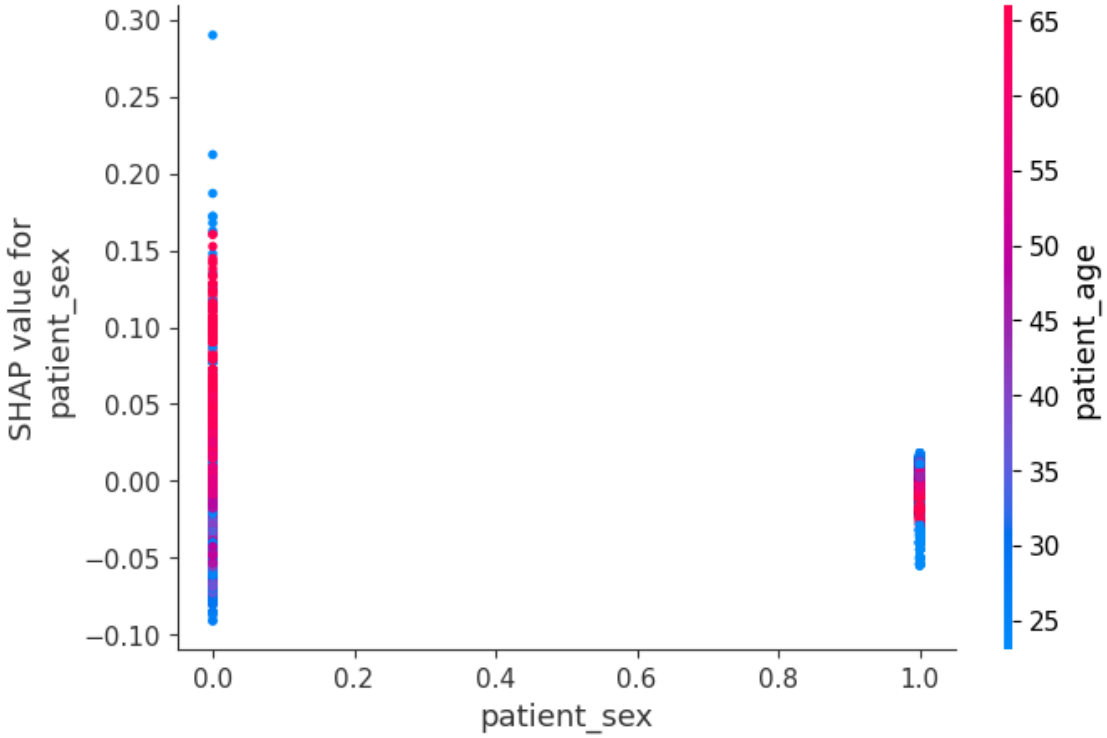
SHAP Dependence Plot in Cohort ALL_SOCIAL_CARE_WORKERS for: UR8_2022



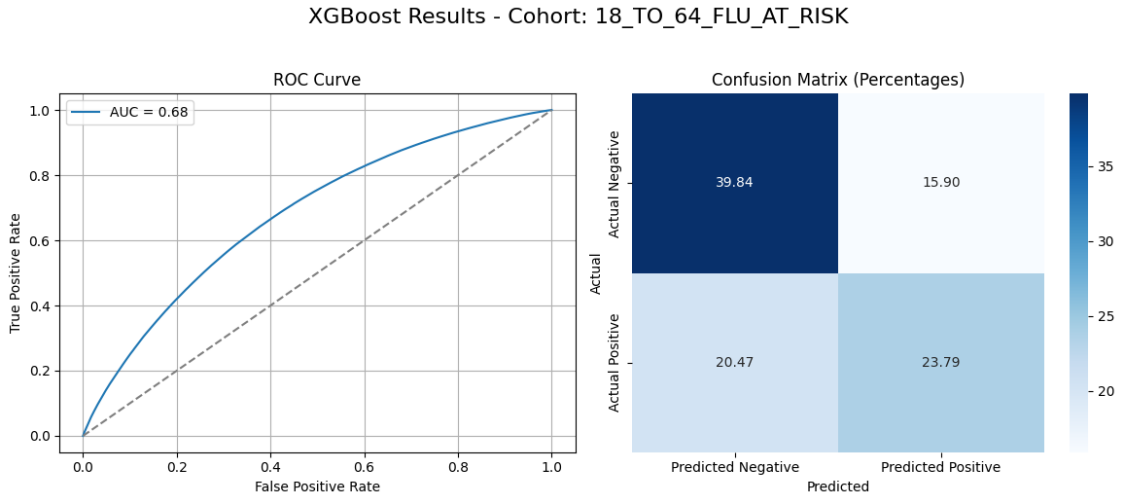
SHAP Dependence Plot in Cohort ALL_SOCIAL_CARE_WORKERS for: SIMD_quintile



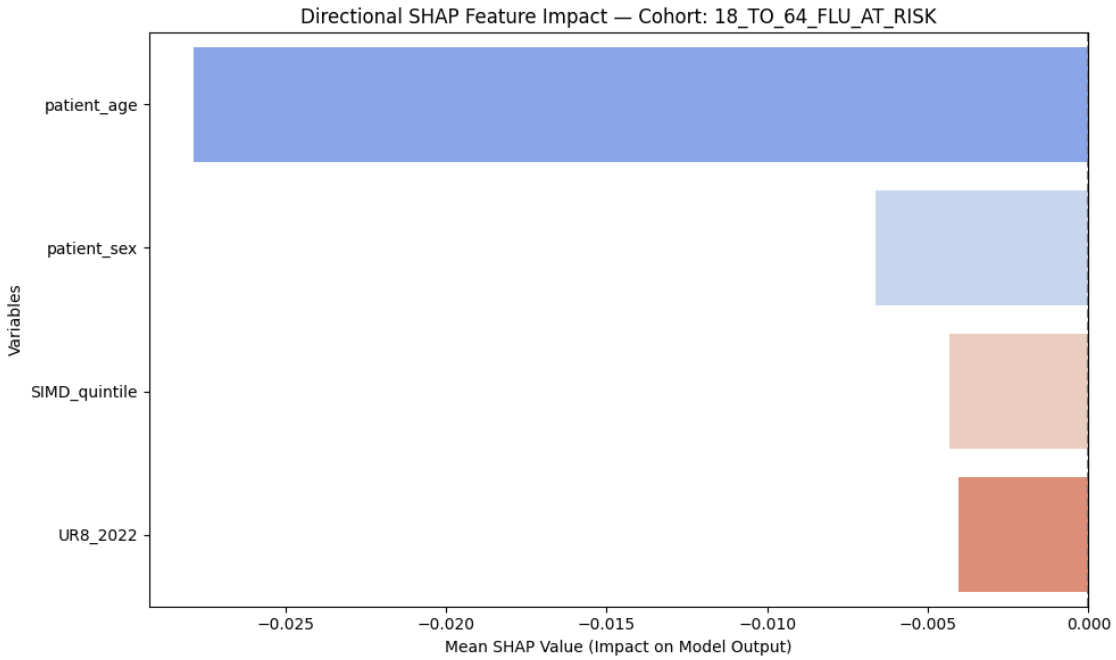
SHAP Dependence Plot in Cohort ALL_SOCIAL_CARE_WORKERS for: patient_sex



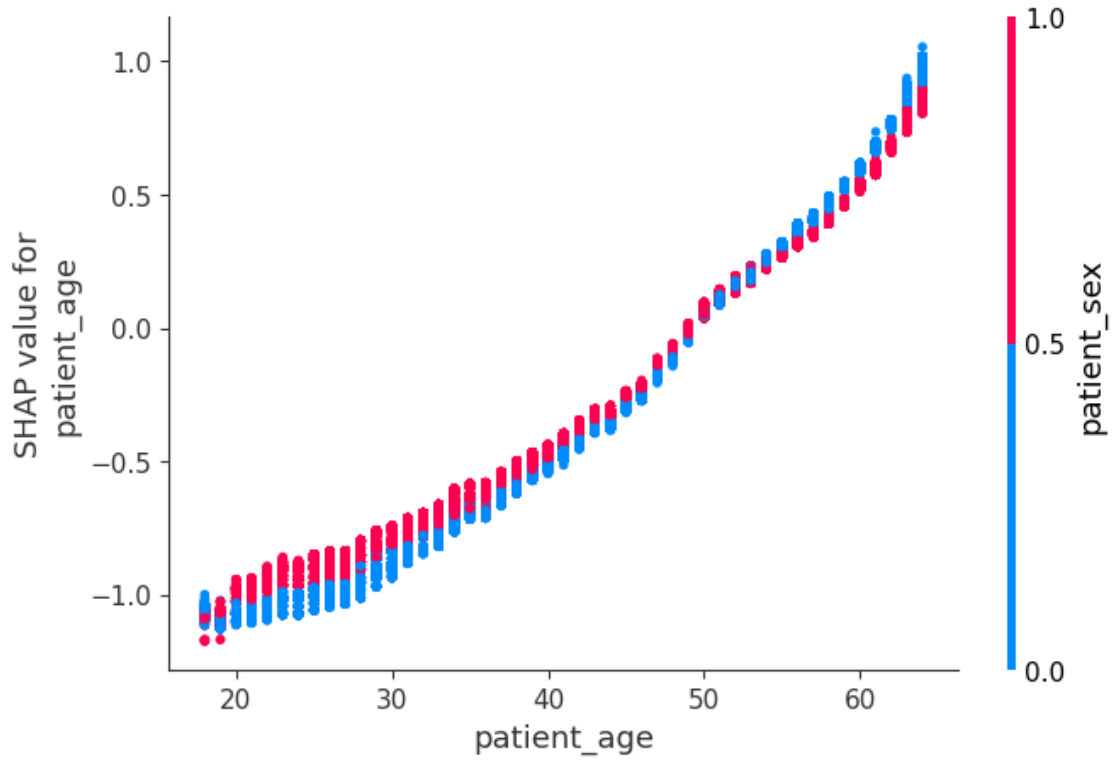

```
======== Cohort: 18_TO_64_FLU_AT_RISK =========
Train size: 1849322
Test size: 792567

/mnt/homes/vayly01/myenv/lib/python3.13/site-packages/xgboost/training.py:183:
UserWarning: [09:32:55] WARNING: /workspace/src/learner.cc:738:
Parameters: { "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)


Model Performance Metrics:
    Metric    Value
  Accuracy 0.636229
 Precision 0.599295
    Recall 0.537423
  F1 Score 0.566675
 AUC Score 0.679473
  Log Loss 0.636801
```



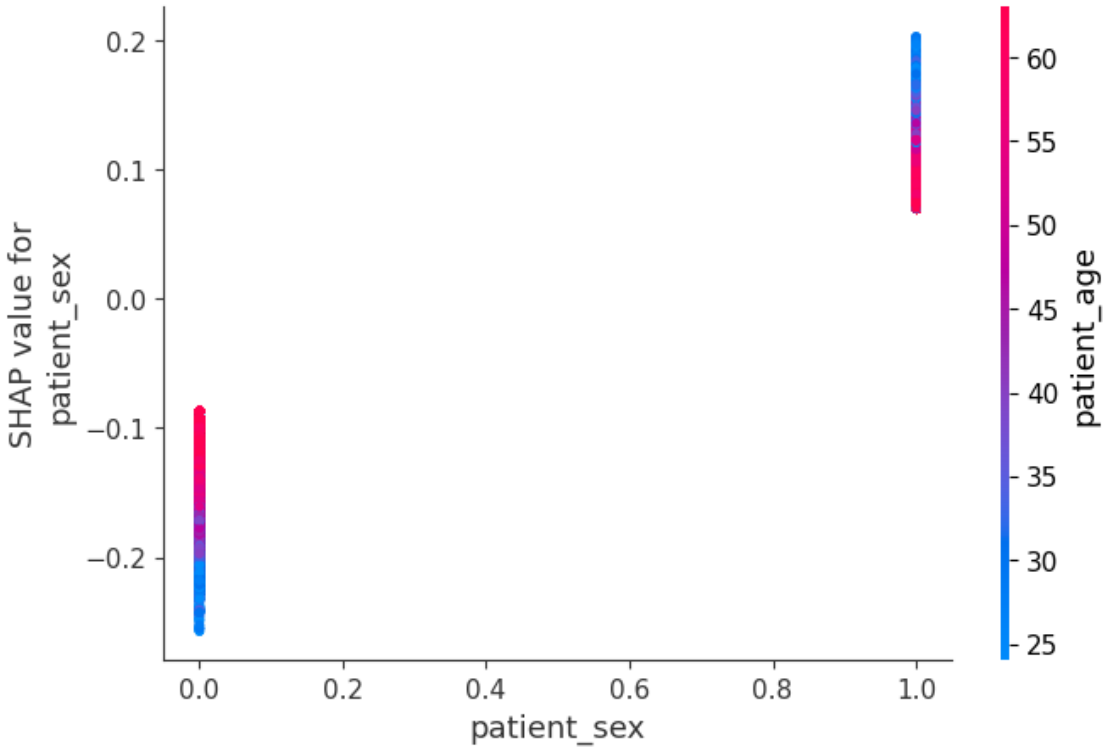

```
Top Feature Importances:
      Feature  Importance
  patient_age      0.6719
SIMD_quintile      0.2043
  patient_sex      0.1003
     UR8_2022      0.0234

Sampling 80000 rows for SHAP from test set (size: 792567)
Actual SHAP sample size: 80000
 99%|===================| 79398/80000 [02:07<00:00]


Directional SHAP Bar Plot - Cohort: 18_TO_64_FLU_AT_RISK
/tmp/ipykernel_4511/78395491.py:135: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=shap_summary_directional.values,
y=shap_summary_directional.index, palette="coolwarm")
```



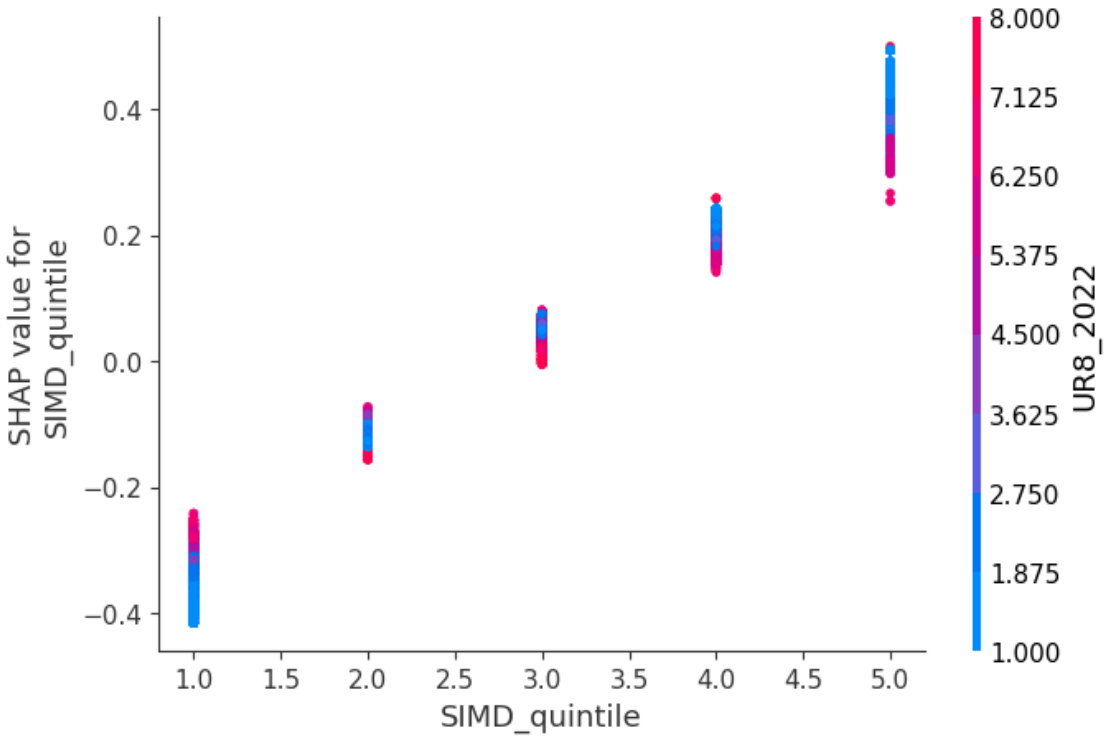
SHAP Dependence Plot in Cohort 18_TO_64_FLU_AT_RISK for: patient_age



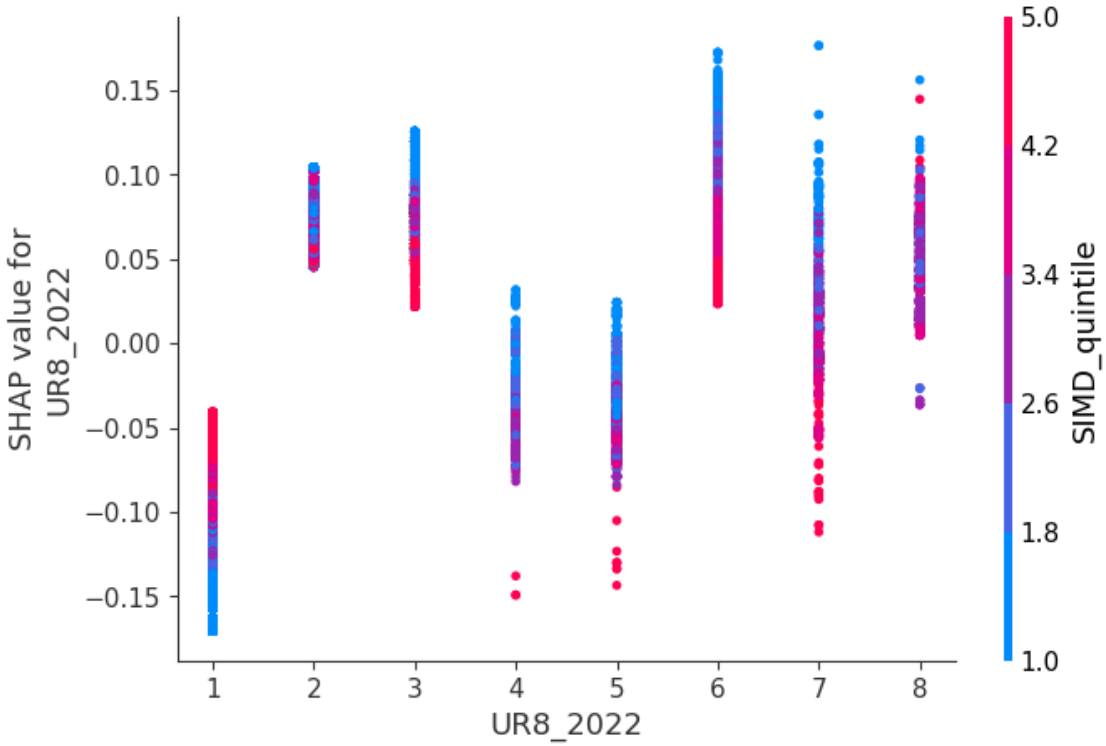
SHAP Dependence Plot in Cohort 18_TO_64_FLU_AT_RISK for: patient_sex



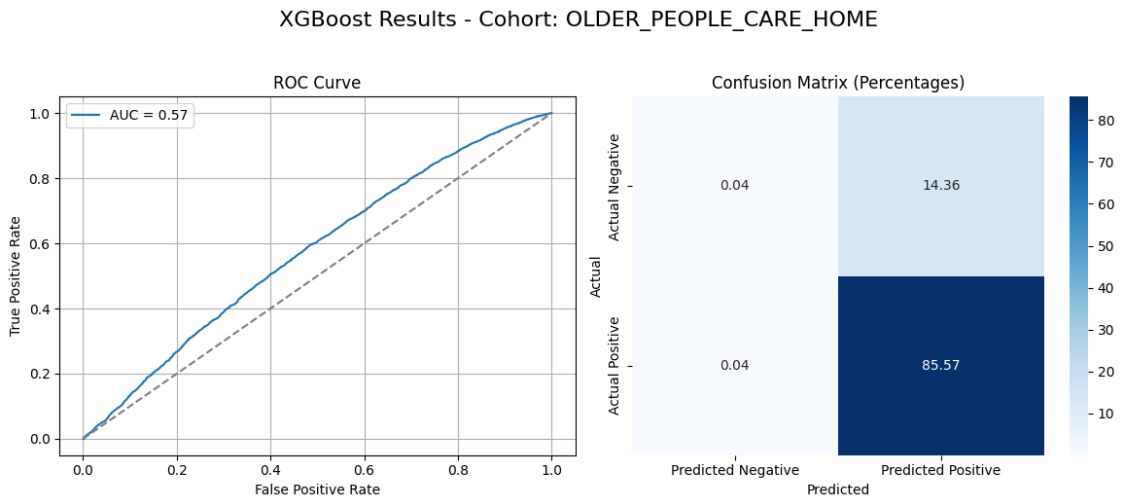
SHAP Dependence Plot in Cohort 18_TO_64_FLU_AT_RISK for: SIMD_quintile



SHAP Dependence Plot in Cohort 18_TO_64_FLU_AT_RISK for: UR8_2022

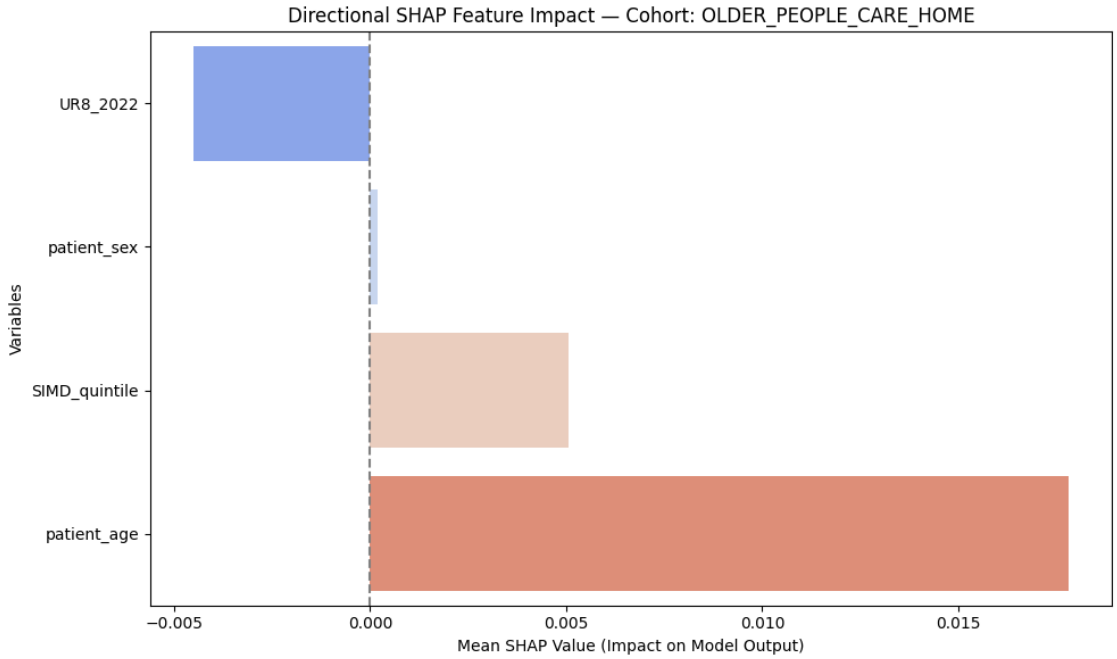


```
======== Cohort: OLDER_PEOPLE_CARE_HOME =========
Train size: 59486
Test size: 25494

/mnt/homes/vayly01/myenv/lib/python3.13/site-packages/xgboost/training.py:183:
UserWarning: [09:35:13] WARNING: /workspace/src/learner.cc:738:
Parameters: { "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)


Model Performance Metrics:
    Metric    Value
  Accuracy 0.856005
 Precision 0.856257
    Recall 0.999588
  F1 Score 0.922387
 AUC Score 0.574277
  Log Loss 0.407426
```

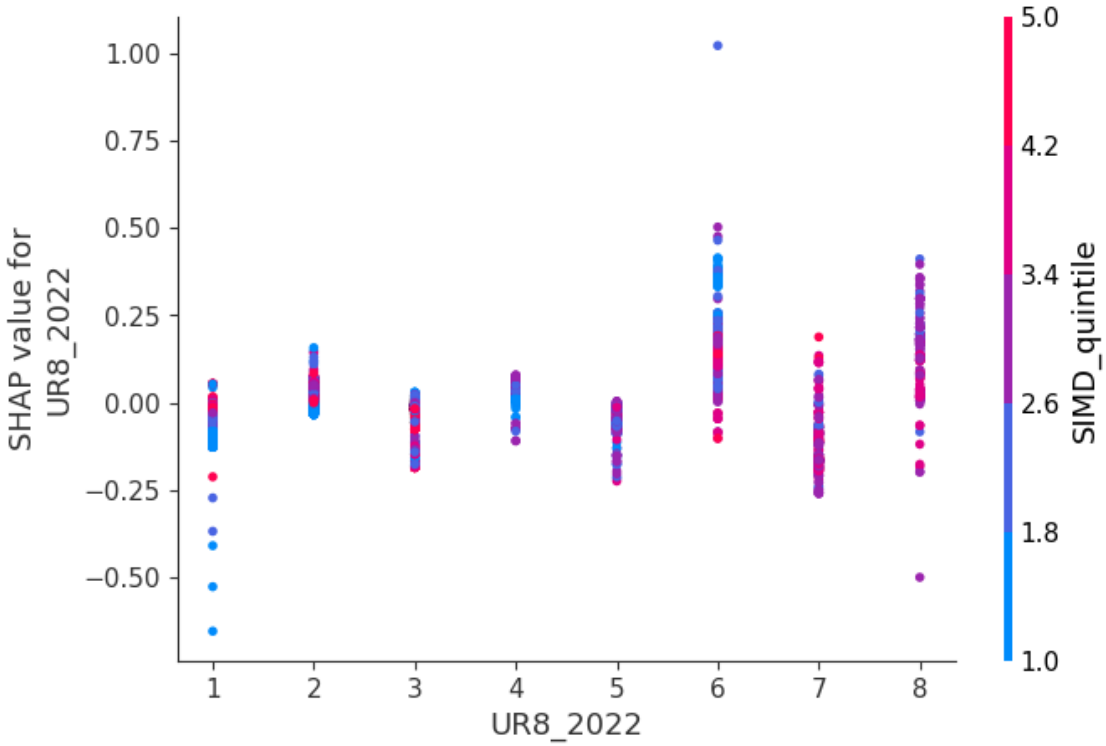



```
Top Feature Importances:
      Feature  Importance
  patient_age      0.5379
SIMD_quintile      0.1792
     UR8_2022      0.1522
  patient_sex      0.1307
```

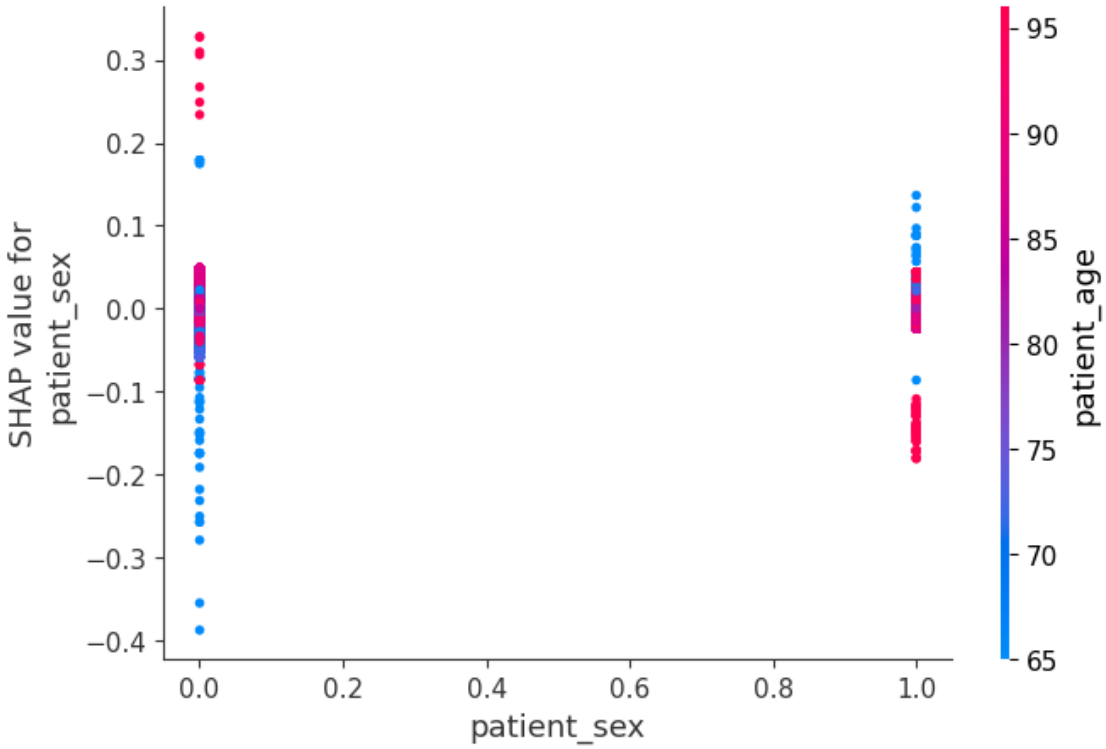
```
Sampling 20000 rows for SHAP from test set (size: 25494)
Actual SHAP sample size: 20000
 95%|=================== | 19000/20000 [00:19<00:01]

Directional SHAP Bar Plot - Cohort: OLDER_PEOPLE_CARE_HOME
/tmp/ipykernel_4511/78395491.py:135: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=shap_summary_directional.values,
y=shap_summary_directional.index, palette="coolwarm")
```



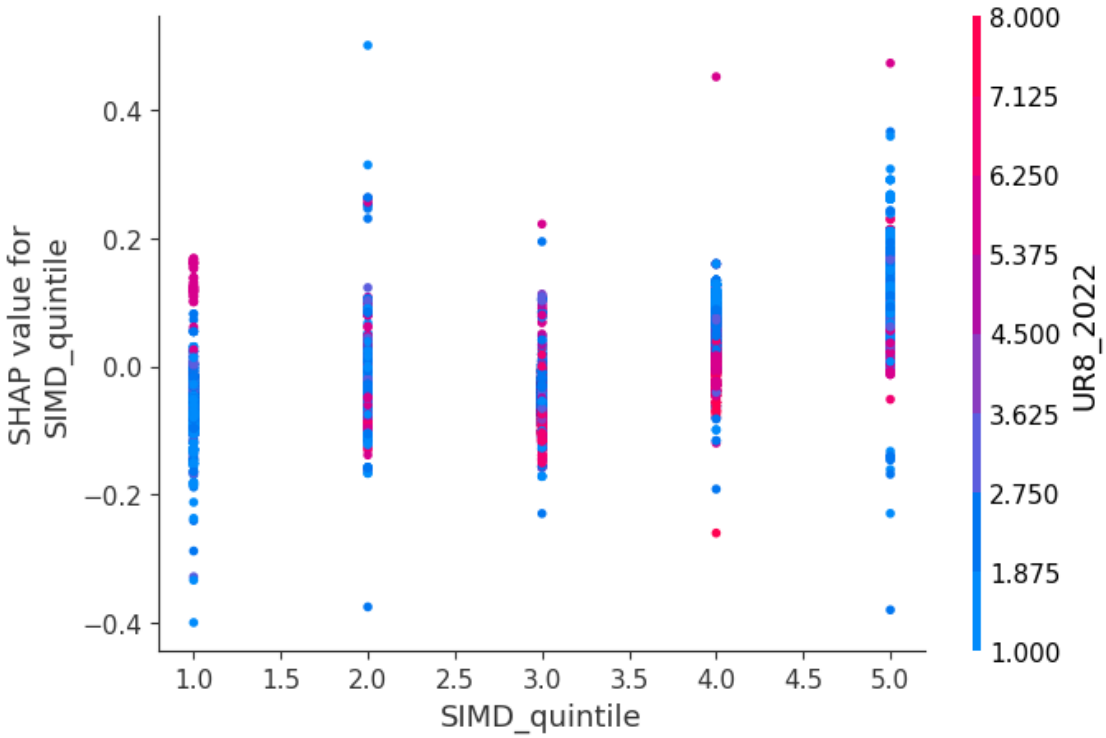

```
SHAP Dependence Plot in Cohort OLDER_PEOPLE_CARE_HOME for: UR8_2022
```



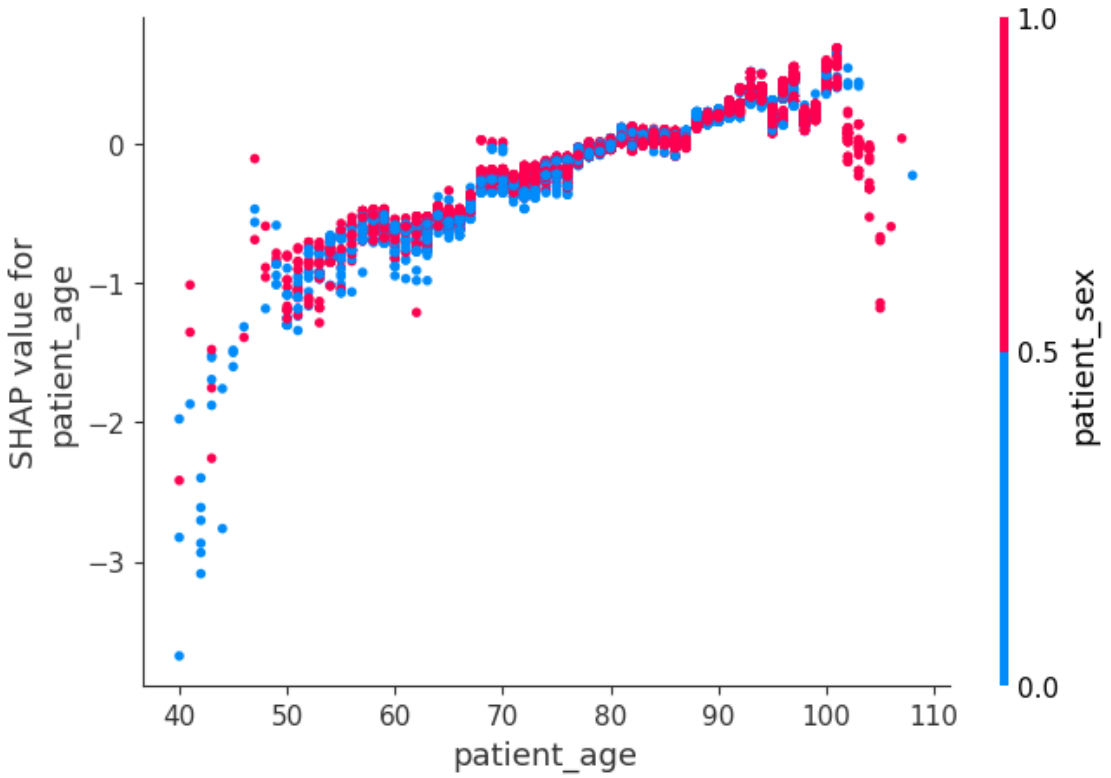
SHAP Dependence Plot in Cohort OLDER_PEOPLE_CARE_HOME for: patient_sex



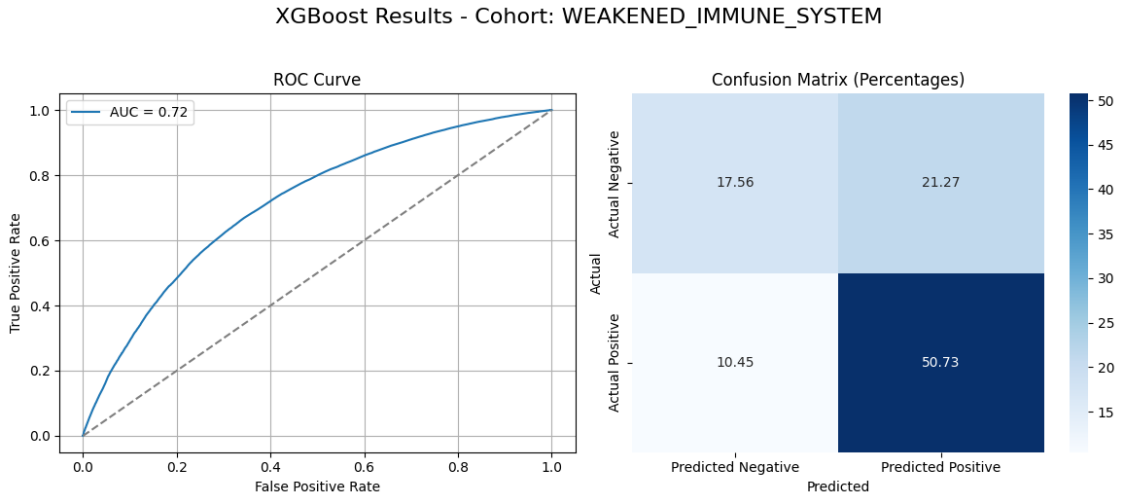
SHAP Dependence Plot in Cohort OLDER_PEOPLE_CARE_HOME for: SIMD_quintile



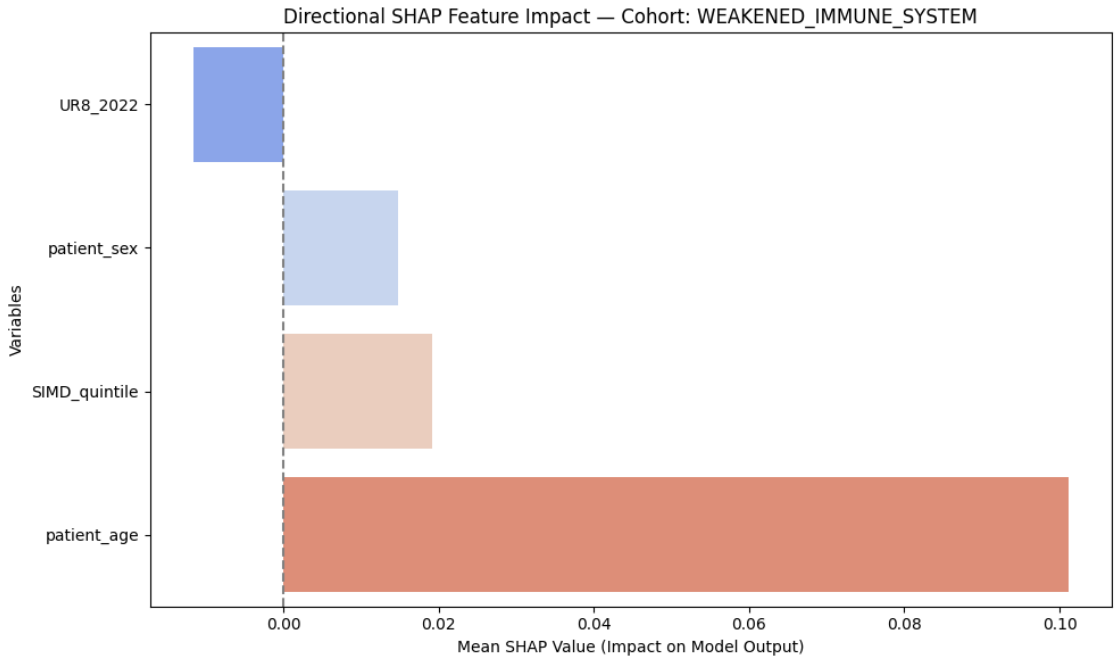
SHAP Dependence Plot in Cohort OLDER_PEOPLE_CARE_HOME for: patient_age



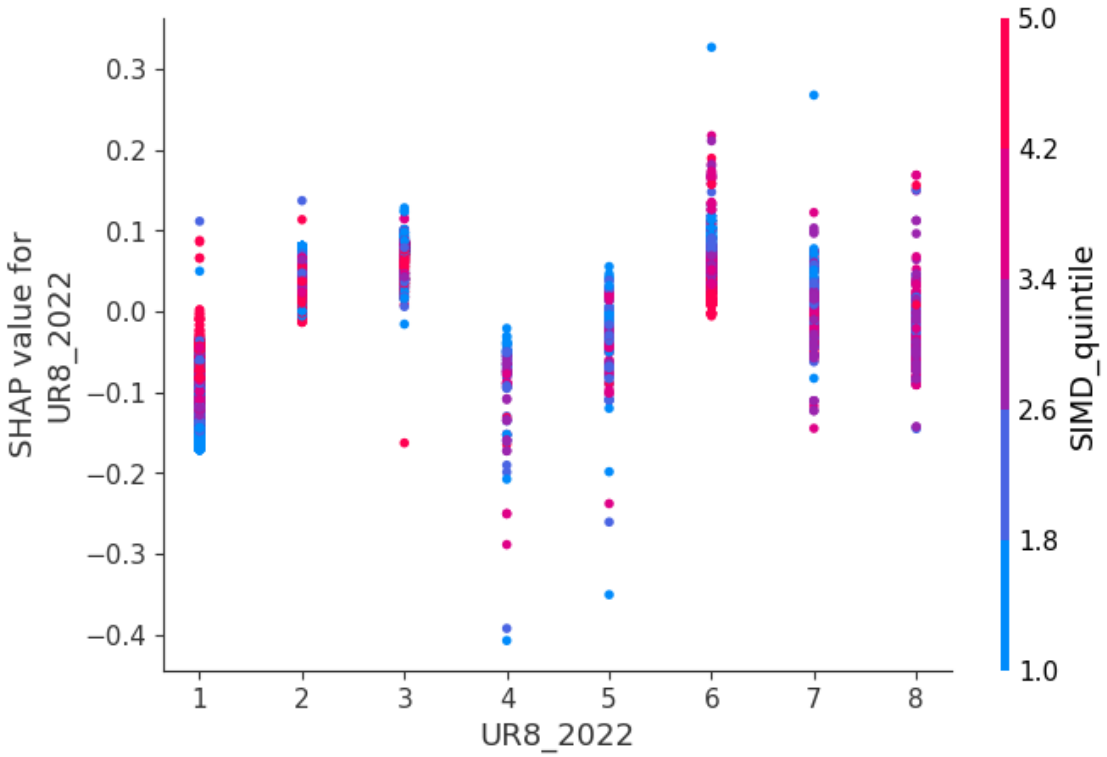

```
======== Cohort: WEAKENED_IMMUNE_SYSTEM =========
Train size: 230736
Test size: 98887

/mnt/homes/vayly01/myenv/lib/python3.13/site-packages/xgboost/training.py:183:
UserWarning: [09:35:36] WARNING: /workspace/src/learner.cc:738:
Parameters: { "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)


Model Performance Metrics:
    Metric    Value
 Accuracy 0.682891
Precision 0.704632
    Recall 0.829261
 F1 Score 0.761884
AUC Score 0.715265
 Log Loss 0.597928
```



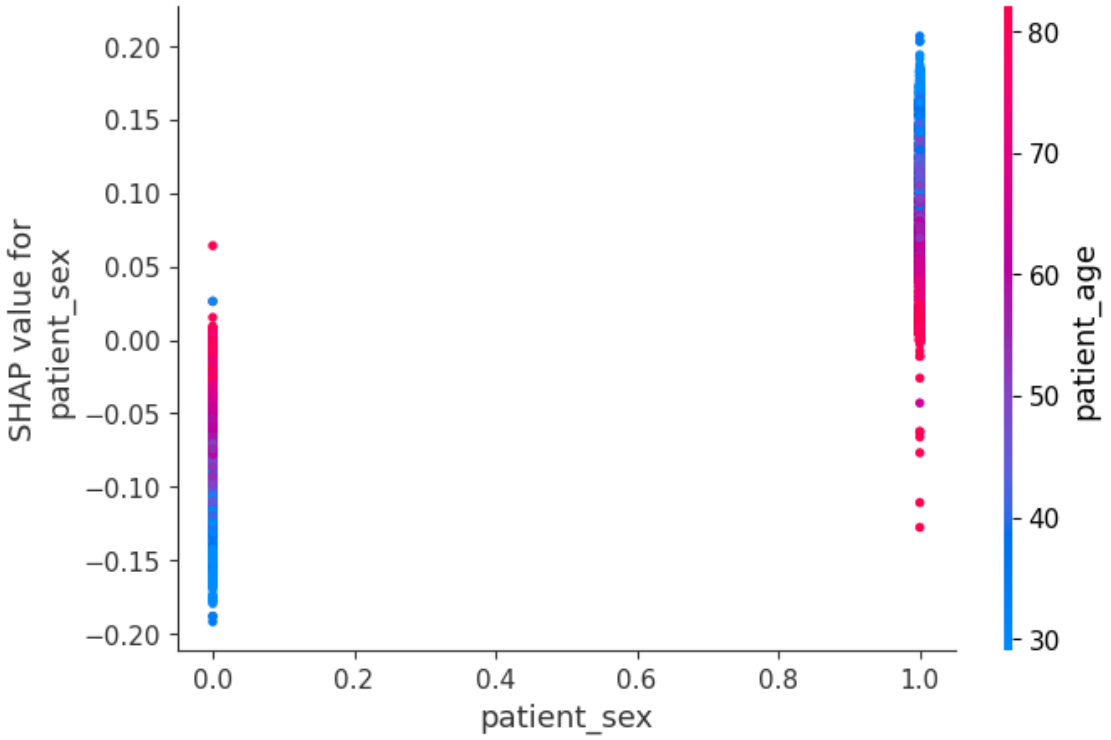

```
Top Feature Importances:
      Feature  Importance
  patient_age      0.7077
SIMD_quintile      0.2057
  patient_sex      0.0603
     UR8_2022      0.0263

Sampling 20000 rows for SHAP from test set (size: 98887)
Actual SHAP sample size: 20000
 97%|=================== | 19379/20000 [00:27<00:00]


Directional SHAP Bar Plot - Cohort: WEAKENED_IMMUNE_SYSTEM

/tmp/ipykernel_4511/78395491.py:135: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=shap_summary_directional.values,
y=shap_summary_directional.index, palette="coolwarm")
```



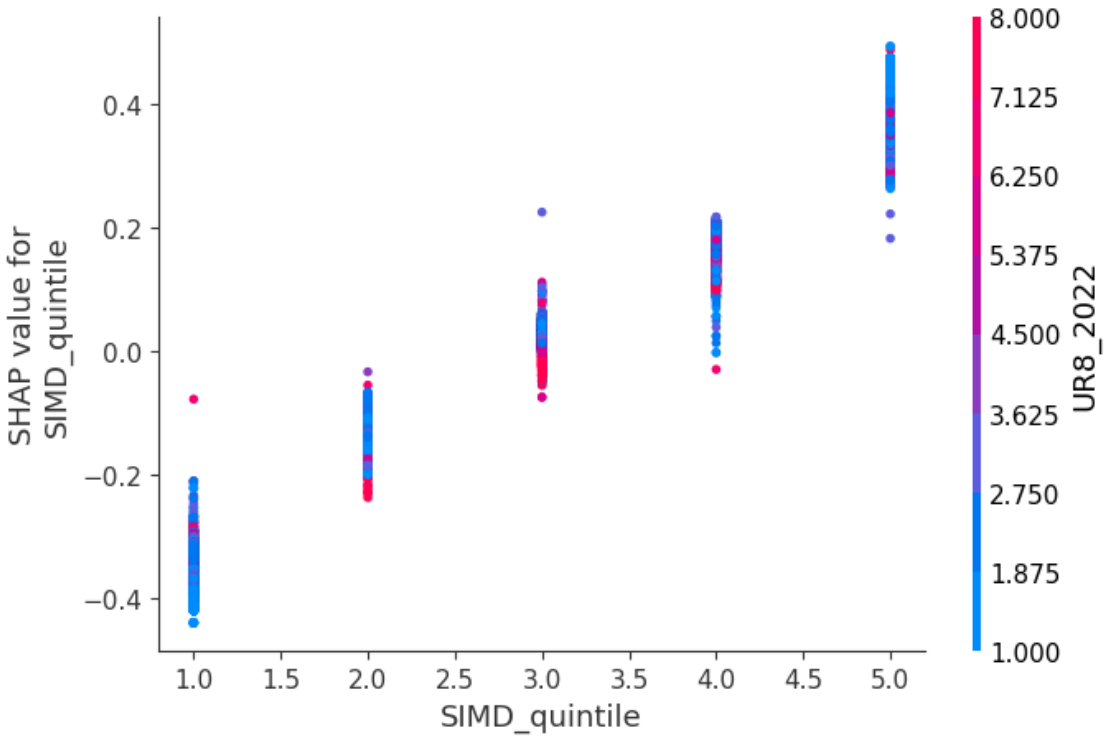
SHAP Dependence Plot in Cohort WEAKENED_IMMUNE_SYSTEM for: UR8_2022



SHAP Dependence Plot in Cohort WEAKENED_IMMUNE_SYSTEM for: patient_sex



SHAP Dependence Plot in Cohort WEAKENED_IMMUNE_SYSTEM for: SIMD_quintile



SHAP Dependence Plot in Cohort WEAKENED_IMMUNE_SYSTEM for: patient_age