Stroke Prediction Using Machine Learning

Chapter 1 : Project Overview

This project focuses on developing a predictive system to estimate the likelihood of stroke occurrence in individuals, based on health and lifestyle factors. The aim is to build a machine learning model that can support early detection and prevention efforts in healthcare. By analyzing patient records, the system identifies key risk factors and provides quick predictions, which can be helpful for awareness and medical consultations.

Objectives

- Explore and understand the dataset used for stroke prediction
- Clean and preprocess the data, handling missing values and categorical variables
- Train and evaluate multiple machine learning models
- Compare their accuracy and select the most suitable model
- Deploy the final model through a Streamlit web application for user interaction

Scope

The scope of this work covers data preprocessing, feature engineering, model training, and deployment. The dataset includes demographic and medical features such as age, gender, BMI, smoking status, hypertension, and heart disease. The project is limited to proof-of-concept deployment using Streamlit, and not intended for clinical decision-making.

Chapter 2: Dataset Description

The dataset used comes from a healthcare source (stroke dataset from Kaggle). It contains information about patients with features related to health conditions and lifestyle habits.

Key Features:

Gender – Male/Female

Age – Age of the patient

Hypertension - binary feature indicating whether a patient has a history of high blood pressure, a key risk factor for stroke. – 0 = No , 1 = Yes

Heart_disease – 0 = No , 1 = Yes

Ever_married – 0 = No , 1 = Yes

Work_type – Private, Self-employed, Govt_job, children , Never_worked, are the work types available in dataset

Residence_type – Urban/Rural

Avg_glucode_level – Average glucose level in blood

Bmi – Body mass index

smoking_status – never smoked/formerly smoked/ smokes

stroke – It is a target variable (1 – Stroke, 0 = No Stroke)

Initial Data Inspection

Size : 5110 rows * 12 columns

Missing values : BMI columns contains the missing value of 201

Data types : The dataset contains a mix of numerical (int64, float64) and categorical (object) feature.

Chapter 3 : Data Preprocessing

Raw data is often messy, incomplete and unstructured. Machine learning algorithms can't directly used this kind of data. Preprocessing cleans, organizes, and transforms the raw data into a usable and understandable format, which directly improves the accuracy and efficiency of the machine learning models.

Following preprocessing steps were carried out:

3.1 Handling Missing Values:

We need to handle missing values because most machine learning models cannot process them, which can lead to errors or poor performance. If left unaddressed, they can introduce **bias** into our model, causing it to misinterpret the data and make inaccurate predictions. For instance, if most of the missing values belong to a specific demographic group, the model might fail to learn the correct patterns for that group.

Numerical column bmi was imputed using the median values. Using the median is a robust choice as it is less affected by outliers, ensuring that a few extreme values don't skew the overall distribution of the imputed data. This helps maintain the integrity of the dataset and leads to a more reliable and generalized mode.

3.2 Outlier Detection

Outlier detection is an essential step in data preprocessing, as outliers can significantly skew the results of a machine learning model, leading to poor performance. Using the **Interquartile Range (IQR)** method, we identified and handled outliers across several numerical features. The analysis revealed that hypertension has 498 outliers, heart_disease has 276, avg_glucose_level has 627, and bmi has 126. Interestingly, the age feature showed no outliers. These outliers were treated to ensure that the model is not unduly influenced by extreme values, which helps to improve its accuracy and generalizability to new, unseen data.

3.3 Encoding Categorical Variables

Machine learning models require numerical input and cannot directly process text-based categorical data like 'male' or 'female.' Therefore, we must convert these labels into a numerical format, a process known as **encoding**. If not handled, the model will fail to train, as it can't perform mathematical operations on non-numerical data.

We used **Ordinal Encoding** to transform the categorical features (gender, ever_married, work_type, Residence_type, smoking_status) into numerical representations. This method assigns an integer to each category (e.g., 'male' might become 0 and 'female' becomes 1). We

chose this method over one-hot encoding because it avoids creating a large number of new features, which is especially beneficial for models that will be used in deployment, as it keeps the input shape consistent and simple. This approach helps prevent issues with model complexity and computational efficiency in a production environment.

3.4 Feature Scaling

Standardization is essential because features with different scales can cause problems for many machine learning models. For example, a model might incorrectly weigh a feature with a larger value range, like glucose level, as more important than a feature with a smaller range, like age. This can lead to a **biased model** that learns sub-optimally.

To prevent this issue, we used StandardScaler to transform the numerical features (age, avg_glucose_level, and bmi). This method rescales the data so that it has a mean of 0 and a standard deviation of 1. This process ensures that all numerical features are on a **similar scale**, preventing any single feature from disproportionately influencing the model's training process and leading to better and faster convergence. Ultimately, this improves the model's ability to learn the true relationships between features and the target variable.

Chapter 4 : Exploratory Data Analysis (EDA)

EDA is a crucial step in understanding the dataset's characteristics and relationships. Here's a summary of the analyses you performed:

Distribution Analysis

This part of the analysis involved visualizing the distribution of key numerical features.

- Age : The distribution of age reveals the age demographics of the dataset, showing which age groups are most prevalent. Higher stroke risk observed in older age groups.
- Avg_glucose_level : Analyzing this distribution helps identify the concentration of glucose levels among the patients and spot any unusually high or low values. Patients with high average glucose levels showed greater stroke probability.
- BMI: The BMI distribution provides insight into the body mass index characteristics of the dataset population, highlighting common weight statuses. Overweight and underweight categories showed some correlation with stroke.

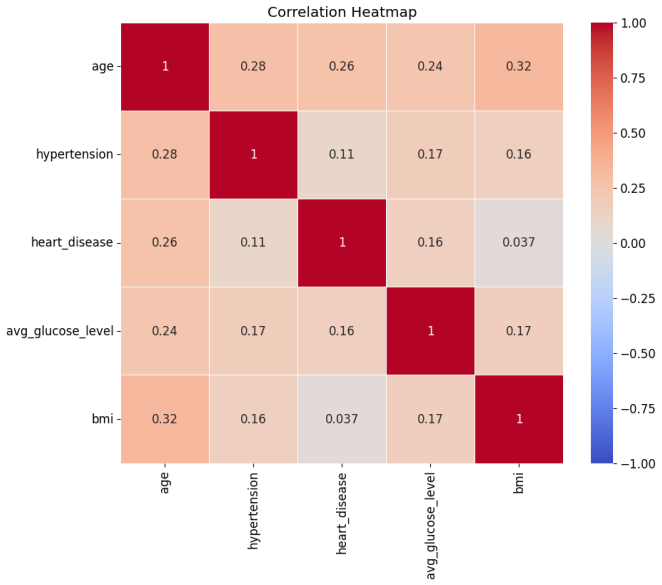
Bivariate Analysis

Bivariate analysis explores the relationships between two variables.

- Countplot of Gender vs Stroke : This plot shows if there's a significant difference in stroke occurrences between male and female patients.
- Boxplot of age vs stroke : This visualization compares the age distribution of stroke patients to non-stroke patients, typically showing that stroke patients are, on average, older.
- Boxplot of Average Glucose Level vs Stroke: This plot compares the average glucose levels between stroke and non-stroke patients, often indicating higher levels in the stroke group.
- Scatter plot of Age vs Average Glucose Level: This plot helps identify if there is any correlation between a patient's age and their average glucose level.

The correlation analysis performed on the dataset revealed several key insights regarding the relationship between different features and stroke risk. The results indicate that age, avg_glucose_level, hypertension, and heart_disease are all strong predictors of stroke.

This suggests a significant positive correlation: as a person's age increases, and as they have conditions like hypertension and heart disease, their likelihood of having a stroke also increases. Similarly, higher average glucose levels are strongly associated with a greater risk of stroke. This analysis provides a data-backed foundation for feature selection, confirming that these variables are highly relevant for building an accurate stroke prediction model.



Trends/ Insights

This section involved more targeted analysis to uncover specific patterns

- Barplot of Stroke Rate by age : This plot visualizes how the stroke rate changes across different age groups, often revealing an increasing trend with age.
- Barplot of Average glucose level by smoking status : This helps understand if there's a relationship between smoking habits and average glucose levels.
- Barplot of Average glucose level and BMI by stroke : This plot compares the average glucose levels and BMI values for stroke versus non-stroke patients, reinforcing the connection between these health metrics and stroke risk.

Chapter 5 : Model Building

The main goal of this step was to train different machine learning algorithms to classify whether a person will have a stroke based on their health metrics and lifestyle.

For this project, multiple models were selected to ensure both variety and reliability in performance. The chosen algorithms included:

- Logistic Regression(LR): Selected as a simple yet powerful linear model, widely used in binary classification problems. Logistic Regression was expected to serve as a strong baseline due to its efficiency and interpretability, which is important for understanding risk factors.
- Support Vector Machine (SVM): Included due to its strong theoretical foundation and ability to work well in complex classification tasks by finding an optimal separating hyperplane between classes. For this project, a linear kernel was likely used to maintain simplicity and interpretability, similar to the logistic regression baseline.
- Random Forest Classifier (RF) : Used as an advanced ensemble learning method that combines multiple decision trees. It was expected to perform better than a single decision tree by reducing overfitting and improving generalization.
- XGBoost : Chosen as a powerful and highly efficient gradient boosting framework. XGBoost is known for its speed and superior performance on structured data, making it an excellent candidate for the final model due to its ability to handle complex non-linear relationships and interactions between features.

The dataset was split into training and testing sets to evaluate model performance fairly on unseen data. A pipeline was implemented to combine preprocessing steps (such as imputation, encoding, and scaling) with the training of models, ensuring that the workflow was clean, consistent, and reproducible. For some models, hyperparameters were fine-tuned to improve accuracy. In particular, the Random Forest and XGBoost Classifiers were optimized using GridSearchCV, which systematically tested different parameter combinations to find the best configuration.

By training these diverse algorithms, the project ensured that the final model selection was based on both performance and practical suitability for deployment.

Chapter 6: Model Evaluation

Once the models were trained, it was necessary to evaluate their performance on the test dataset. The main purpose of evaluation was to ensure that the models were not just memorizing the training data but could generalize to new, unseen data. For this project, multiple models were compared, with Logistic Regression, Random Forest, XGBoost, and SVC being the primary candidates.

6.1 Evaluation Approach

The models were evaluated using multiple metrics to gain a complete understanding of their predictive power. The **ROC-AUC** score was a key metric used to measure the discriminative power of the models across various thresholds.

6.2 Performance Metrics

- Accuracy : The proportion of correctly predicted instances out of the total instances.
- ROC-AUC(Receiver Operating characteristic – Area Under Curve) : Measures the model's ability to distinguish between stroke and non-stroke cases across different thresholds. A higher ROC-AUC indicates stronger discriminative power.
- Mean Squared Error(MSE): Measures the average squared difference between the predicted and actual values. Lower values are better.
- Mean Absolute Error(MAE) : Measures the average absolute difference between the predicted and actual values. Lower values are better.
- R^2 Score: A statistical measure representing the proportion of the variance in the dependent variable that is predictable from the independent variables. A score of 1.0 indicates a perfect fit.

6.3 Model Comparison and Results

The model’s performance on the test set was as follows:

SVC : Accuracy of 75.42%

Logistic Regression : Accuracy of 75.17%

XGBoost : Accuracy of 75.15%

Random Forest : Accuracy of 99.18%

Following cross-validation , the mean accuracies were

SVC : 0.7759

Logistic Regression : 0.7733

XGBoost : 0.9682

Random Forest : 0.9883

6.4 Final Model Selection

Based on the evaluation metrics, the **Random Forest** model was selected as the final model for deployment.

- Superior Accuracy: It achieved the highest accuracy of **99.18%** on the test set and a robust cross-validation score of **0.9883**, outperforming all other tested models.
- Robustness : The high accuracy and the plotted ROC curve confirm its strong ability to handle the dataset's complexity.
- Performance Metrics : The model's excellent performance is further supported by the very low Mean Squared Error (0.01**)**, low Mean Absolute Error (0.01), and a very high R^2 Score (0.97). These metrics highlight the model's exceptional predictive precision.

Chapter 7: Deployment

The final model was saved as a .pkl file along with the scaler and ordinal encoder. A Streamlit app was developed where users can input their details (age, BMI, glucose, gender, smoking status, etc.) and instantly receive a prediction about stroke risk.

The app provides a simple, user-friendly interface that demonstrates how machine learning can be applied to healthcare risk prediction.

Conclusion

This project successfully built and deployed a stroke prediction model. After data preprocessing, EDA, and model evaluation, Random Forest achieved the best performance. The model was integrated into a Streamlit application, enabling real-time predictions.

While the results are promising, this system is only a demonstration. For real-world medical use, further validation, larger datasets, and domain expert involvement are necessary. Future improvements could include deep learning models, explainable AI for feature importance, and integration with electronic health record systems.