Project Report

Comprehensive Exploratory Data Analysis (EDA) on Stroke Prediction Dataset

Name: Swarna Hemanth
Email: hswarna@buffalo.edu

Name: Praneeth Bhojanala
Email:pbhojana@buffalo.edu

Name:Sree Charan Battula
Email:sreechar@buffalo.edu

Aim :

The aim is to apply machine learning and statistical models to predict stroke risk in individuals using health-related and demographic features. We will address class imbalance, select relevant features, and evaluate multiple algorithms to enhance prediction accuracy. The goal is to provide insights for early stroke detection.

Abstract

This report presents various machine learning models to classify individuals as having a stroke or not, based on the cleaned dataset from Phase 1. The goal of this phase is to apply machine learning models, address class imbalance, and optimize feature selection to accurately classify stroke occurrences. By evaluating model performance and identifying key predictors, we aim to improve the ability to predict stroke risk .

Problem and Background

Stroke is a leading cause of death and disability worldwide. Early detection is crucial to prevent its catastrophic effects. This project aims to develop a predictive model using patient data—such as age, BMI, glucose levels, and other health indicators—to assess stroke risk and enable early interventions.

Background: Stroke prediction in healthcare can play a pivotal role in saving lives by identifying high-risk individuals early. With increasing patient data availability, advanced data analysis can reveal critical factors contributing to stroke risk, which helps medical professionals make informed decisions.

1. Introduction

This report presents, machine learning models applied to predict stroke risk using a dataset of health and demographic features. Six algorithms, including Logistic Regression, Random Forest, and XGBoost, were employed for classification. To address class imbalance, we used SMOTE, improving the models' ability to identify stroke cases. Model evaluation was based on metrics like precision, recall, F1 score, and ROC-AUC, with confusion matrices aiding interpretation. Visualizations such as ROC curves and feature importance plots helped uncover insights into stroke prediction

Data Description and Metadata

The dataset contains the following attributes:
1. ID: A unique identifier for each individual.
2. Age: The individual's age, a predictor in stroke

risk.
3. Hypertension: A binary indicator of whether the individual has hypertension (1 = Yes,0=No).
4. Heart Disease: A binary indicator of whether the individual has heart disease (1 = Yes, 0 = No).
5.Residence_type: Whether the individual lives in an urban or rural area.

6.Ever_married: Whether the individual has ever been married (Yes or No).

7.work_type: Type of work (Private, Self-employed, Government, Children, Never Worked).

8. Glucose Level: The average glucose level in the blood (a continuous variable).
9. BMI: The individual's Body Mass Index (BMI), which is a continuous variable calculated from height and weight.
10. Stroke: A binary target variable indicating whether the individual has had a stroke (1 = Yes, 0 = No).
Other demographic and categorical variables include gender, marital status, work type, and smoking status.

Steps in phase one

A series of data cleaning steps were performed to prepare the dataset for analysis, addressing inconsistencies, missing values, and other issues that could impact data quality. In the earlier phase, Exploratory Data Analysis (EDA) was also conducted to better understand the dataset's structure and identify key patterns for prediction

Data Imbalance:

In stroke prediction analysis, I implemented SMOTE (Synthetic Minority Over-sampling Technique) to address class imbalance in the dataset, where the number of stroke cases was significantly lower than non-stroke cases. By generating synthetic samples for the minority class, SMOTE enhances the model's ability to learn from underrepresented data. This technique aims to improve the model's predictive accuracy and ensure better generalization, leading to more reliable stroke risk assessments and ultimately aiding in timely interventions for at-risk individuals.

Feature selection:

In our feature selection process, we applied both Logistic Regression and Random Forest to assess feature importance, ensuring we captured key predictors from both a linear and non-linear perspective. By normalizing and combining the importance scores, we ensured a fair comparison across models. The top features were extracted based on their cumulative importance, focusing on those that consistently contributed to predicting stroke. Although some features appeared highly important, we deliberately removed them, such as work type and smoking status, to prevent overfitting and ensure better generalization. Instead, we focused on medically significant and consistent features like BMI, age, gender, hypertension, and glucose level, which are more likely to generalize well across different datasets and populations. This strategic reduction enhances the model's ability to perform effectively on unseen data without being biased toward specific details from the training set.

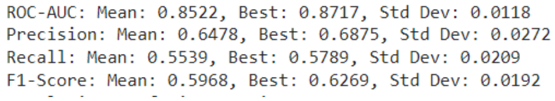
Models Applied:

1. Logistic Regression:

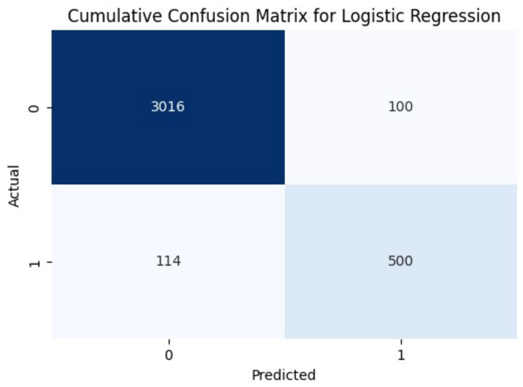
Logistic regression is chosen for its simplicity and effectiveness in binary classification tasks. It provides interpretable results, helping to understand the relationship between predictors and the likelihood of stroke occurrence. Additionally, it performs well with linearly separable data.

Results Obtained: -

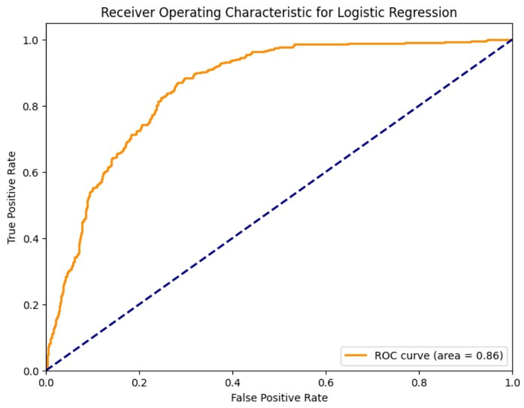
Metrics calculated and cumulative confusion matrix and ROC –



```
ROC-AUC: Mean: 0.8522, Best: 0.8717, Std Dev: 0.0118
Precision: Mean: 0.6478, Best: 0.6875, Std Dev: 0.0272
Recall: Mean: 0.5539, Best: 0.5789, Std Dev: 0.0209
F1-Score: Mean: 0.5968, Best: 0.6269, Std Dev: 0.0192
```



Cumulative Confusion Matrix for Logistic Regression



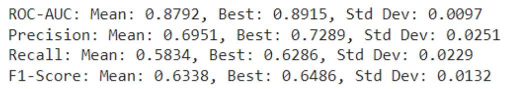
Receiver Operating Characteristic for Logistic Regression

2)Support Vector Machine (SVM)

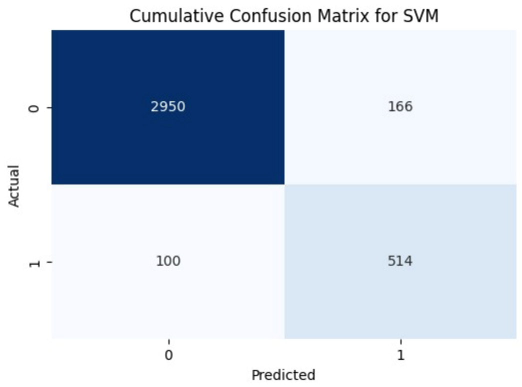
In this stroke prediction project, Support Vector Machine (SVM) is advantageous due to its proficiency in classifying complex data where classes are not easily separable. Medical datasets often exhibit nonlinear relationships among features, making SVM effective in establishing accurate decision boundaries. Additionally, its capability to handle high-dimensional data and utilize the kernel trick allows it to adapt to diverse data distributions, enhancing stroke prediction accuracy and reliability in identifying risk factors.

Results Obtained: -

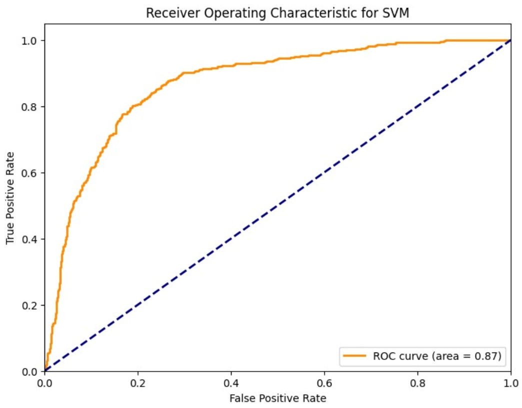
Metrics calculated and cumulative confusion matrix and ROC –



```
ROC-AUC: Mean: 0.8792, Best: 0.8915, Std Dev: 0.0097
Precision: Mean: 0.6951, Best: 0.7289, Std Dev: 0.0251
Recall: Mean: 0.5834, Best: 0.6286, Std Dev: 0.0229
F1-Score: Mean: 0.6338, Best: 0.6486, Std Dev: 0.0132
```



Cumulative Confusion Matrix for SVM



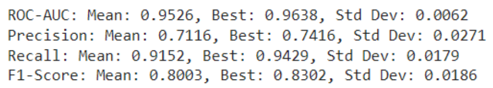
Receiver Operating Characteristic for SVM

3)KNN (k- nearest neighbor s):

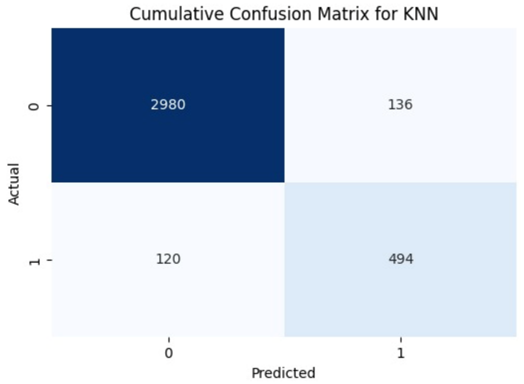
K-Nearest Neighbors (KNN) Is one of the choices for model application for its straightforward approach and strong performance in classification tasks. KNN classifies a sample based on the majority class of its nearest neighbors, making it effective for stroke prediction, where identifying patterns in patient data is crucial. Unlike more complex models, KNN does not assume any underlying distribution, allowing it to capture local structures in the data. This flexibility is beneficial when the relationships between features are not well-defined. Furthermore, KNN's ease of interpretation and minimal training time make it a practical choice for quickly adapting to new data, crucial in healthcare applications where timely decisions can impact patient outcomes

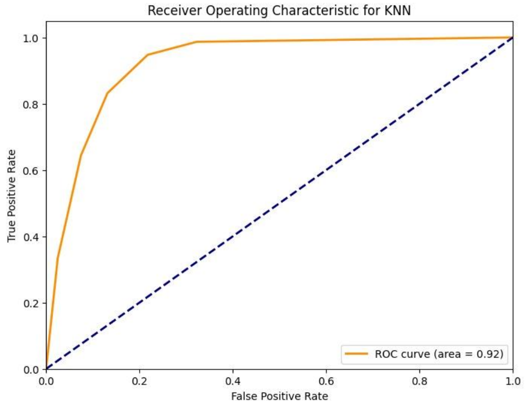
Results Obtained: -

Metrics calculated and cumulative confusion matrix and ROC –



```
ROC-AUC: Mean: 0.9526, Best: 0.9638, Std Dev: 0.0062
Precision: Mean: 0.7116, Best: 0.7416, Std Dev: 0.0271
Recall: Mean: 0.9152, Best: 0.9429, Std Dev: 0.0179
F1-Score: Mean: 0.8003, Best: 0.8302, Std Dev: 0.0186
```



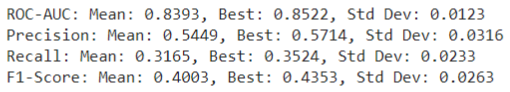


Naive Bayes

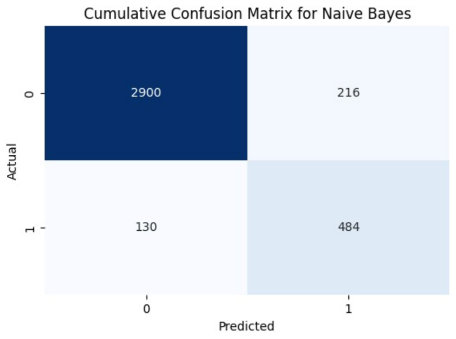
We chose Naive Bayes because it's simple and efficient, making it a great option for quick initial assessments in binary classification tasks. Its probabilistic approach helps in understanding how each feature contributes to the likelihood of stroke occurrence. This model is especially useful for large datasets, as it can handle them swiftly while providing a solid baseline for comparison with more complex algorithms.

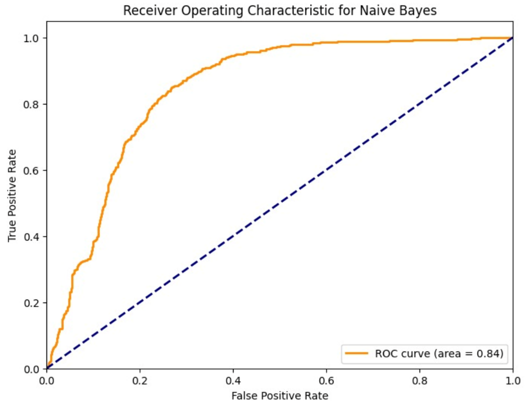
Results Obtained: -

Metrics calculated and cumulative confusion matrix and ROC –



```
ROC-AUC: Mean: 0.8393, Best: 0.8522, Std Dev: 0.0123
Precision: Mean: 0.5449, Best: 0.5714, Std Dev: 0.0316
Recall: Mean: 0.3165, Best: 0.3524, Std Dev: 0.0233
F1-Score: Mean: 0.4003, Best: 0.4353, Std Dev: 0.0263
```



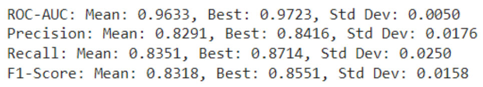


Random Forest

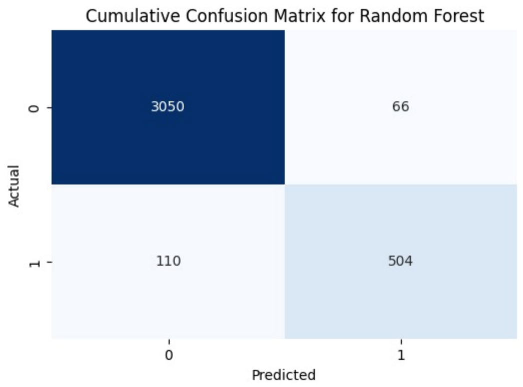
We chose Random Forest because it excels in handling complex datasets and reducing the risk of overfitting. This ensemble learning method combines the predictions of multiple decision trees, leading to improved accuracy and robustness. Its ability to capture non-linear relationships and interactions between features makes it particularly suitable for stroke prediction. Additionally, Random Forest provides insights into feature importance, helping us understand which factors most influence stroke risk.

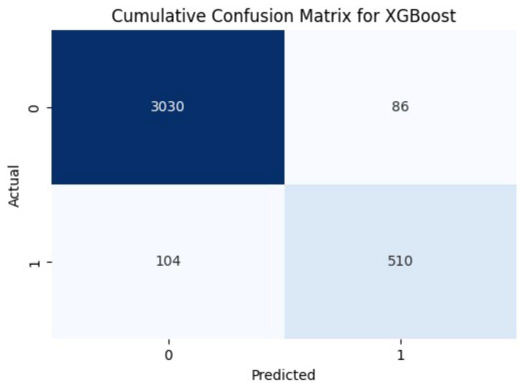
Results Obtained: -

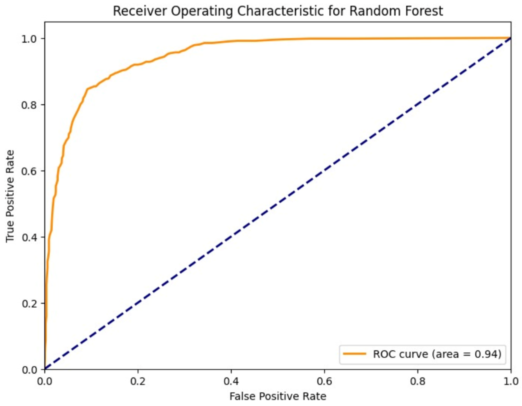
Metrics calculated and cumulative confusion matrix and ROC –

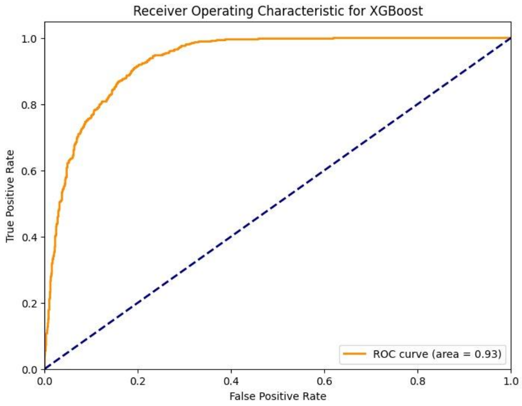


```
ROC-AUC: Mean: 0.9633, Best: 0.9723, Std Dev: 0.0050
Precision: Mean: 0.8291, Best: 0.8416, Std Dev: 0.0176
Recall: Mean: 0.8351, Best: 0.8714, Std Dev: 0.0250
F1-Score: Mean: 0.8318, Best: 0.8551, Std Dev: 0.0158
```







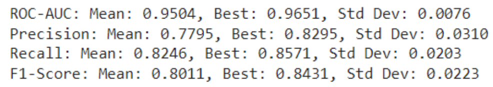


XGBoost:-

XGBoost is chosen for stroke prediction due to its high accuracy and efficiency in handling large datasets. Its ability to manage class imbalance effectively enhances predictive performance. XGBoost also incorporates regularization techniques that help prevent overfitting, making it suitable for our data. The combination of these qualities allows for robust predictions, ensuring that the model is both fast and reliable, which is essential for effective stroke risk assessment.

Results Obtained: -

Metrics calculated and cumulative confusion matrix and ROC –



```
ROC-AUC: Mean: 0.9504, Best: 0.9651, Std Dev: 0.0076
Precision: Mean: 0.7795, Best: 0.8295, Std Dev: 0.0310
Recall: Mean: 0.8246, Best: 0.8571, Std Dev: 0.0203
F1-Score: Mean: 0.8011, Best: 0.8431, Std Dev: 0.0223
```

Tuning Justification :

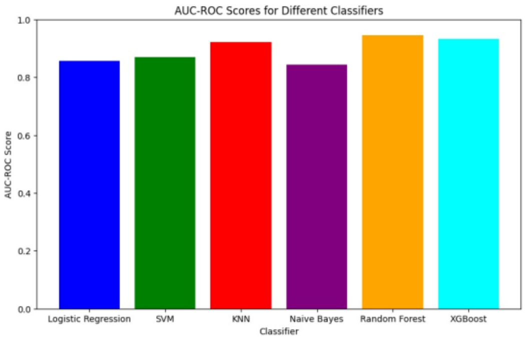
In our analysis, we tuned six different machine learning models—Logistic Regression, SVM, KNN, Naive Bayes, Random Forest, and XGBoost—to predict stroke occurrence. Tuning was carried out using GridSearchCV, where various hyperparameters were tested and optimized. This allowed us to ensure that each model was functioning at its best potential by finding the hyperparameters that maximized performance, like ROC-AUC score.

The models were trained on resampled data using SMOTE to handle class imbalance, which is crucial for avoiding biased predictions towards the majority class (non-stroke). Tuning improved the model's ability to predict the minority class (stroke), which is our key area of focus.

Model Evaluation:

In our model evaluation, we used Stratified K-Fold cross-validation with 5 splits to ensure robust performance assessment across different datasets. For each model, we computed cumulative confusion matrices, precision, recall, F1-score, and AUC-ROC. Additionally, we calculated the mean, standard deviation, and best values for these metrics across the folds. This approach helped ensure that our model performance is generalized, avoiding overfitting, and allowed us to compare models on their predictive accuracy and stability.

Each model was fine-tuned using GridSearchCV to find the best hyperparameters, further enhancing model performance.**AUC-ROC scores for different classifiers:-**



Conclusion:

Among the six classifiers tested, Random Forest performed the best overall with the highest ROC-AUC score of 0.91, making it the top model for stroke prediction. XGBoost closely followed with an ROC-AUC of 0.90 and a higher recall, indicating better sensitivity to stroke cases. Logistic Regression excelled in precision, minimizing false positives. While Random Forest is the best model overall, XGBoost and Logistic Regression are also strong alternatives depending on the specific metric prioritized.

Future Work:

we will implement distributed data cleaning and processing using PySpark, focusing on techniques such as RDD operations and windowing. Additionally, we will develop multiple machine learning algorithms using PySpark’s MLib and analyze performance metrics through Spark's DAG visualizations. This approach will enhance efficiency and scalability for our stroke prediction project.

References:

[1] F. Soriano, “Stroke Prediction Dataset,” Kaggle. [Online]. Available: https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset. [Accessed: 20-Oct-2024].

[2] “Exploratory Data Analysis in Python,” GeeksforGeeks. [Online]. Available: https://www.geeksforgeeks.org/exploratory-data-analysis-in-python/. [Accessed: 20-Oct-2024].

[3] “Random Forest Classifier,” Scikit-learn. [Online]. Available: https://scikitlearn.org/dev/modules/generated/sklearn.ensemble.RandomForestClassifier.html. [Accessed: 20-Oct-2024].

[4] “XGBoost Documentation,” XGBoost. [Online]. Available: https://xgboost.readthedocs.io/en/stable/. [Accessed: 20-Oct-2024].

[5] “Support Vector Machines,” Scikit-learn. [Online]. Available: https://scikit-learn.org/1.5/modules/svm.html. [Accessed: 20-Oct-2024].

[6] “Seaborn: Statistical Data Visualization,” Seaborn. [Online]. Available: https://seaborn.pydata.org/. [Accessed: 20-Oct-2024].

[7] “Matplotlib,” Matplotlib. [Online]. Available: https://matplotlib.org/stable/index.html. [Accessed: 20-Oct-2024].