**Model Validation Report**

<<modelop.storedModel.modelMetaData.name>>

<<modelop.storedModel.modelMetaData.description>>

<<modelop.storedModel.id>>

**Prepared by:** *Enter the document preparer’s name here*

**Model Developed by:** <<modelop.storedModel.createdBy>>

**Model Developed on:** <<modelop.deployableModel.createdDate>>

**Executive Summary**

*Provide a concise summary description of the model and its proposed use(s), model results, and key conclusions from the model development process. A reader should be able to obtain a solid understanding of the model solely from reading the Executive Summary.*

**Statistical Analysis / Testing Summary**

|  |  |  |
| --- | --- | --- |
| **Category** | **Passes** | **Reason** |
| Performance | <<modelop.modelTestResult.dmnRuleResults.[testCategory=Performance].passes>> | <<modelop.modelTestResult.dmnRuleResults.[testCategory=Performance].reason>> |
| Data Drift - Kolmogorov Smirnov | <<modelop.modelTestResult.dmnRuleResults.[testCategory=KS Data Drift].passes>> | <<modelop.modelTestResult.dmnRuleResults.[testCategory=KS Data Drift].reason>> |
| Concept Drift - Kolmogorov Smirnov | <<modelop.modelTestResult.dmnRuleResults.[testCategory=KS Concept Drift].passes>> | <<modelop.modelTestResult.dmnRuleResults.[testCategory=KS Concept Drift].reason>> |
| Characteristic Stability | <<modelop.modelTestResult.dmnRuleResults.[testCategory=Characteristic Stability].passes>> | <<modelop.modelTestResult.dmnRuleResults.[testCategory=Characteristic Stability].reason>> |
| Bias Disparity | <<modelop.modelTestResult.dmnRuleResults.[testCategory=Bias Disparity].passes>> | <<modelop.modelTestResult.dmnRuleResults.[testCategory=Bias Disparity].reason>> |
| Autocorrelation | <<modelop.modelTestResult.dmnRuleResults.[testCategory=Autocorrelation].passes>> | <<modelop.modelTestResult.dmnRuleResults.[testCategory=Autocorrelation].reason>> |
| Homoscedacticity | <<modelop.modelTestResult.dmnRuleResults.[testCategory=Homoscedacticity].passes>> | <<modelop.modelTestResult.dmnRuleResults.[testCategory=Homoscedacticity].reason>> |
| Normality | <<modelop.modelTestResult.dmnRuleResults.[testCategory=Normality].passes>> | <<modelop.modelTestResult.dmnRuleResults.[testCategory=Normality].reason>> |
| Linearity | <<modelop.modelTestResult.dmnRuleResults.[testCategory=Linearity].passes>> | <<modelop.modelTestResult.dmnRuleResults.[testCategory=Linearity].reason>> |
| Mulitcolinearity | <<modelop.modelTestResult.dmnRuleResults.[testCategory=Mulitcolinearity].passes>> | <<modelop.modelTestResult.dmnRuleResults.[testCategory=Mulitcolinearity].reason>> |

**Statistical Analysis**

**Performance Metrics**

*Provide a textual analysis of the classification metrics and what they indicate*

|  |
| --- |
| <<modelop.modelTestResult.testResults.(performance).test\_id=performance\_regression\_metrics>> |

Table - Classification Metrics

**Stability Analysis**

*Provide a textual analysis of the stability analysis and what they indicate*

<<modelopgraph.stability.eOverallQual\_TotalSF>>

|  |
| --- |
| <<modelop.modelTestResult.testResults.(stability)[0].(values).(eOverallQual\_TotalSF).(stability\_analysis\_table)>> |

Table 2 - Stability for Feature eOverallQual\_TotalSF

<<modelopgraph.stability.GrLivArea>>

|  |
| --- |
| <<modelop.modelTestResult.testResults.(stability)[0].(values).(GrLivArea).(stability\_analysis\_table)>> |

Table 3 - Stability for Feature GrLivArea

<<modelopgraph.stability.KitchenQual>>

|  |
| --- |
| <<modelop.modelTestResult.testResults.(stability)[0].(values).(KitchenQual).(stability\_analysis\_table)>> |

Table 4 - Stability for Feature KitchenQual

<<modelopgraph.stability.BsmtQual >>

|  |
| --- |
| <<modelop.modelTestResult.testResults.(stability)[0].(values).(BsmtQual).(stability\_analysis\_table)>> |

Table 5 - Stability for Feature BsmtQual

<<modelopgraph.stability.TotalBsmtSF >>

|  |
| --- |
| <<modelop.modelTestResult.testResults.(stability)[0].(values).(TotalBsmtSF).(stability\_analysis\_table)>> |

Table 6 - Stability for Feature TotalBsmtSF

<<modelopgraph.stability.prediction>>

|  |
| --- |
| <<modelop.modelTestResult.testResults.(stability)[0].(values).(prediction).(stability\_analysis\_table)>> |

Table 7 - Stability for Feature prediction

**Statistical Diagnostics**

*Below is a detailed list of the standardized statistical diagnostics tests for the model.*

Autocorrelation Metrics:

|  |
| --- |
| *Durbin-Watson Test:*  *dw\_statistic* : <<modelop.modelTestResult.testResults.(dw\_statistic)>> |

Normality Metrics:

|  |
| --- |
| *Kolmogorov-Smirnov Test:*  *ks\_statistic* : <<modelop.modelTestResult.testResults.(ks\_statistic)>>  *ks\_p\_value*: <<modelop.modelTestResult.testResults.(ks\_p\_value)>> |

|  |
| --- |
| *Cramer-von Mises Test:*  *cvm\_statistic*: <<modelop.modelTestResult.testResults.(cvm\_statistic)>>  *cvm\_p\_value*: <<modelop.modelTestResult.testResults.(cvm\_p\_value)>> |

|  |
| --- |
| *Anderson-Darling Test:*  *ad\_statistic:*: <<modelop.modelTestResult.testResults.(ad\_statistic)>>  *ad\_p\_value*: <<modelop.modelTestResult.testResults.(ad\_p\_value)>> |

Linearity Metrics:

|  |
| --- |
| *Pearson Correlation:*  <<modelop.modelTestResult.testResults.(linearity)[0].values)>> |

Homoscedasticity Metrics:

|  |
| --- |
| *Ljung-Box Q Test:*  <<modelop.modelTestResult.testResults.(homoscedasticity\_ljung\_box)[0].values)>> |

|  |
| --- |
| *Engle's Lagrange Multiplier Test:*  *engle\_lm\_statistic:* <<modelop.modelTestResult.testResults.(engle\_lm\_statistic)>>  *engle\_lm\_p\_value*: <<modelop.modelTestResult.testResults.(engle\_lm\_p\_value)>> |

|  |
| --- |
| *Breusch-Pagan Test:*  *breusch\_pagan\_lm\_statistic:* <<modelop.modelTestResult.testResults.(breusch\_pagan\_lm\_statistic)>>  *breusch\_pagan\_lm\_p\_value*: <<modelop.modelTestResult.testResults.(breusch\_pagan\_lm\_p\_value)>>  *breusch\_pagan\_f\_statistic:* <<modelop.modelTestResult.testResults.(breusch\_pagan\_f\_statistic)>>  *breusch\_pagan\_f\_p\_value*: <<modelop.modelTestResult.testResults.(breusch\_pagan\_f\_p\_value)>> |

Variance Inflation Factors:

|  |
| --- |
| *Ljung-Box Q Test:*  <<modelop.modelTestResult.testResults.(multicollinearity)[0].values)>> |

**Ethical Fairness Analysis**

*Provide a textual analysis of bias results and control measures put in place to eliminate the bias in the provided data and results.*

**Bias Analysis for Protected Classes**

<<modelopgraph.groupbias.gender>>