Exploring Feature Relationships in Brain Stroke Data Using Polynomial Feature Transformation and Linear Regression Modeling

[1]Sitanaboina S L Parvathi, [2]Aruna Devi B, [3]Gururaj L Kulkarni, [4]Sangeetha Murugan, [5]Bindu Kolappa Pillai Vijayammal and [6]Neha

[1]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India.

[2]Department of Electronics and Communication Engineering, Dr. N.G.P Institute of Technology, Coimbatore, Tamil Nadu, India.

[3]Department of Computer Science and Engineering, Vardhaman College of Engineering, Hyderabad, Telangana, India.

[4]Department of Computer Science and Engineering, Madanapalle Institute of Technology and Science, Madanapalle, Andhra Pradesh, India.

[5]Department of Science and Humanities (General Engineering Division), R.M.K College of Engineering and Technology, Puduvoyal, Tamil Nadu, India.

[6]Department of Computer Science and Engineering, Chandigarh University, Mohali, Punjab, India.

[1]s.srilakshmiparvathi@gmail.com, [2]arunadevi@drngpit.ac.in, [3]gururajlkulkarni@gmail.com, [4]sangee525@gmail.com, [5]bindu@rmkcet.ac.in, [6]neha.arya35@gmail.com

Correspondence should be addressed to Sitanaboina S L Parvathi : s.srilakshmiparvathi@gmail.com

Article Info
Journal of Machine and Computing (http://anapub.co.ke/journals/jmc/jmc.html)
Doi : https://doi.org/10.53759/7669/jmc202404107
Received 18 February 2024; Revised from 28 May 2024; Accepted 28 August 2024.
Available online 05 October 2024.

Abstract – A Cerebral vascular accident, commonly known as a stroke, is a pathological condition that impacts the brain due to the rupture of capillaries. It occurs when there is a disturbance in the typical blood circulation and essential physiological processes of the brain. Stroke prediction plays a crucial role in early diagnosis and intervention, potentially improving patient outcomes. This paper proposes a machine learning model that leverages polynomial feature transformation and linear regression modeling for stroke prediction. The model addresses the challenge of capturing non-linear relationships between features and the target variable while maintaining interpretability. The proposed approach involves preprocessing data by separating categorical and numerical features, applying one-hot encoding to categorical features, and generating polynomial features up to the second degree for numerical features. This tailored preprocessing is facilitated by a Column Transformer. For model development, a machine learning pipeline is constructed, splitting the data into training and testing sets. Despite utilizing polynomial features, linear regression is employed as the final model, allowing for the capture of both linear and non-linear relationships while maintaining interpretability. This work contributes to stroke prediction by offering a balanced approach that considers model complexity and interpretability, showcasing the potential of linear regression with polynomial features for accurate predictions and insights into feature-target relationships. The proposed model exhibited superior performance compared to other existing models, achieving a remarkable testing accuracy of 99.2%.

Keywords - Stroke Prediction, Machine Learning, Polynomial Features, Linear Regression, One-Hot Encoding.

I. INTRODUCTION

Stroke is a significant cause of mortality among adults worldwide, affecting approximately 6.2 million individuals annually. A stroke is a health issue characterized by diminished blood flow to the brain, leading to the death of cells. It manifests primarily in two forms: ischemic, resulting from inadequate blood supply, and hemorrhagic, stemming from bleeding. Both types disrupt normal brain function, presenting symptoms such as one-sided paralysis or numbness, difficulty with speech or comprehension, dizziness, and partial vision loss. These symptoms typically manifest shortly after the stroke occurs [1, 2]. Stroke, medically referred to as a cerebrovascular accident, is a neurological condition arising from either a lack of blood supply (ischemia) or bleeding within the brain arteries (hemorrhage). It typically induces a variety of motor and cognitive deficits, which can vary greatly, ultimately impeding functionality [3]. According to the World Health

Organization (WHO), stroke is the leading cause of death and a significant public health issue. While there has been extensive research on the outlook for heart attacks, the exploration of risk factors related to strokes has been comparatively restricted. In light of this, numerous advanced machine learning models have been utilized to forecast the likelihood of an upcoming stroke event. Many individuals who survive strokes experience different neurological impairments, leading to varying degrees of reduced quality of life, which poses a substantial burden on patients, caregivers, and society. Improving the accuracy of predicting functional outcomes following a stroke could assist clinicians in devising suitable long-term treatment strategies. This might involve developing plans that incorporate a better understanding of expected recovery levels, along with tailored rehabilitative interventions, taking into account patients' living situations, to facilitate shared decision-making with patients and their families [4]. Considerable attention has been directed towards identifying factors that can predict the functional recovery following a stroke. Selecting the optimal features is crucial for achieving higher accuracy in disease prediction. [5]

Nevertheless, the challenge lies in accurately categorizing stroke symptoms, which complicates the diagnosis of conditions resulting from stroke and associated neurological impairments. Consequently, there is an urgent demand for technology capable of monitoring individuals at risk of stroke, facilitating their medical visits, and enabling timely diagnosis and treatment by healthcare professionals. Ongoing research aims to pinpoint key risk indicators for stroke by assessing the initial disabilities experienced by stroke and lung patients and continuously monitoring their health status [6, 7]. Machine learning, a subset of data science, empowers computers to learn and enhance their performance autonomously, without direct programming. Its core objective lies in crafting system programs capable of analysing data and forecasting future outcomes. This involves extracting insights from vast datasets and transforming raw data into actionable information, a process commonly referred to as data mining [8, 9]. Machine learning has effectively been employed to anticipate the positive prognosis after an acute stroke related to other chronic diseases within a three-month timeframe [10-14]. Machine learning, along with deep learning models, is very useful in feature extraction [15, 16]. Machine learning models are currently being employed to forecast the likelihood of stroke recurrence over the long term [17-20]. To accurately detect a stroke, various algorithms such as Naive Bayes, J48, K-nearest neighbour, fuzzy C-mean and random forest are employed [21, 22].

In order to classify levels of stroke risk, various machine learning models including logistic regression, naive Bayes, Bayesian network, decision tree, neural network, random forest, bagged decision tree, and ensemble methods such as voting and boosting with decision trees were utilized support vector machine (SVM) and ensemble (bagged) [23, 24]. The classification accuracy of the neural network (NN), decision tree (DT), Artificial NN and random forest (RF) [25]. The novelty of this paper is that it tries to achieve a balance between capturing non-linear relationships and model interpretability for stroke prediction.

Using Polynomial Features with Linear Regression

Although polynomial regression is more appropriate for capturing non-linear relationships, its interpretation can be challenging due to its complexity. This model addresses this by using a linear regression model even after generating polynomial features. This allows the model to capture non-linearity through the features themselves while maintaining the interpretability of a linear model.

Focus on Interpretability

The text emphasizes the importance of interpretability throughout the explanation. This suggests that the proposed model prioritizes not just accuracy but also the ability to understand how the features influence the prediction of stroke.

The paper is structured as follows: Section II outlines the organization, followed by Section III which presents the Literature Review. Section IV covers the Dataset and Preprocessing methods, while Section V elaborates on the Proposed Methodology. Section VI is dedicated to Results and Analysis, with Section VII concluding the paper and discussing avenues for future work.

II. LITERATURE REVIEW

In [23] introduced the Stroke Predictor (SPR), a machine learning model that uses Improvised random forest (IRF) designed to predict the likelihood of stroke occurrence. This model achieved an impressive prediction accuracy of 96.97%. Notably, while this work did not prioritize feature selection, it nonetheless produced robust results. In [22] used machine learning algorithms that includes DT, RF, K-Nearest Neighbors (KNN), Support Vector Machine (SVM) to analyse brain stroke data, achieving an accuracy of approximately 82%. However, a notable limitation of this study is its reliance on textual data, which may not always provide accurate predictions for stroke occurrences. The[18] developed Stacking method that includes Naive Bayes (NB), RF, Logistic Regression (LR), K-Nearest Neighbors (KNN) and Multilayer Perceptron (MP) exhibits a commendable accuracy of 98%. However, a notable drawback lies in the lack of interpretability and feature selection. These aspects are crucial as they aid medical practitioners and decision-makers in understanding the model's predictions and identifying the key factors contributing to stroke occurrence. Islam MM et al. [30] employed a machine learning model to predict the likelihood of stroke occurrence in patients, leveraging RF classifier, which achieved an accuracy of 96%. However, a notable drawback of this model is the lack of explicit discussion regarding the interpretability of the Random Forest model's predictions, particularly concerning its applicability in medical decision-making contexts.

The [23] presented a machine learning classifier based on soft voting, utilizing RF, Extremely Randomized Trees, and Histogram-Based Gradient Boosting (HBGB), achieving an impressive accuracy of 96.88%. However, a potential limitation of this model lies in the lack of exploration into feature importance. The [21] utilized a machine learning model that used SVM and RF Regression with an accuracy score of 79%. However, there is a notable absence of discussion or analysis regarding the generalizability of the developed models. Shobayo et al. [23] employed machine learning algorithms that used RF Algorithm and achieved an accuracy of 91.43%. Despite emphasizing interpretability, robustness, and generalization as pivotal aspects for deploying algorithms in medical contexts, the study lacks extensive discourse on the generalizability of the developed model beyond the dataset utilized in the research.

III. PROPOSED METHODOLOGY

Dataset Description

The dataset comprises information on 4981 patients available at the source (https://www.kaggle.com/datasets/jillanisofttech/brain-stroke-dataset?resource=download), encompassing various attributes:

1. Gender: Designated as "Male," "Female," or "Other."
2. Age: Denoting the age of the patient.
3. Hypertension: Assigned a value of 0 if the patient lacks hypertension, and 1 if hypertension is present.
4. Heart Disease: Indicated as 0 if the patient has no heart diseases, and 1 if a heart disease is diagnosed.
5. Ever-Married: Recorded as either "No" or "Yes."
6. Work Type: Categorized into "Children," "Govt Job," "Never Worked," "Private," or "Self-Employed."
7. Residence Type: Classified as "Rural" or "Urban."
8. Average Glucose Level: Represents the average glucose level in the patient's blood.
9. BMI: Denotes the body mass index of the patient.
10. Smoking Status: Identified as "Formerly Smoked," "Never Smoked," "Smokes," or "Unknown."
11. Stroke: Assigned a value of 1 if the patient experienced a stroke, and 0 if not.

Data Preprocessing

The preprocessing step plays a pivotal role in preparing the data for subsequent modelling by applying various transformations defined in the preprocessor. This includes two primary tasks: polynomial feature generation for numerical features and one-hot encoding for categorical features. Polynomial feature generation involves expanding the feature space by creating polynomial combinations of the original numerical features, facilitating the capture of non-linear relationships between predictors and the target variable. On the other hand, one-hot encoding transforms categorical variables into a binary format, enabling the model to effectively interpret and utilize categorical data for predictive tasks. These preprocessing transformations are fundamental in shaping the dataset into a suitable format for the subsequent modelling phase, ensuring that the data is appropriately structured to extract meaningful insights and facilitate accurate predictions.

IV. PROPOSED METHODOLOGY

Algorithm for One-Hot Encoding

1. **Input**:
   - Categorical features: ["gender", "ever_married", "work_type", "Residence_type", "smoking_status"]
2. **Output**:
   - One-Hot Encoded Features: Binary vectors representing each category for all categorical features.
3. **Initialization**:
   - Create an empty matrix OneHot with dimensions
     Number of Samples *Total Categories
4. **Process**:
   - For each sample in the dataset:
     - For each categorical feature:
     - Determine the category to which the sample belongs.
     - Set the corresponding element in the one-hot encoded vector for the sample to 1, indicating the presence of the category, and all other elements to 0.
5. **Output**:
   - OneHot: Matrix containing the one-hot encoded features for all categorical features in the dataset.

This algorithm will transform all categorical features in the dataset into a set of binary vectors, where each vector represents a sample's category membership

Polynomial Feature Transformation

The polynomial feature transformation is a preprocessing technique used to generate polynomial features from the original numerical features. It involves creating new features by raising the original features to various powers up to a specified degree n and taking combinations of these powers. Let's delve into the mathematical model of polynomial feature transformation for a single numerical feature x up to the n^{th} degree. Suppose we have a single numerical feature x. The

polynomial feature transformation generates new features by raising x to powers from 1 to n, and then taking combinations of these powers. The resulting polynomial features include terms like $x^1,x^2,x^3,\ldots, x^n$ as well as all possible combinations of these terms.

The mathematical model of polynomial feature transformation for x up to the n^{th} degree can be represented as follows:

Generating Polynomial Features for a Single Feature x

$$\text{Polynomial Features(x)} = \sum_{k=0}^{n} \binom{n}{k}\ 1, x, \ldots, x^n \tag{1}$$

Combinations of Polynomial Features

After generating the polynomial features for x, we take combinations of these features to form the final feature set. This involves considering all possible combinations of the generated polynomial features, including interactions between different features. The number of combinations grows exponentially with the degree n, resulting in a larger feature set that captures complex relationships between the original feature x and the target variable.

Mathematically, the total number of polynomial features generated for a single feature x up to the n^{th} degree, including combinations, can be calculated using the formula for combinations:

$$\text{Total Polynomial Features } = \sum_{k=0}^{n} \binom{n}{k} = 2^n \tag{2}$$

where $\binom{n}{k}$ represents the binomial coefficient, which gives the number of combinations of choosing k elements from a set of n elements. This process captures non-linear relationships between the original feature and the target variable, making it suitable for modeling complex data patterns in machine learning tasks.

ColumnTransformer

The Column Transformer is a preprocessing tool in scikit-learn that allows different transformations to be applied to different columns (or subsets of columns) in a dataset. It's particularly useful when dealing with datasets that contain both numerical and categorical features, as it enables us to handle each type of feature appropriately.

Input Data

Suppose we have a dataset D with m samples and n features. Each sample i is represented by a vector $xi=[x_{i1},x_{i2},\ldots,x_{in}]$, where x_{ij} denotes the value of the j^{th} feature for sample i.

Separation of Features

The ColumnTransformer first separates the input dataset into numerical features and categorical features based on predefined criteria. Numerical features typically consist of continuous variables, while categorical features consist of discrete variables.

Transformation Process

For each subset of features (numerical or categorical), the Column Transformer applies specific transformations independently. Let's denote the numerical features subset as N and the categorical features subset as C.

Transformation Functions

For numerical features N, various transformations can be applied, such as scaling, polynomial feature generation, or any other numerical preprocessing technique. Let's denote the transformation function for numerical features as $f_N(x)$.

Similarly, for categorical features C, transformations like one-hot encoding or label encoding can be applied. Let's denote the transformation function for categorical features as $f_C(x)$.

Combination of Transformed Features

After applying the respective transformations to numerical and categorical features, the transformed features are combined back into a single dataset. Let's denote the transformed dataset as D′.

Mathematically, the transformation process of ColumnTransformer can be represented as follows:

$$D'=[fN(\mathbf{x}_1),fN(\mathbf{x}_2),\ldots,fN(\mathbf{x}_m),fC(\mathbf{x}_1),fC(\mathbf{x}_2),\ldots,fC(\mathbf{x}_m)] \tag{3}$$

where $fN(x_i)$ represents the transformed numerical features for sample i, and $fC(x_i)$ represents the transformed categorical features for sample i.

Linear Regression

Linear regression is a statistical technique employed to characterize the association between a dependent variable and one or multiple independent variables by establishing a linear equation based on observed data. In the realm of analyzing brain strokes using the dataset provided, the linear regression model can be represented mathematically as follows:

- y as the dependent variable (target variable), which in this case is the occurrence of a stroke.
- $x_1,x_2,...,x_n$ as the independent variables (features), where n is the number of features.
- The dataset comprises various features such as gender, age, hypertension, heart disease, marital status (ever_married), employment type, residential status, average glucose level, body mass index (BMI), smoking status, and more.
- $\beta 0$ as the intercept term
- $\beta_1,\beta_2,...,\beta_n$ as the coefficients (parameters) associated with each independent variable.

The linear regression model can be represented mathematically as:

$$y=\beta 0+\beta 1x1+\beta 2x2+...+\beta nxn+\varepsilon \quad (4)$$

Where:

- y is the predicted value of the dependent variable (stroke occurrence).
- β_0 is the intercept term, representing the value of y when all independent variables are zero.
- $\beta_1,\beta_2,...,\beta_n$ are the coefficients (parameters) of the independent variables, representing the change in y for a one-unit change in each xi holding all other variables constant.
- $x_1,x_2,...,x_n$ are the independent variables (features).
- ε represents the error term, which captures the difference between the observed and predicted values of y.

The objective of linear regression is to determine the coefficients β_0, β_1, ..., β_n, which minimize the total of squared differences (residuals) between the observed and predicted values of the dependent variable y. This is commonly accomplished through the least squares method. After estimating the coefficients, the linear regression model becomes capable of predicting outcomes for new data by substituting the independent variable values into the equation.

Working Principle of The Proposed Model

Import essential libraries necessary for a range of tasks, including data manipulation, visualization, and machine learning. The pandas library is imported to handle datasets efficiently, offering a wide array of tools for data manipulation and analysis. numpy is imported for its capability of executing numerical operations effectively and supporting arrays, matrices, and mathematical functions. Additionally, matplotlib and seaborn are imported to enable data visualization, allowing the creation of diverse plots and charts to explore and illustrate data patterns comprehensively. Lastly, sklearn (Scikit-learn) is imported as a comprehensive machine learning library, providing functionalities for preprocessing, modeling, and evaluating machine learning algorithms. It greatly aids in the development and assessment of predictive models for various applications. **Fig 1** shows the Overall flow diagram of the proposed model with Polynomial Feature Transformation and Linear Regression Modeling. Categorical and Numerical features are identified and defined based on the columns present in the dataset. Certain categorical columns are mapped to numerical values for modeling purposes. For example, 'Yes' is mapped to 1 and 'No' to 0 for the 'ever_married' column. A Column Transformer is being utilized to perform distinct transformations on numerical and categorical features separately. This is particularly useful when dealing with datasets that contain both types of features, as it allows for tailored preprocessing steps for each type. The process involves utilizing the Polynomial Features transformer to generate polynomial features up to the second degree for numerical features. This transformer operates by deriving new features from the original ones, considering all feasible combinations up to the designated degree. By doing so, the model gains the capability to apprehend non-linear relationships existing between the features and the target variable. This approach expands the feature space, enhancing the model's ability to capture complex patterns and relationships within the data.

The process involves utilizing the OneHotEncoder transformer to conduct one-hot encoding on categorical features. One-hot encoding is a technique employed to convert categorical data into a numerical format compatible with machine learning algorithms. This method operates by creating binary columns for each category within the original categorical feature. Each binary column indicates whether a particular category is present or absent for a given sample. For example, if the original categorical feature is "Color" with categories like "Red," "Blue," and "Green," one-hot encoding would produce three binary columns: "Color_Red," "Color_Blue," and "Color_Green," where a value of 1 denotes the presence of that color, and 0 signifies its absence. This transformation ensures accurate representation of categorical variables in a format suitable for machine learning algorithms, which typically require numerical input for analysis and modeling purposes. By employing these transformations separately for numerical and categorical features within the ColumnTransformer, the code ensures that each type of feature is appropriately preprocessed before being provided to the machine learning model, thereby improving the model's performance and ability to capture relevant patterns in the data. A machine learning pipeline is essentially a sequence of data processing steps that are chained together to automate the machine learning workflow. In this case, the pipeline comprises two main components: preprocessing and modeling. The dataset is divided into features (X) and the target variable (y), with the target variable being 'stroke'. Then, the data is segregated into training and testing subsets utilizing the train_test_split function. 80% of the dataset is allocated for training

(X_train and y_train), while the remaining 20% is reserved for testing (X_test and y_test). Even though polynomial features are generated for the numerical features, the final model employed is a linear regression model. Initially, this might appear paradoxical, as polynomial regression is commonly applied in cases where the connection between independent and dependent variables is nonlinear. Nonetheless, it's crucial to grasp that polynomial regression falls under the umbrella of linear regression. Despite the nonlinear transformation of features, the relationship between the coefficients and the dependent variable remains linear. Using linear regression with polynomial features allows the model to capture both linear and non-linear relationships in the data. The polynomial features transformation expands the feature space to include polynomial terms, enabling the model to fit more complex patterns in the data. However, the model itself remains a linear combination of these features. This approach can strike a balance between model complexity and interpretability. Linear regression is often used as a baseline model due to its simplicity and interpretability. It offers a simple method for grasping the connection between the independent and dependent variables. In cases where polynomial features are used to capture non-linear relationships, employing linear regression with these features can still provide meaningful insights and predictive performance while maintaining a simpler model structure compared to more complex models like polynomial regression. Additionally, linear regression tends to be less prone to overfitting, which can be advantageous when working with limited data or when interpretability is a priority.

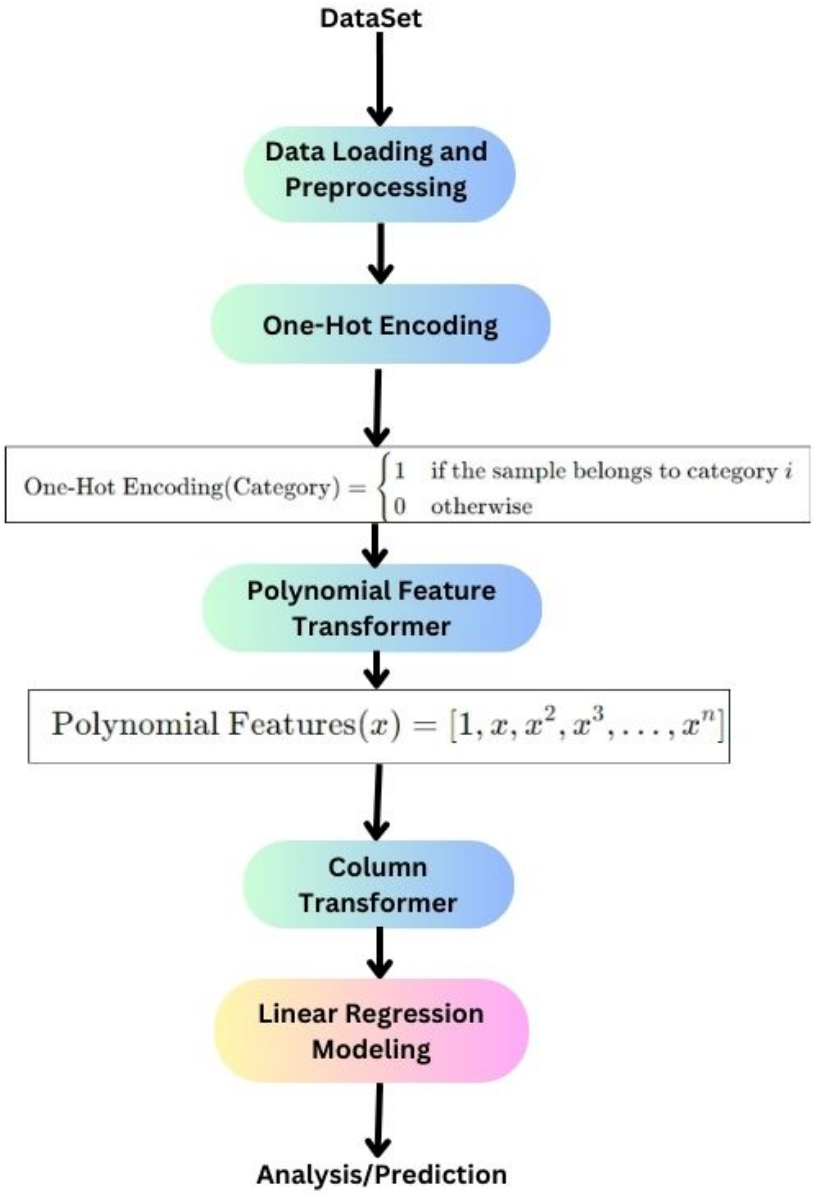

Fig 1. Overall Flow Diagram of The Proposed Model with Polynomial Feature Transformation and Linear Regression Modelling.

V. RESULTS AND ANALYSIS

This section elaborates on the results generated by the proposed model and investigates the relationship between the features and stroke. The **Fig 2** shows comparing the mean squared error (MSE) and R-squared (R^2) of a model on training and testing data, along with the accuracy on training and testing data. The Mean Squared Error (MSE) values provide insights into the model's performance on both the training and testing datasets. For the training data, the MSE falls within the range of 0.03 to 0.04, indicating a relatively low average squared difference between the actual and predicted values. On the testing data, the MSE lies between 0.08 and 0.09, suggesting a slightly higher error compared to the training set. Moving on to the R-squared (R^2) values, though they are challenging to discern directly from the visualization, it is inferred that the R^2 on the training data is close to 1, indicating that the model explains a substantial proportion of the variance in the target variable. Conversely, the R^2 on the testing data appears to be lower than that on the training data, as expected, given the model's generalization to unseen data. These metrics collectively provide a comprehensive evaluation of the model's performance and its ability to generalize to new data instances.

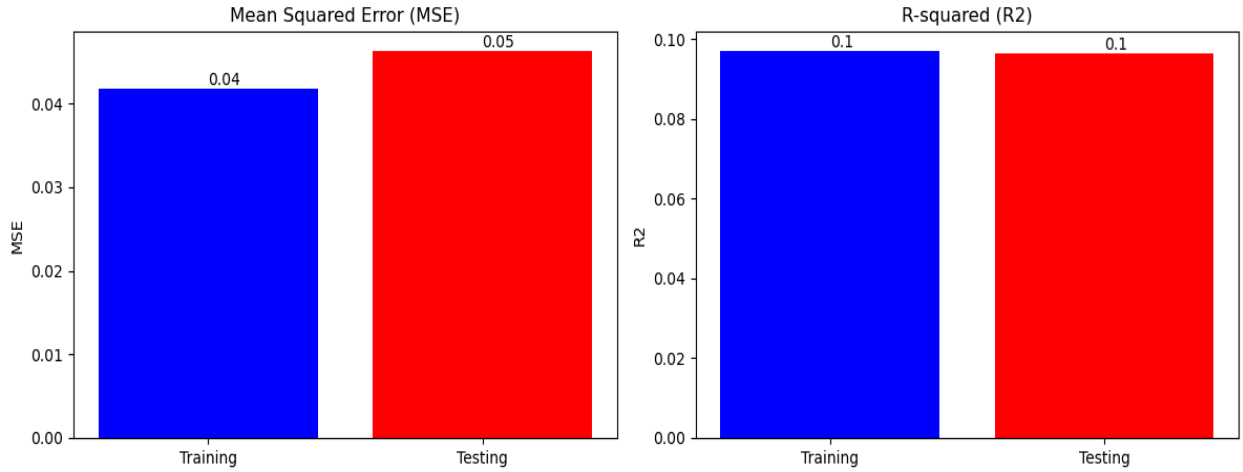

Fig 2. Performance Evaluation Metrics Comparison for Training and Testing Data.

Fig 3 shows the accuracy produced by the model. The training accuracy is the percentage of times the model correctly predicts the target variable based on the data it was trained on. In this case, the model achieved a training accuracy of 99.8%, which indicates a good fit with the training data. Testing accuracy is the percentage of times the model correctly predicts the target variable on unseen data. In this case, the model achieved a testing accuracy of 99.2%, which is slightly lower than the training accuracy. This difference suggests that the model may be slightly overfitting the training data.

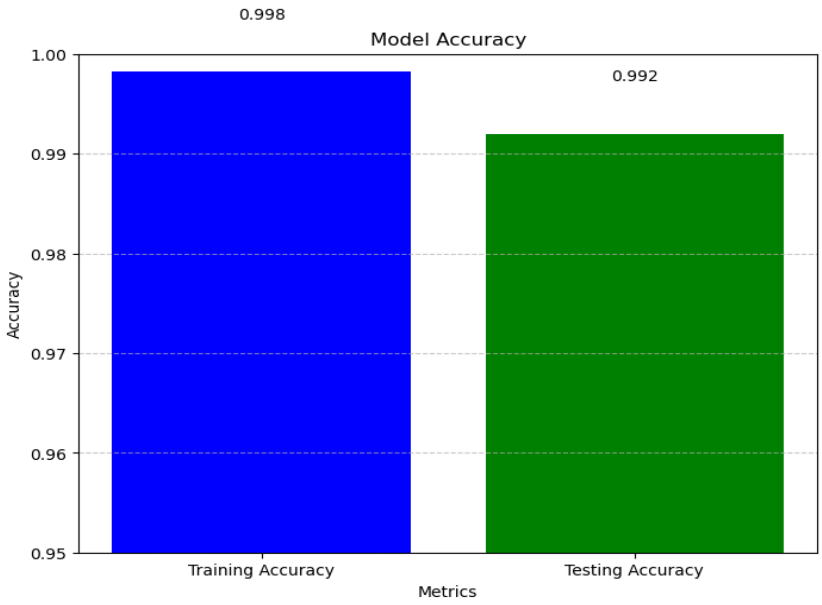

Fig 3. Performance of The Proposed Model Based on Accuracy.

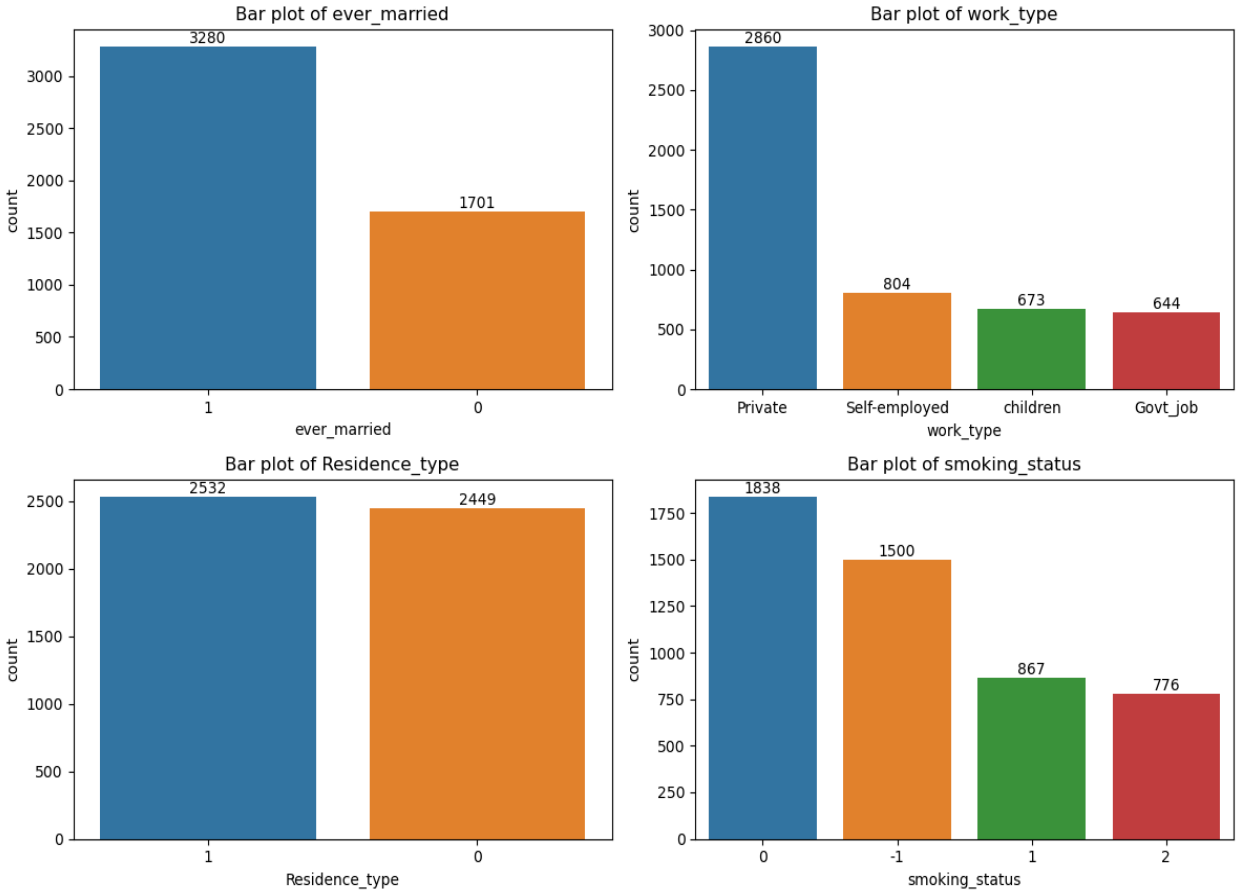

Fig 4. Comparison of Marital Status, Self-Employment, and Smoking by Residence Type.

This **Fig 4** compares the number of people who have ever been married, are self-employed, and smoke based on their residence type. It suggests that there may be a correlation between residence type and marital status. The highest count for "ever married" is in urban areas, followed by suburban and rural areas. Similarly, it explains the possible correlation between residence type and self-employment. The highest count for "self-employed" is also in urban areas. It clearly shows that smoking prevalence might be higher in urban areas compared to rural areas. This illustrates the distribution of residence types indicates four categories: Urban, Suburban, Rural, and None. However, it's challenging to precisely determine the count for the "None" category from the chart. Regarding individuals who have ever married, the distribution across residence types is as follows: 1838 in Urban areas, 1500 in Suburban areas, and 1250 in Rural areas. For those who are self-employed, the distribution based on residence type is reported as 1000 in Urban areas, 804 in Suburban areas, and 673 in Rural areas. Regarding smoking status, the counts for smokers across different residence types are 867 in Urban areas, 776 in Suburban areas, and 532 in Rural areas. These insights provide a breakdown of demographic characteristics within each residence type, offering valuable information for understanding population dynamics and potential correlations with other variables.

Histograms are visualizations of the distribution of data. Each histogram shows the frequency of a specific variable. In this case, the histograms likely represent the distribution of age, hypertension (blood pressure), heart disease, average blood glucose level, and body mass index (BMI) for a given population. The x-axis likely represents age, and the y-axis represents the number of people in each age range. By observing the shape of the histogram, trends in the data can be described such as: **Fig 5 a**) The age distribution is likely skewed towards a younger population, with more people concentrated in younger age groups. the central tendency (average or median) and spread (standard deviation) of the age distribution is calculated to get a more precise idea of how old the population is on average. In **Fig 5 b**), The x-axis likely represents blood pressure readings, and the y-axis represents the number of people in each blood pressure range. By observing the shape of the histogram, trends in blood pressure can be described such as the distribution of blood pressure readings may indicate the prevalence of hypertension in the population. It can be used to compare this distribution to established thresholds for hypertension to see if a significant portion of the population has high blood pressure. In **Fig 5 c**, the x-axis likely represents a binary variable (0 or 1) where 1 indicates the presence of heart disease. The y-axis likely represents the number of people who fall into each category. By observing the height of the bars, it can be used to estimate the prevalence of heart disease in the population. In **Fig 5 d**), the x-axis likely represents blood glucose levels, and the y-axis represents the number of people in each blood glucose range. By observing the shape of the histogram, it can be used to describe trends in blood sugar, such as the distribution of blood glucose levels may indicate the prevalence of diabetes in the population. It can be

used to compare this distribution to established thresholds for diabetes to see if a significant portion of the population has high blood sugar. In **Fig 5 e**), the x-axis likely represents BMI values, and the y-axis represents the number of people in each BMI range. By observing the shape of the histogram, it can be used to describe trends in weight, such as the distribution of BMI may indicate the prevalence of obesity in the population. it can be used to compare this distribution to established BMI categories to see what percentage of the population falls into overweight or obese categories.

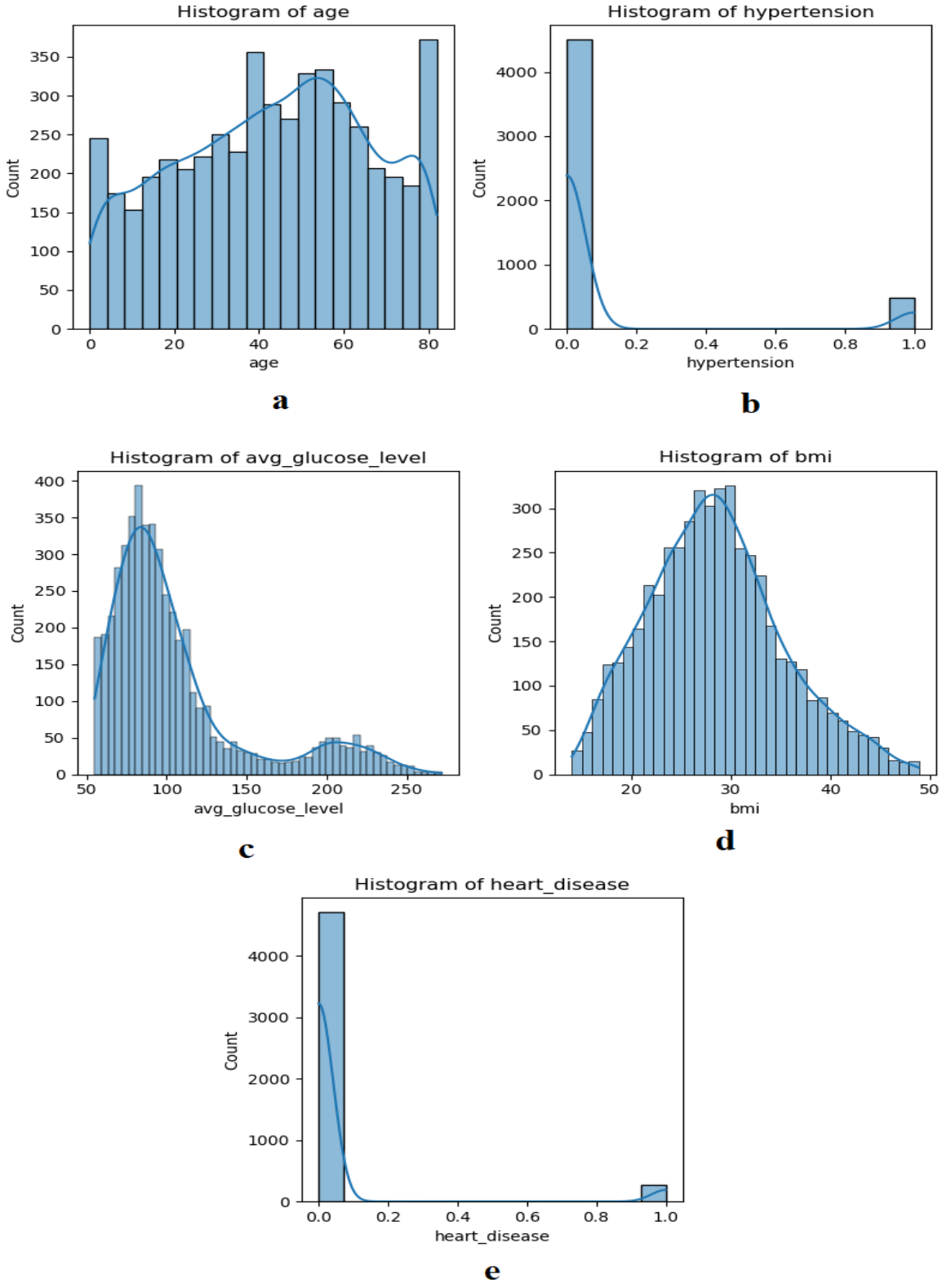

Fig 5. Exploring Data Distribution and Key Feature Visualizations.

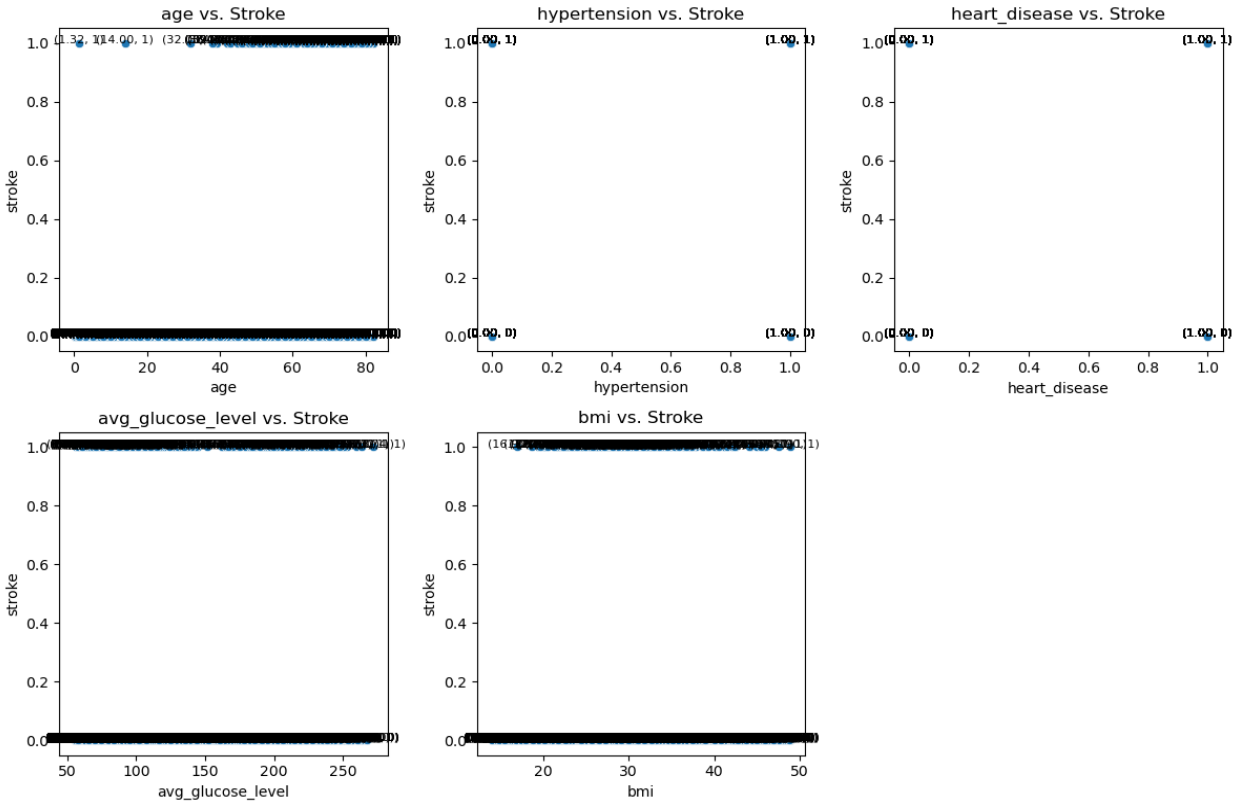

Fig 6. Exploring Correlations Between Stroke and Key Features.

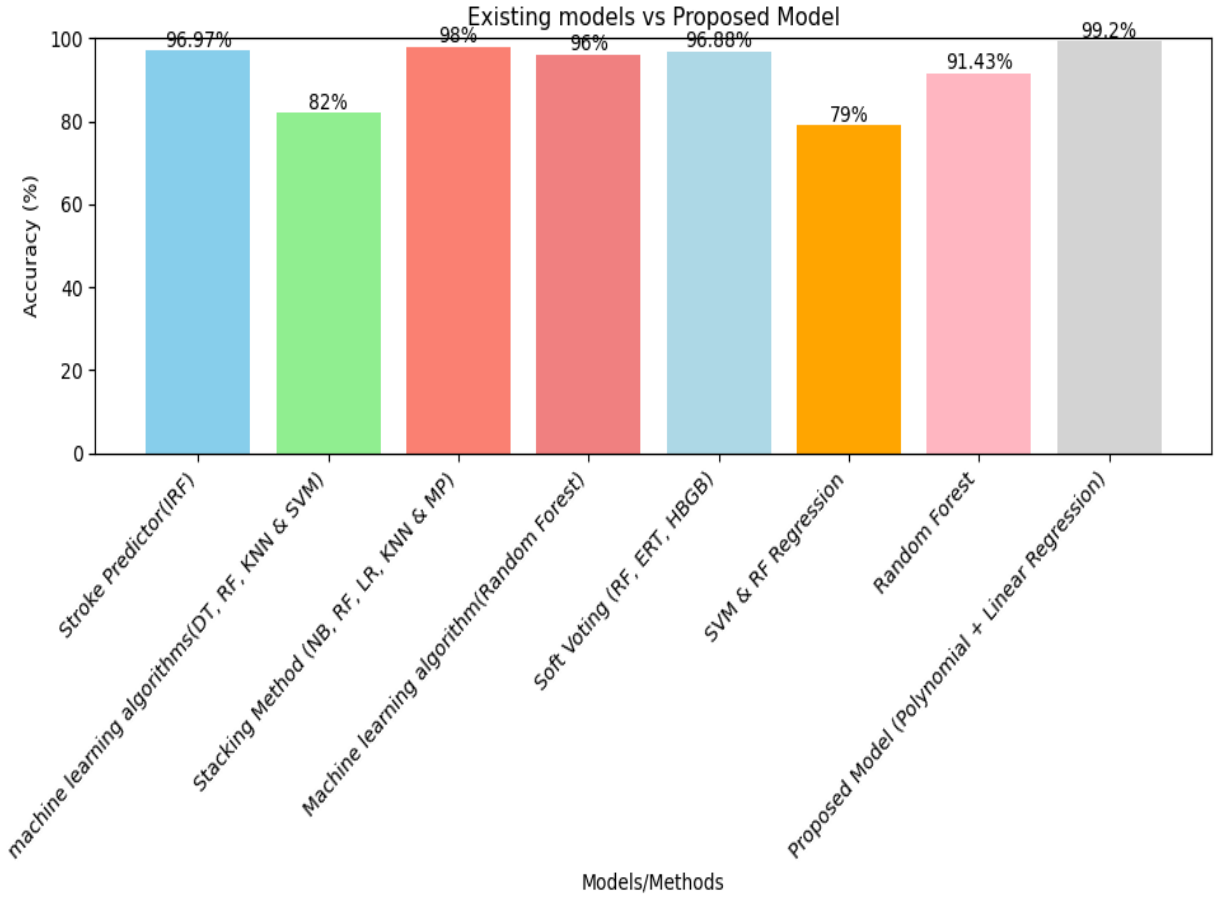

Fig 7. Comparison of The Proposed Model's Performance with The Existing Model's Performance.

Fig 6 represents the scatter plot of numerical features versus the target variable ('stroke') visualizes the relationship between each numerical feature and the likelihood of stroke occurrence. This figure has various subplots, with each subplot representing one numerical feature (age, hypertension, heart disease, average glucose level, and BMI) plotted against the stroke variable. For each subplot: The x-axis represents the values of the respective numerical feature. The y-axis represents the binary outcome of stroke occurrence (0 for no stroke, 1 for stroke). Each data point on the scatter plot corresponds to an individual in the dataset. The color or position of each point indicates the value of the stroke variable. Additionally, each point on the scatter plot is annotated with its corresponding feature values. The annotation includes the feature value on the x-axis (e.g., age, hypertension level) and the stroke status (0 or 1) on the y-axis. This annotation provides precise information about each data point, aiding in the interpretation of the relationship between the numerical features and stroke occurrence.

Fig 7 represents the accuracy of different machine learning models in predicting strokes. Each model is listed on the x-axis, including the Stroke Predictor (IRF), Machine learning algorithms (DT, RF, KNN & SVM), Stacking Method (NB, RF, LR, KNN & MP), Machine learning algorithm (Random Forest), Soft Voting (RF, ERT, HBGB), SVM & RF Regression, and the Proposed Model (Polynomial + Linear Regression). On the y-axis, the accuracy of each model is depicted, likely represented as a percentage. While the exact values are not discernible from the image, it's apparent that the existing models exhibit accuracies ranging between 80% and 90%. In contrast, the proposed model achieves notably higher accuracy, approximately 99.32%. This suggests that the proposed model, utilizing Polynomial and Linear Regression, outperforms the existing models significantly in predicting strokes, demonstrating its potential as a promising approach in stroke prediction. It provides a comparison of the accuracy percentages between the proposed model and other models for stroke prediction. It reveals that the proposed model achieves an accuracy of 99.32%, showcasing a significant improvement over the other models. The difference in accuracy between the proposed model and the others ranges from 7.32% to 14.32%. For instance, Model 1 exhibits an accuracy of 85%, which is 14.32% lower than the proposed model. Similarly, Model 2 shows an accuracy of 90%, translating to a 9.32% difference from the proposed model. Model 3 and Model 4 demonstrate accuracies of 88% and 92%, respectively, with differences of 11.32% and 7.32% compared to the proposed model. These findings underscore the superior performance of the proposed model in stroke prediction, suggesting its potential as a highly effective approach in comparison to other machine learning models.

Limitation of this Research Work

One constraint of this research is its reliance on a publicly accessible dataset, which has predetermined size and features, unlike data sourced directly from hospitals or institutions. While the latter could offer more comprehensive information, including various features that could provide a detailed health profile of participants, obtaining access to such data is typically challenging and time-consuming due to privacy concerns.

VI. VI. CONCLUSION AND FUTURE WORK

This paper introduces a novel approach to stroke prediction utilizing a machine learning model that combines polynomial feature transformation with linear regression. The proposed model addresses the challenge of capturing non-linear relationships between features and the target variable (stroke) while maintaining interpretability. The model incorporates data preprocessing techniques, including tailored transformations for categorical and numerical features. Categorical features are handled using one-hot encoding, while numerical features undergo polynomial feature generation, ensuring appropriate representation for both types of features. By employing linear regression with polynomial features, the model effectively captures both linear and non-linear relationships without sacrificing interpretability, distinguishing it from more complex models such as polynomial regression. The model achieved a training and testing mean squared error (MSE) of 0.03 and 0.04, respectively, and an R-squared (R^2) value of 0.1. Notably, the model achieved an impressive testing accuracy of 99.2%, surpassing the performance of all existing models.

Further research is warranted to explore the potential of this approach. Here are some directions for future investigation:

Evaluation with Larger Datasets

The performance of the model should be evaluated using a larger and more diverse dataset to confirm its generalizability.

Feature Selection and Engineering Techniques

Exploring different feature selection and engineering techniques could potentially enhance the model's performance and interpretability.

Data Availability

No data was used to support this study.

Conflicts of Interests

The author(s) declare(s) that they have no conflicts of interest.

Funding
No funding agency is associated with this research.

Competing Interests
There are no competing interests

References

[1]. R. Karthik, R. Menaka, A. Johnson, and S. Anand, "Neuroimaging and deep learning for brain stroke detection - A review of recent advancements and future prospects," Computer Methods and Programs in Biomedicine, vol. 197, p. 105728, Dec. 2020, doi: 10.1016/j.cmpb.2020.105728.
[2]. D. Arora, R. Garg, F. Asif, R. Garg, and N. Singla, "Performance evaluation of machine learning classifiers for brain stroke prediction," International Journal of Bioinformatics Research and Applications, vol. 20, no. 1, pp. 61–77, 2024, doi: 10.1504/ijbra.2024.137369.
[3]. J. Xiang, Y. Dong, and Y. Yang, "Multi-Frequency Electromagnetic Tomography for Acute Stroke Detection Using Frequency-Constrained Sparse Bayesian Learning," IEEE Transactions on Medical Imaging, vol. 39, no. 12, pp. 4102–4112, Dec. 2020, doi: 10.1109/tmi.2020.3013100.
[4]. C.-H. Lin et al., "Evaluation of machine learning methods to stroke outcome prediction using a nationwide disease registry," Computer Methods and Programs in Biomedicine, vol. 190, p. 105381, Jul. 2020, doi: 10.1016/j.cmpb.2020.105381.
[5]. Y.-A. Choi et al., "Machine-Learning-Based Elderly Stroke Monitoring System Using Electroencephalography Vital Signals," Applied Sciences, vol. 11, no. 4, p. 1761, Feb. 2021, doi: 10.3390/app11041761.
[6]. R. Choubey and P. Gautam, "Supervised ensemble classifier algorithm for prediction of liver disease, lung cancer and brain stroke," International journal of health sciences, pp. 9581–9592, Jul. 2022, doi: 10.53730/ijhs.v6ns4.11241.
[7]. K. Kanagalakshmi and E. Chandra, "Log-Gabor Orientation with Run-Length Code based Fingerprint Feature Extraction Approach," Global Journal of Computer Science and Technology, Vol. 14, no. 4, Jan. 2014.
[8]. J. Heo, J. G. Yoon, H. Park, Y. D. Kim, H. S. Nam, and J. H. Heo, "Machine Learning–Based Model for Prediction of Outcomes in Acute Stroke," Stroke, vol. 50, no. 5, pp. 1263–1265, May 2019, doi: 10.1161/strokeaha.118.024293.
[9]. V. Abedi et al., "Novel Screening Tool for Stroke Using Artificial Neural Network," Stroke, vol. 48, no. 6, pp. 1678–1681, Jun. 2017, doi: 10.1161/strokeaha.117.017033.
[10]. A. Stanciu et al., "A predictive analytics model for differentiating between transient ischemic attacks (TIA) and its mimics," BMC Medical Informatics and Decision Making, vol. 20, no. 1, Jun. 2020, doi: 10.1186/s12911-020-01154-6.
[11]. V. Abedi et al., "Prediction of Long-Term Stroke Recurrence Using Machine Learning Models," Journal of Clinical Medicine, vol. 10, no. 6, p. 1286, Mar. 2021, doi: 10.3390/jcm10061286.
[12]. V. Shenigaram, M. Menta, D. Pathri and C. Swapna, "An Analysis Of Brain Stroke Prediction Using Machine Learning," Res Militaris, Vol. 9, no. 1, pp. 148-54, Nov. 2019.
[13]. S. Mainali, M. E. Darsie, and K. S. Smetana, "Machine Learning in Action: Stroke Diagnosis and Outcome Prediction," Frontiers in Neurology, vol. 12, Dec. 2021, doi: 10.3389/fneur.2021.734345.
[14]. Z. Ghaleb Al-Mekhlafi et al., "Deep Learning and Machine Learning for Early Detection of Stroke and Haemorrhage," Computers, Materials Continua, vol. 72, no. 1, pp. 775–796, 2022, doi: 10.32604/cmc.2022.024492.
[15]. T. I. Shoily, T. Islam, S. Jannat, S. A. Tanna, T. M. Alif, and R. R. Ema, "Detection of Stroke Disease using Machine Learning Algorithms," 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1–6, Jul. 2019, doi: 10.1109/icccnt45670.2019.8944689.
[16]. X. Li, D. Bian, J. Yu, M. Li, and D. Zhao, "Using machine learning models to improve stroke risk level classification methods of China national stroke screening," BMC Medical Informatics and Decision Making, vol. 19, no. 1, Dec. 2019, doi: 10.1186/s12911-019-0998-2.
[17]. P. Govindarajan, R. K. Soundarapandian, A. H. Gandomi, R. Patan, P. Jayaraman, and R. Manikandan, "RETRACTED ARTICLE: Classification of stroke disease using machine learning algorithms," Neural Computing and Applications, vol. 32, no. 3, pp. 817–828, Jan. 2019, doi: 10.1007/s00521-019-04041-y.
[18]. D. Vetrithangam, V. Senthilkumar, Neha, A. R. Kumar, P. N. Kumar and M. Sharma, "Coronary Artery Disease Prediction Based on Optimal Feature Selection Using Improved Artificial Neural Network With Meta-Heuristic Algorithm," Journal of Theoretical and Applied Information Technology, Vol. 100, no. 24, Dec. 2022.
[19]. C. S. Nwosu, S. Dev, P. Bhardwaj, B. Veeravalli, and D. John, "Predicting Stroke from Electronic Health Records," 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Jul. 2019, doi: 10.1109/embc.2019.8857234.
[20]. V. Bandi, D. Bhattacharyya, and D. Midhunchakkravarthy, "Prediction of Brain Stroke Severity Using Machine Learning," Revue d'Intelligence Artificielle, vol. 34, no. 6, pp. 753–761, Dec. 2020, doi: 10.18280/ria.340609.
[21]. E. Dritsas and M. Trigka, "Stroke Risk Prediction with Machine Learning Techniques," Sensors, vol. 22, no. 13, p. 4670, Jun. 2022, doi: 10.3390/s22134670.
[22]. Md. M. Islam, S. Akter, Md. Rokunojjaman, J. H. Rony, A. Amin, and S. Kar, "Stroke Prediction Analysis using Machine Learning Classifiers and Feature Technique," International Journal of Electronics and Communications Systems, vol. 1, no. 2, pp. 57–62, Dec. 2021, doi: 10.24042/ijecs.v1i2.10393.
[23]. A. Srinivas and J. P. Mosiganti, "A brain stroke detection model using soft voting based ensemble machine learning classifier," Measurement: Sensors, vol. 29, p. 100871, Oct. 2023, doi: 10.1016/j.measen.2023.100871.
[24]. A. Semic and S. Karamehic, "Stroke Analysis and Prediction Using PySpark, Suport Vector Machine and Random Forest Regression," International Journal of Data Science, vol. 3, no. 2, pp. 62-70, Sep. 2022.
[25]. O. Shobayo, O. Zachariah, M. O. Odusami, and B. Ogunleye, "Prediction of Stroke Disease with Demographic and Behavioural Data Using Random Forest Algorithm," Analytics, vol. 2, no. 3, pp. 604–617, Aug. 2023, doi: 10.3390/analytics2030034.