Model Validation Report

# Model Performance Metrics

This section includes generic model performance metrics.

**Prediction Distribution**

Probability distribution from model prediction output.

|  |  |  |
| --- | --- | --- |
| Testing Dataset |  |  |
| Test | Model |  |
|  | Benchmark Model |  |

**Accuracy**

From all the classes, how many are predicted correctly.

|  |  |  |
| --- | --- | --- |
| Testing Dataset | Model | Benchmark Model |
| Test | 0.965 | 0.947 |

**Precision**

= True Positive / (True Positive + False Positive)   
 From all that are predicted as positive, how many are actually positive.

|  |  |  |
| --- | --- | --- |
| Testing Dataset | Model | Benchmark Model |
| Test | 0.976 | 0.974 |

**Recall**

= True Positive / (True Positive + False Negative)   
 From all the positive classes, how many are predicted correctly.

|  |  |  |
| --- | --- | --- |
| Testing Dataset | Model | Benchmark Model |
| Test | 0.93 | 0.884 |

**F1 Score**

= 2\*Recall\*Precision / (Recall + Precision).   
 F1-score helps to measure Recall and Precision at the same time.

|  |  |  |
| --- | --- | --- |
| Testing Dataset | Model | Benchmark Model |
| Test | 0.952 | 0.927 |

**Receiver Operating Characteristic (ROC) Curve**

Shows the performance of the model at all classification thresholds. Higher AUC indicates a better model.

|  |  |  |  |
| --- | --- | --- | --- |
| Testing Dataset |  |  | AUC |
| Test | Model |  | 0.998362 |
|  | Benchmark Model |  | 0.946282 |

**Precision Recall Curve**

Shows the tradeoff between precision and recall for different thresholds in binary classification problems. A high area under the curve represents both high recall and high precision, where high precision relates to a low false positive rate, and high recall relates to a low false negative rate. Please refer to the link for more details: https://medium.com/@douglaspsteen/precision-recall-curves-d32e5b290248

|  |  |  |  |
| --- | --- | --- | --- |
| Testing Dataset |  |  | AUC |
| Test | Model |  | 0.997325 |
|  | Benchmark Model |  | 0.957951 |

# Statistical Metrics

This section includes statistical metrics.

**PSI**

A metric to measure how much a variable has shifted in distribution between two samples over time. Please refer the link for more details: https://towardsdatascience.com/psi-and-csi-top-2-model-monitoring-metrics-924a2540bed8

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Testing Dataset |  | PSI |  | train\_count | train\_perc | test\_count | test\_perc | index\_value |
| Test | Model | 0.06749121 | (0.00, 0.10] | 111 | 0.369 | 32 | 0.364 | 0.0 |
|  |  |  | (0.10, 0.20] | 18 | 0.06 | 7 | 0.08 | 0.006 |
|  |  |  | (0.20, 0.30] | 3 | 0.01 | 3 | 0.034 | 0.029 |
|  |  |  | (0.30, 0.40] | 0 | 0.0 | 2 | 0.023 | 0.0 |
|  |  |  | (0.40, 0.50] | 0 | 0.0 | 3 | 0.034 | 0.0 |
|  |  |  | (0.50, 0.60] | 2 | 0.007 | 1 | 0.011 | 0.002 |
|  |  |  | (0.60, 0.70] | 2 | 0.007 | 0 | 0.0 | 0.0 |
|  |  |  | (0.70, 0.80] | 9 | 0.03 | 3 | 0.034 | 0.001 |
|  |  |  | (0.80, 0.90] | 14 | 0.047 | 5 | 0.057 | 0.002 |
|  |  |  | (0.90, 1.00] | 142 | 0.472 | 32 | 0.364 | 0.028 |
|  | Benchmark Model | 0.00955804 | (0.00, 0.10] | 0 | 0.0 | 0 | 0.0 | 0.0 |
|  |  |  | (0.10, 0.20] | 0 | 0.0 | 0 | 0.0 | 0.0 |
|  |  |  | (0.20, 0.30] | 0 | 0.0 | 0 | 0.0 | 0.0 |
|  |  |  | (0.30, 0.40] | 0 | 0.0 | 0 | 0.0 | 0.0 |
|  |  |  | (0.40, 0.50] | 2 | 0.012 | 1 | 0.025 | 0.01 |
|  |  |  | (0.50, 0.60] | 0 | 0.0 | 0 | 0.0 | 0.0 |
|  |  |  | (0.60, 0.70] | 3 | 0.018 | 0 | 0.0 | 0.0 |
|  |  |  | (0.70, 0.80] | 0 | 0.0 | 0 | 0.0 | 0.0 |
|  |  |  | (0.80, 0.90] | 0 | 0.0 | 0 | 0.0 | 0.0 |
|  |  |  | (0.90, 1.00] | 166 | 0.971 | 39 | 0.975 | 0.0 |
|  |  |  |  |  |  |  |  |  |

**CSI**

It compares the distribution of an independent variable in the training data set to a testing data set. It detects shifts in the distributions of input variables that are submitted for scoring over time. Please refer the link for more details: https://towardsdatascience.com/psi-and-csi-top-2-model-monitoring-metrics-924a2540bed8

**perimeter\_mean**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Testing Dataset |  | CSI |  | train\_count | train\_perc | test\_count | test\_perc | index\_value |
| Test | Model | 0.049 | (43.79, 58.26] | 13 | 0.029 | 1 | 0.009 | 0.023 |
|  |  |  | (58.26, 72.73] | 73 | 0.161 | 20 | 0.177 | 0.002 |
|  |  |  | (72.73, 87.20] | 151 | 0.333 | 34 | 0.301 | 0.003 |
|  |  |  | (87.20, 101.67] | 92 | 0.203 | 24 | 0.212 | 0.0 |
|  |  |  | (101.67, 116.14] | 47 | 0.104 | 15 | 0.133 | 0.007 |
|  |  |  | (116.14, 130.62] | 41 | 0.09 | 8 | 0.071 | 0.005 |
|  |  |  | (130.62, 145.09] | 26 | 0.057 | 8 | 0.071 | 0.003 |
|  |  |  | (145.09, 159.56] | 6 | 0.013 | 2 | 0.018 | 0.002 |
|  |  |  | (159.56, 174.03] | 2 | 0.004 | 1 | 0.009 | 0.004 |
|  |  |  | (174.03, 188.50] | 3 | 0.007 | 0 | 0.0 | 0.0 |
|  |  |  |  |  |  |  |  |  |

**symmetry\_mean**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Testing Dataset |  | CSI |  | train\_count | train\_perc | test\_count | test\_perc | index\_value |
| Test | Model | 0.036 | (0.11, 0.13] | 3 | 0.007 | 1 | 0.009 | 0.001 |
|  |  |  | (0.13, 0.15] | 30 | 0.066 | 5 | 0.044 | 0.009 |
|  |  |  | (0.15, 0.17] | 108 | 0.237 | 25 | 0.221 | 0.001 |
|  |  |  | (0.17, 0.19] | 128 | 0.281 | 36 | 0.319 | 0.005 |
|  |  |  | (0.19, 0.20] | 109 | 0.24 | 27 | 0.239 | 0.0 |
|  |  |  | (0.20, 0.22] | 48 | 0.105 | 13 | 0.115 | 0.001 |
|  |  |  | (0.22, 0.24] | 16 | 0.035 | 4 | 0.035 | 0.0 |
|  |  |  | (0.24, 0.26] | 9 | 0.02 | 1 | 0.009 | 0.009 |
|  |  |  | (0.26, 0.28] | 3 | 0.007 | 0 | 0.0 | 0.0 |
|  |  |  | (0.28, 0.30] | 1 | 0.002 | 1 | 0.009 | 0.011 |
|  |  |  |  |  |  |  |  |  |

**Kolmogorov–Smirnov statistic**

:Quantifies a distance of the distribution within the training sample and testing sample, or between the two. Please refer the link for more details: https://en.wikipedia.org/wiki/Kolmogorov%E2%80%93Smirnov\_test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testing Dataset |  | Train | Test | Train vs. Test |
| Test | Model | 0.721 | 0.119 | 0.135 |
|  | Benchmark Model | 0.715 | 0.073 | 0.029 |

**Gini Impurity**

A measurement used to build Decision Trees to determine how the features of a dataset should split nodes to form the tree. It calculates the probability of a certain randomly selected feature that is classified incorrectly. An attribute with the smallest Gini Impurity should be selected for splitting the node. Please refer to the link for more details: https://www.learndatasci.com/glossary/gini-impurity/

|  |  |  |  |
| --- | --- | --- | --- |
| Testing Dataset | Feature | Model | Benchmark Model |
| Test |  |  |  |
|  | radius\_mean | 0 | 0 |
|  | texture\_mean | 0 | 0 |
|  | perimeter\_mean | 0 | 0 |
|  | area\_mean | 0 | 0 |
|  | smoothness\_mean | 0 | 0 |
|  | compactness\_mean | 0 | 0 |
|  | concavity\_mean | 0 | 0 |
|  | concave points\_mean | 0 | 0 |
|  | symmetry\_mean | 0 | 0 |
|  | fractal\_dimension\_mean | 0 | 0 |
|  | radius\_se | 0 | 0 |
|  | texture\_se | 0 | 0 |
|  | perimeter\_se | 0 | 0 |
|  | area\_se | 0 | 0 |
|  | smoothness\_se | 0 | 0 |
|  | compactness\_se | 0 | 0 |
|  | concavity\_se | 0 | 0 |
|  | concave points\_se | 0 | 0 |
|  | symmetry\_se | 0 | 0 |
|  | fractal\_dimension\_se | 0 | 0 |
|  | radius\_worst | 0 | 0 |
|  | texture\_worst | 0 | 0 |
|  | perimeter\_worst | 0 | 0 |
|  | area\_worst | 0 | 0 |
|  | smoothness\_worst | 0 | 0 |
|  | compactness\_worst | 0 | 0 |
|  | concavity\_worst | 0 | 0 |
|  | concave points\_worst | 0 | 0 |
|  | symmetry\_worst | 0 | 0 |
|  | fractal\_dimension\_worst | 0 | 0 |
|  |  |  |  |

# Transparency Metrics

This section includes LIME and SHAP interpretability under the framework of MAS's FEAT metrics.

Global interpretability refers to the ability to explain the behavior of a model across its entire input space. It focuses on understanding how the model works in general, rather than on a specific instance. Global interpretability techniques aim to provide insights into the relationships between features and how they contribute to the overall behavior of the model.

**LIME Interpretability**

LIME is a method that helps to explain the predictions of any machine learning model by approximating the model locally with an interpretable model. LIME works by perturbing the input data and observing the changes in the output of the model. The perturbed data is then used to train a new, interpretable model that approximates the behavior of the original model in that local region. The resulting model can then be used to explain why the original model made a particular prediction.   
 The LIME chart shows the coefficients assigned to each feature by the interpretable model. The coefficients indicate the direction and magnitude of the impact of each feature on the model's output. Positive coefficients indicate that the feature has a positive impact on the Class1, while negative coefficients indicate a positive impact on Class 0.   
 Please refer to the link for more details: https://arxiv.org/abs/1602.04938)

**Global LIME Interpretability**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testing Dataset | Model | | Benchmark Model | |
| Test |  | |  | |
|  | area\_worst | -0.049212 | concave points\_mean | 0.212603 |
|  | perimeter\_worst | -0.047976 | concave points\_se | -0.07688 |
|  | radius\_worst | -0.042589 | perimeter\_worst | 0.075082 |
|  | concave points\_worst | -0.037297 | texture\_worst | 0.072188 |
|  | concave points\_mean | -0.036125 | area\_se | 0.062821 |
|  | concavity\_worst | -0.028271 | concave points\_worst | 0.028526 |
|  | texture\_worst | -0.026239 | perimeter\_se | 0.02834 |
|  | area\_se | -0.022849 | texture\_mean | 0.028332 |
|  | texture\_mean | -0.022609 | smoothness\_mean | 0.016566 |
|  | concavity\_mean | -0.019982 | radius\_worst | -0.012524 |
|  | area\_mean | -0.012998 | symmetry\_worst | -0.011367 |
|  | radius\_se | -0.010164 | symmetry\_se | 0.008334 |
|  | smoothness\_worst | -0.008896 | compactness\_worst | 0.007931 |
|  | perimeter\_mean | -0.008421 | radius\_se | -0.007414 |
|  | radius\_mean | -0.008323 | symmetry\_mean | 0.007393 |
|  | smoothness\_mean | -0.00658 | smoothness\_worst | 0.00709 |
|  | compactness\_worst | -0.005556 | fractal\_dimension\_se | -0.007019 |
|  | perimeter\_se | -0.005082 | compactness\_mean | 0.006038 |
|  | symmetry\_worst | -0.004696 | concavity\_mean | 0.005987 |
|  | compactness\_se | 0.00443 | fractal\_dimension\_mean | -0.004746 |
|  | compactness\_mean | 0.004159 | area\_mean | 0.004323 |
|  | texture\_se | -0.003594 | compactness\_se | -0.003988 |
|  | concave points\_se | 0.003285 | fractal\_dimension\_worst | 0.002106 |
|  | symmetry\_mean | -0.002781 | radius\_mean | -0.001982 |
|  | fractal\_dimension\_se | 0.00204 | area\_worst | -0.001606 |
|  | symmetry\_se | 0.001783 | smoothness\_se | -0.001271 |
|  | concavity\_se | 0.000486 | perimeter\_mean | -0.00119 |
|  | fractal\_dimension\_mean | -0.000453 | concavity\_se | -0.001148 |
|  | fractal\_dimension\_worst | 0.000294 | texture\_se | 0.000323 |
|  | smoothness\_se | 4.4e-05 | concavity\_worst | 0.00016 |
|  |  |  |  |  |

**Importance**

SHAP is a method for interpreting the output of any machine learning model by computing the contribution of each feature to the final prediction. SHAP is based on the concept of Shapley values from cooperative game theory, which measures the marginal contribution of each feature to the final prediction by averaging over all possible feature combinations. By computing the SHAP values for each feature, we can determine which features had the most significant impact on the prediction.   
 The SHAP plot shows the average impact of each feature on the model's output across all instances in the dataset. The features are sorted in descending order of importance, and the magnitude and direction of the impact are shown.   
 Please refer to the link for more details: https://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

**Global Importance**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testing Dataset | Model | | Benchmark Model | |
| Test |  | |  | |
|  | radius\_worst | 0.003013 | perimeter\_worst | -0.003275 |
|  | perimeter\_worst | -0.002687 | concave points\_mean | -0.002244 |
|  | area\_worst | -0.002462 | concave points\_worst | 0.001688 |
|  | area\_mean | 0.001875 | concave points\_se | 0.00165 |
|  | texture\_worst | 0.001706 | perimeter\_se | -0.001281 |
|  | concave points\_worst | -0.001681 | radius\_worst | 0.001138 |
|  | texture\_mean | -0.001481 | area\_se | 0.000888 |
|  | concave points\_mean | 0.00135 | perimeter\_mean | 0.000825 |
|  | concavity\_worst | -0.000725 | texture\_worst | 0.000413 |
|  | compactness\_mean | 0.00065 | texture\_mean | 0.0002 |
|  | smoothness\_worst | 0.000625 | radius\_mean | 0 |
|  | symmetry\_worst | -0.000569 | area\_mean | 0 |
|  | radius\_se | -0.000531 | smoothness\_mean | 0 |
|  | radius\_mean | -0.000525 | compactness\_mean | 0 |
|  | concave points\_se | 0.000506 | concavity\_mean | 0 |
|  | smoothness\_se | 0.000488 | symmetry\_mean | 0 |
|  | compactness\_se | -0.000469 | fractal\_dimension\_mean | 0 |
|  | area\_se | 0.000456 | radius\_se | 0 |
|  | concavity\_mean | -0.000438 | texture\_se | 0 |
|  | compactness\_worst | 0.000344 | smoothness\_se | 0 |
|  | fractal\_dimension\_se | 0.000325 | compactness\_se | 0 |
|  | symmetry\_mean | 0.000313 | concavity\_se | 0 |
|  | smoothness\_mean | 0.000244 | symmetry\_se | 0 |
|  | fractal\_dimension\_worst | -0.000219 | fractal\_dimension\_se | 0 |
|  | fractal\_dimension\_mean | -0.000131 | area\_worst | 0 |
|  | concavity\_se | 0.000125 | smoothness\_worst | 0 |
|  | symmetry\_se | -7.5e-05 | compactness\_worst | 0 |
|  | texture\_se | 6.9e-05 | concavity\_worst | 0 |
|  | perimeter\_se | -5e-05 | symmetry\_worst | 0 |
|  | perimeter\_mean | -4.4e-05 | fractal\_dimension\_worst | 0 |
|  |  |  |  |  |