Munifa Nusrat

ID – w2021652

**Task (1) – Domain Understanding: Classification**

**a)**

|  |  |
| --- | --- |
| **Attribute Name** | **Retain or Drop** |
| Test ID | Drop |
| Systemic Illness | Drop |
| Sore Throat | Retain |
| Rectal Pain | Retain |
| Penile Oedema | Retain |
| Oral Lesions | Retain |
| Solitary Lesion | Retain |
| Swollen Tonsils | Retain |
| HIV Infection | Retain |
| Red blood cells | Retain |
| White blood cells | Retain |
| Home ownership | Drop |
| Age | Retain |
| Month of Birth | Drop |
| Health Insurance | Drop |
| Sexually Transmitted Infection | Retain |
| MPOX | Retain |

**Task (2) – Data Understanding: Producing Your Experimental Designing**

1. **basic statistical description**

A screenshot of a computer

Description automatically generated

**A screenshot of a computer

Description automatically generated Measurement scale type**

**Distribution of the class variable**

A screenshot of a computer

Description automatically generated **Task (3) – Data Preparation: Cleaning and Transforming your data.**

**a)**

|  |  |  |
| --- | --- | --- |
| **Dataset or Variable Issue?** | **Name of variable** | **Issue description** |
| Dataset Issue | Null Values | The issue with the dataset is that it contains null values. |
| Variable Issue | Age | The issue with this variable is that it contains negative values, large, unusual data and values in alphabets. |
| Variable issue | White Blood cell | The values are large, and they vary from the rest of the values of the rest of the values in the dataset. |
| Variable Issue | Oral Lesions | The issue with this variable is that it contains both categorical and numeric values – YES, NO,0,1. |
| Variable Issue | Red Blood Cell | The values are large, and they vary from the rest of the values of the rest of the values in the dataset. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Dataset or Variable Issue? | Name of variable | The Issue | Solution | Justification |
| Dataset Issue | Null Values | Null values. | Remove them. | The number of null values is very few compared to the dataset. |
| Variable issue | Age | Negative values, unusual data, and values in alphabets. | Remove the null value, remove the unusual data, we changed all the format to float. | As age is numeric, all values should be float. |
| Variable issue | White Blood cell | Huge values and they differ from the rest of the values. | Min-Max Scaling | By normalizing, all values are in the same scale type. |
| Variable Issue | Oral Lesions | Values are categorical - ‘YES’ and ‘NO’ and numeric - 0 and 1. | Min-Max Scaling | By normalizing, all values are in the same scale type. |
| Variable Issue | Red Blood Cell | Huge values and they differ from the rest of the values. | Min-Max Scaling | By normalizing, all values are in the same scale type. |

**c)** The suggested solution involves creating a function that includes preprocessing, data splitting, and normalization-

**A screenshot of a computer

Description automatically generated**

**Normalization-**

**A screenshot of a computer

Description automatically generated**

**Task (4) – Modelling: Create Predictive Classification Models.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithm**  **Name** | **Type of**  **Algorithm** | **Learnable Parameters** | **Possible Hyper -**  **Parameters** | **Python package**  **source code to call**  **the algorithm** |
| **LR** | Supervised learning Logistic Regression | **['coef\_', 'intercept\_']** | **{'penalty': ['l1', 'l2'], 'C': [0.001, 0.01, 0.1, 1, 10, 100, 1000], 'fit\_intercept': [True, False], 'solver': ['liblinear', 'saga'], 'max\_iter': [100, 200, 300, 400, 500]}** | **from sklearn.linear\_model import LogisticRegression** |
| **DT** | Supervised learning DT | **['tree\_']** | **{'criterion': ['gini', 'entropy'], 'splitter': ['best', 'random'], 'max\_depth': [None, 10, 20, 30, 40, 50], 'min\_samples\_split': [2, 5, 10], 'min\_samples\_leaf': [1, 2, 4]}** | **from sklearn.neighbors import KNeighborsClassifier** |
| **KNN** | Supervised learning KNN | **n/a** | **{'n\_neighbors': [3, 5, 7, 9], 'weights': ['uniform', 'distance'], 'algorithm': ['auto', 'ball\_tree', 'kd\_tree', 'brute'], 'p': [1, 2]}** | **from sklearn.tree import DecisionTreeClassifier** |
| **SVM (RBF)** | Supervised learning SVM (RBF) | **['dual\_coef\_', 'support\_', 'n\_support\_']** | **{'C': [0.1, 1, 10, 100], 'gamma': ['scale', 'auto'], 'kernel': ['rbf']}** | **from sklearn.svm import SVC** |
| **NB** | Supervised learning NB | **['class\_count\_', 'feature\_count\_']** | **Smoothing parameters** | **from sklearn.naive\_bayes import GaussianNB**  **from sklearn.naive\_bayes import BernoulliNB** |

**b**) The 75-25 test split ratio,75% training and 25 % testing, is a common choice in machine learning and statistics due to its ability to provide sufficient training data, enhance robust model training, and provide an adequate test set. This ratio is commonly used and allows for comparisons between studies and experiments. It strikes a balance between having enough data for training and maintaining a significant portion for testing, making it particularly useful for larger training sets and robust evaluations.

**Task (5) – Evaluation: How good are your model.**

**a)**

A chart with different colored squares

Description automatically generated**A diagram of a confused matrix

Description automatically generated with medium confidenceA chart with different colored squares

Description automatically generatedA chart with blue and yellow squares

Description automatically generated**

**A chart with blue and yellow squares

Description automatically generatedA chart with blue and green squares

Description automatically generated**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metric**  **Name** | **“USE” or “DO NOT USE”** | **Justification in relation**  **to the success criteria** | **Model**  **Name** | **Metric Score** |
| Accuracy | **Do not use** | high positive and low negative value in Mpox class variable. | LR | **0.668** |
| DT | **0.587** |
| KNN | **0.668** |
| SVM(RBF) | **0.683** |
| NB | **0.669**  **0.670** |
| Recall | **Use** | Provided the maximum true positives | LR | **0.890** |
| DT | **0.659** |
| KNN | **0.829** |
| SVM(RBF) | **0.932** |
| NB | **0.897**  **0.891** |
| Precision | **Do not use** | Success rate was lower compared to other metrics | LR | **0.684** |
| DT | **0.681** |
| KNN | **0.702** |
| SVM(RBF) | **0.684** |
| NB | **0.682**  **0.685** |
| F-Measure | **Use** | Used in these kind of datasets | LR | **0.773** |
| DT | **0.670** |
| KNN | **0.761** |
| SVM(RBF) | **0.789** |
| NB | **0.775**  **0.775** |
| AUC-ROC | **use** | Provides a better visuals for the positives and negative for MPOX | LR | **0.675** |
| DT | **0.559** |
| KNN  SVM(RBF) | **0.648** |
| **0.678** |
| NB | **0.657**  **0.663** |

**c)** The Support Vector Machine (SVM) with an RBF kernel outperforms other models in key metrics, with high accuracy, precision, recall, and F1 score. It balances high recall with minimal false negatives, and its strong discriminatory power makes it the preferred choice for detecting individuals with MPOX in healthcare.

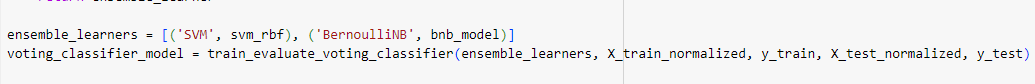
**d)** The SVM model with RBF kernel underwent fine-tuning using GridSearchCV and 10-fold cross-validation. The optimal hyperparameters were identified, resulting in improved recall, F1 score, and high accuracy. Hyperparameter tuning improved the model's generalization and performance, particularly in identifying positive cases, reducing false negatives and aligning with the project's goal.

A chart with blue squares and numbers

Description automatically generatedA screenshot of a computer code

Description automatically generated

**e)** We used ensemble learning to accurately detect individuals with Mpox using SVM, RBF Kernel, and Bernoulli Naive Bayes classifiers, based on their performance metrics and project success criteria.



**f)** The most effective model for MPOX detection was determined to be the Support Vector Machine (SVM) with an RBF kernel that was chosen. Its outstanding performance, especially a noteworthy recall of 93.27% and an overall F1 score of 78.97%, supports this decision. SVM's success was largely due to its capacity to identify complex patterns in large, complicated healthcare datasets. The RBF kernel-shaped decision boundaries of the algorithm worked well for identifying minute patterns that suggested the presence of MPOX. But it's important to recognise its limits, such as possible difficulties handling massive datasets and the computationally demanding fine-tuning procedure.

Concerns about privacy and potential biases are ethically raised when utilising this approach for MPOX screening, especially if it is implemented without thorough validation across a range of demographic groups. To guarantee responsible and equitable deployment in real-world applications, it is imperative to strike a compromise between model accuracy and ethical considerations. Upholding the model's dependability and addressing new ethical issues require constant review and improvements.

A screenshot of a computer

Description automatically generatedA diagram of a confused matrix

Description automatically generated with medium confidence