Brain Stroke Prediction

Dissertation submitted in fulfilment of the requirements for the Degree of

BACHELORS OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

WITH SPECIALIZATION IN DATA SCIENCE(AI & ML)

By

Garvit Joshi

12106692

Supervisor

Mr. Ved Prakash Chaubey

School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab (India)

2021 - 2025

DECLARATION STATEMENT

I hereby declare that the research work reported in the dissertation/dissertation proposal entitled "BRAIN STROKE PREDICTION USING MACHINE LEARNING" in partial fulfilment of the requirement for the award of Degree for Bachelor of Technology in Computer Science and Engineering at Lovely Professional University, Phagwara, Punjab is an authentic work carried out under the supervision of my research supervisor Mr. Ved Prakash Chaubey Sir. I have not submitted this work elsewhere for any degree or diploma.

I understand that the work presented herewith is in direct compliance with Lovely Professional University's Policy on plagiarism, intellectual property rights, and the highest standards of moral and ethical conduct. Therefore, to the best of my knowledge, the content of this dissertation represents an authentic and honest research effort conducted, in its entirety, by me. I am fully responsible for the contents of my dissertation work.

Signature of Candidate

Garvit Joshi

12106692

SUPERVISOR'S CERTIFICATE

This is to certify that the work reported in the B. Tech Dissertation/dissertation proposal entitled "**BRAIN STROKE PREDICTION USING MACHINE LEARNING"**, submitted by **Garvit Joshi** at **Lovely Professional University, Phagwara, India** is a Bonafide record of his / her original work carried out under my supervision. This work has not been submitted elsewhere to any other degree.

Signature of Supervisor

(Mr. Ved Prakash Chaubey)

Date: 23-04-2024

Counter Signed by:

1) **Concerned HOD:**
   HoD's Signature: _____

   HoD Name: _____

   Date: _____

2) **Neutral Examiners:**

   External Examiner

   Signature: _____

   Name: _____

   Affiliation: _____

   Date: _____

   Internal Examiner

   Signature: _____

   Name: _____

   Date: _____

TABLE OF CONTENT

Checklist for Dissertation-III Supervisor

Name: _____UID: _____Domain: _____
Registration No: _____Name of student:_____

Title of Dissertation:

_____

- ☐ Front pages are as per the format.
- ☐ Topic on the PAC form and title page are same.
- ☐ Front page numbers are in roman and for report, it is like 1, 2, 3…….
- ☐ TOC, List of Figures, etc. are matching with the actual page numbers in the report.
- ☐ Font, Font Size, Margins, line Spacing, Alignment, etc. are as per the guidelines.
- ☐ Color prints are used for images and implementation snapshots.
- ☐ Captions and citations are provided for all the figures, tables etc. and are numbered and center aligned.
- ☐ All the equations used in the report are numbered.
- ☐ Citations are provided for all the references.
- ☐ **Objectives are clearly defined.**
- ☐ Minimum total number of pages of report is 50.
- ☐ Minimum references in report are 30.

Here by, I declare that I had verified the above mentioned points in the final dissertation report.

Signature of Supervisor with UID

Abstract

A stroke, often referred to as a cerebrovascular accident (CVA), occurs when a portion of the brain loses blood flow, which causes the area of the body that those brain cells control to become dysfunctional. Because of poor blood flow or bleeding into the brain tissue, this decrease of blood supply may be ischemic or hemorrhagic. Due to the possibility of fatality or lifelong impairment, a stroke is a medical emergency. Ischemic strokes may be managed; however, this treatment must begin within a few hours of the onset of stroke symptoms. If a stroke is suspected, the patient, their family, or witnesses should call emergency medical assistance right away. A transient ischemic attack (TIA or mini stroke) is a brief ischemic stroke in which the symptoms disappear on their own. This situation also demands an immediate evaluation to lower the risk of a future stroke. If all symptoms go away within 24 hours, that would be a stroke, not a TIA. Stroke is the second leading cause of mortality globally, claims the World Health Organization (WHO). accounting for around 11% of all fatalities. Our ML model uses a dataset for survival prediction to determine a patient's likelihood of suffering a stroke based on inputs including gender, age, various illnesses, and smoking status. Our dataset, in contrast to most others, concentrates on characteristics that would be significant risk factors for a brain stroke.

Introduction

Brain stroke, also known as a cerebrovascular accident (CVA), is a medical emergency characterized by the sudden interruption of blood flow to the brain, leading to a range of neurological impairments. According to the World Health Organization (WHO), strokes are one of the leading causes of mortality and long-term disability worldwide. Early detection and prompt intervention are crucial in preventing the devastating consequences of strokes and improving patient outcomes.

In recent years, advancements in machine learning and artificial intelligence have revolutionized healthcare by enabling the development of predictive models for various medical conditions. Leveraging the power of machine learning algorithms, researchers and healthcare professionals have been exploring the potential to predict brain strokes based on a combination of demographic, clinical, and lifestyle factors.

This project aims to contribute to the ongoing efforts in stroke prevention and management by developing a machine learning-based predictive model for brain stroke prediction. By analyzing relevant patient data, including demographic information, medical history, and lifestyle factors, the model seeks to identify individuals at risk of experiencing a stroke. Early identification of high-risk individuals can facilitate timely interventions, such as lifestyle modifications, medication, or medical procedures, to mitigate the risk of stroke occurrence.In this report, we present the methodology, findings, and implications of our brain stroke prediction model. We discuss the data collection process, feature selection, model development, and evaluation metrics used to assess the performance of the predictive model. Furthermore, we explore the potential applications of the model in clinical practice and discuss avenues for future research and improvement.

By harnessing the capabilities of machine learning, this project endeavours to contribute to the advancement of stroke prevention strategies and ultimately enhance the quality of care for individuals at risk of experiencing a brain stroke.

Problem Statement

1. Significant Burden on Society:

Problem: Brain stroke is a leading cause of death and disability worldwide. According to the World Health Organization (WHO), stroke is the second leading cause of death globally, responsible for approximately 14.4 million deaths each year [WHO, Prevent Brain Stroke].

Data Impact: This immense death toll translates to a significant loss of human life and potential. It creates a burden on families and healthcare systems.

2. Rising Healthcare Costs:

Problem: Stroke survivors often require long-term care and rehabilitation, leading to substantial healthcare costs. A study published in the Journal of the American Heart Association found that the total annual cost of stroke in the United States is estimated to be over $80 billion [Journal of the American Heart Association, on the cost of stroke care in the US].

Data Impact: These ever-increasing costs strain healthcare budgets and limit resources available for other medical needs.

3. Devastating Long-Term Effects:

Problem: Stroke can lead to various long-term disabilities, including paralysis, speech impairments, cognitive decline, and emotional problems. This significantly impacts a survivor's quality of life and ability to function independently.

Data Impact: Millions of people live with stroke-related disabilities, reducing their overall well-being and societal participation.

Methodology for Brain Stroke Prediction Model

a) Data Collection:

Collect diverse datasets containing demographic, clinical, and lifestyle factors relevant to stroke prediction. Ensure data quality and integrity by verifying sources, addressing missing values, and removing duplicates if any.

b) Data Preprocessing:

Clean and preprocess the collected data by handling missing values, encoding categorical variables, and scaling numerical features. Perform exploratory data analysis (EDA) to gain insights into the distribution and characteristics of the data.

c) Model Development:

Select appropriate machine learning algorithms for stroke prediction, such as logistic regression, decision trees, random forests, or neural networks. Design the architecture of the predictive model, including the number of layers, neurons, and activation functions. Incorporate feature selection techniques to identify the most relevant predictors for stroke risk.

d) Model Training:

Split the preprocessed data into training and testing sets to train and evaluate the performance of the predictive model. Utilize training algorithms to optimize model parameters and minimize prediction errors. Validate the model using cross-validation techniques to assess its robustness and generalizability.

e) Deployment and Future Directions:

Deploy the trained model in a real-world setting, such as a healthcare facility or online platform, for stroke risk assessment. Monitor the performance of the deployed model and collect feedback from users for continuous improvement. Explore future directions for enhancing the model's predictive accuracy and scalability, such as incorporating additional features, refining algorithms, or leveraging emerging technologies like deep learning and reinforcement learning.

Data Collection

We acquired datasets from Kaggle, encompassing demographic, clinical, and lifestyle factors, and then utilized Python libraries to preprocess and analyze the data. Numpy handled numerical computations, pandas managed data manipulation, and Matplotlib facilitated data visualization. Numpy supported efficient array operations, pandas enabled tabular data manipulation, and Matplotlib visualized insights for exploratory analysis.

Attribute	Description
id	Unique identifier
gender	"Male", "Female" or "Other"
age	Age of the patient
hypertension	0 if the patient doesn't have hypertension, 1 if the patient has hypertension
heart_disease	0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease
ever_married	"No" or "Yes"
work_type	"Children", "Govt_job", "Never_worked", "Private" or "Self-employed"
Residence_type	"Rural" or "Urban"
avg_glucose_level	Average glucose level in blood
bmi	Body mass index
smoking_status	"Formerly smoked", "Never smoked", "Smokes" or "Unknown"*
stroke	1 if the patient had a stroke or 0 if not

```
import numpy as np
# NumPy is imported for numerical operations and array manipulations, commonly used for data processing and scientific computing.
import pandas as pd
# Pandas is imported for data manipulation and analysis, particularly for handling structured data in tabular form (dataframes).
import matplotlib.pyplot as plt
# Matplotlib is imported for creating static, interactive, and animated visualizations in Python.
import seaborn as sns
# Seaborn is imported for statistical data visualization, providing a high-level interface for drawing attractive and informative statistical graphics.
```

✓ 3.3s

Data Preprocessing

In the data preprocessing phase of our brain stroke prediction project, we focus on two essential tasks: Exploratory Data Analysis (EDA) and handling null values.

Exploratory Data Analysis (EDA):

EDA involves examining the dataset to understand its structure, distribution, and relationships between variables. Through EDA, we gain valuable insights into the data that inform subsequent preprocessing steps and model development. Key aspects of EDA include:

1. Summary Statistics: Computing descriptive statistics such as mean, median, and standard deviation to understand the central tendency and dispersion of numerical features.
2. Data Visualization: Creating visualizations such as histograms, box plots, and scatter plots to visualize the distribution of data, identify outliers, and explore relationships between variables.
3. Correlation Analysis: Calculating correlation coefficients to quantify the strength and direction of relationships between numerical variables.

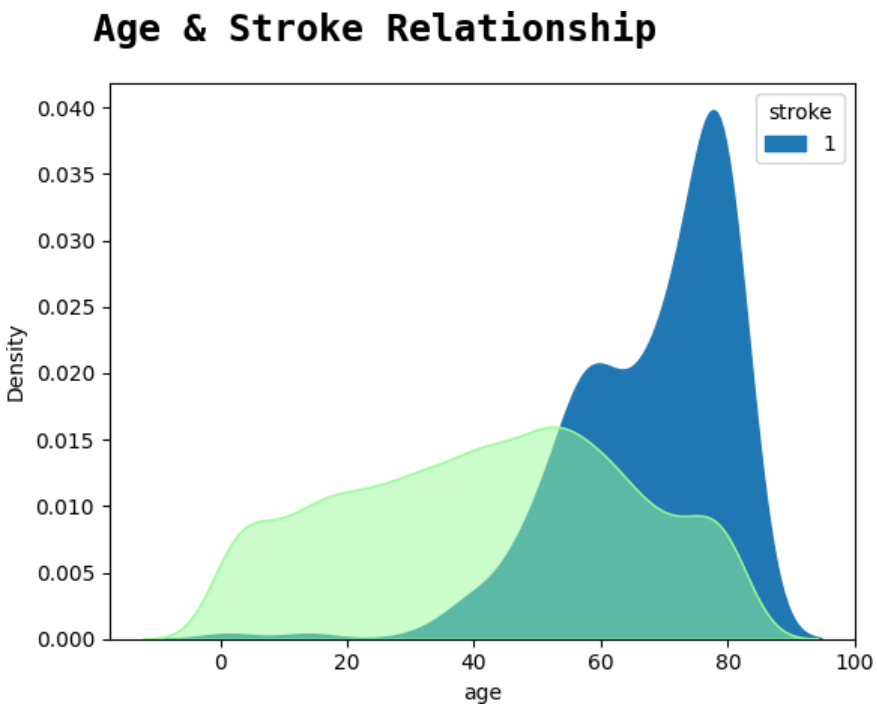
Handling Null Values:

Null values, or missing values, are common in real-world datasets and can adversely affect the performance of machine learning models. To address null values:

1. Identification: We identify the presence of null values in the dataset using functions like isnull() or info().
2. Imputation: For numerical features, we may impute missing values using strategies such as mean, median, or interpolation. For categorical features, we may impute missing values with the mode or a separate category.
3. Deletion: In cases where null values are too numerous or cannot be reliably imputed, we may opt to remove rows or columns.

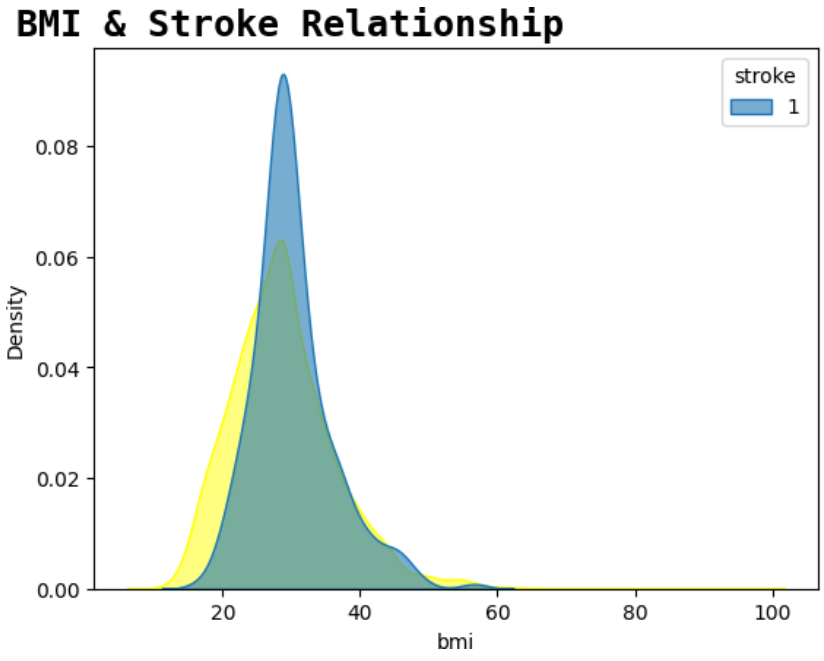
Correlation in Columns:

1. Age-Storke:



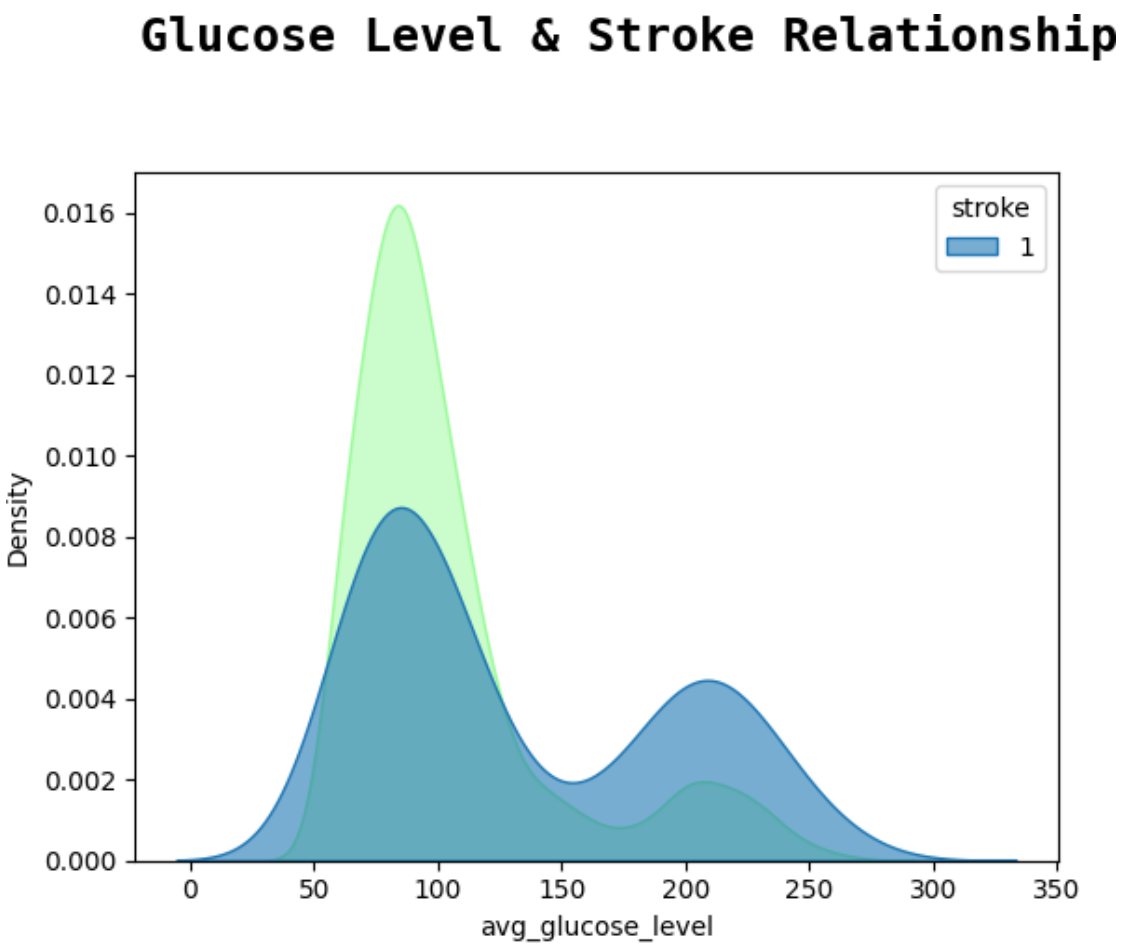
- There is a strong correlation between age and stroke risk.
- Age is a significant factor in the risk of stroke.

2. BMI-Stroke:



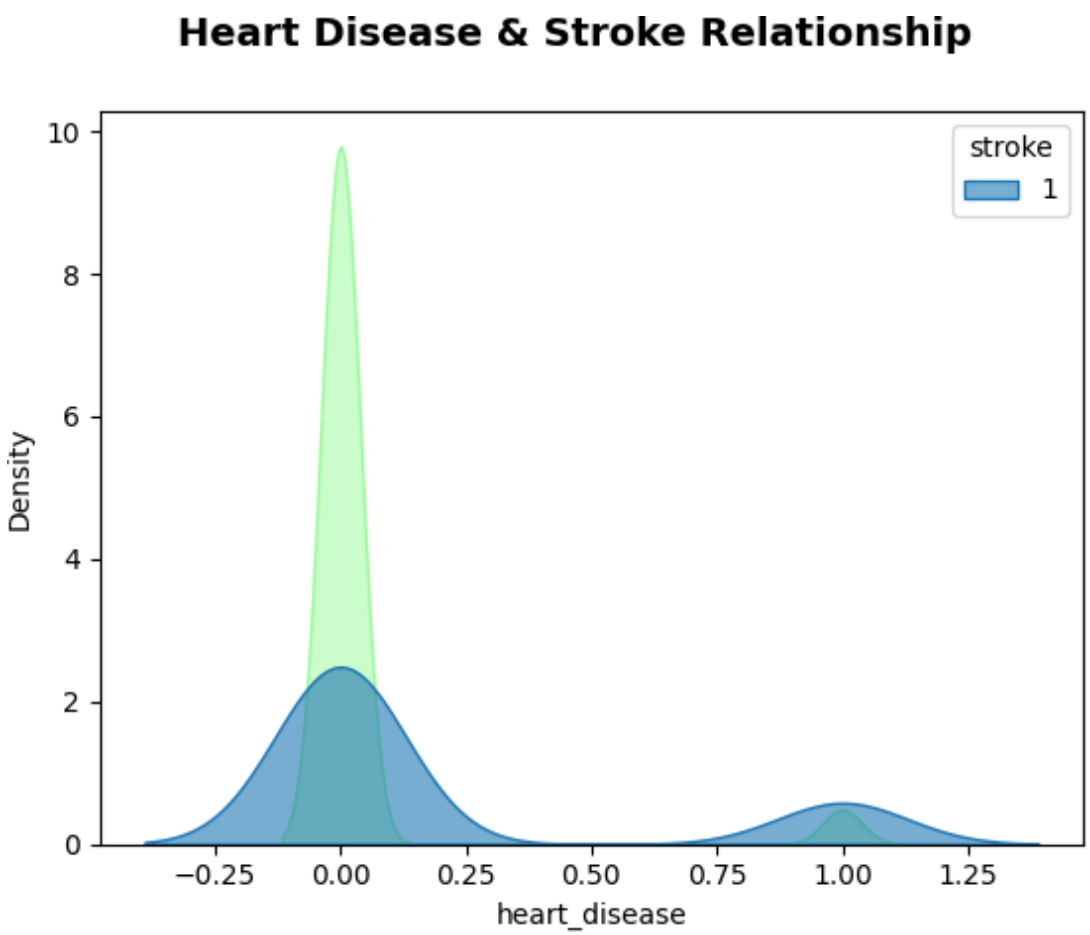
- There is a positive correlation between BMI and stroke risk. This means that as BMI increases, the risk of stroke also increases.
- People with a higher BMI are more likely to experience a stroke than people with a lower BMI.

3. Glucose_Level-Stroke:



- The graph does not show a clear relationship between glucose level and stroke. The data points are scattered throughout the graph, and there is no clear trend.
- The average glucose level in the graph is higher than the average stroke level. This is because the line labeled "avg_glucose_level" is higher than the horizontal line labeled "stroke". However, it is important to note that the graph does not show the range of normal glucose levels or stroke levels.

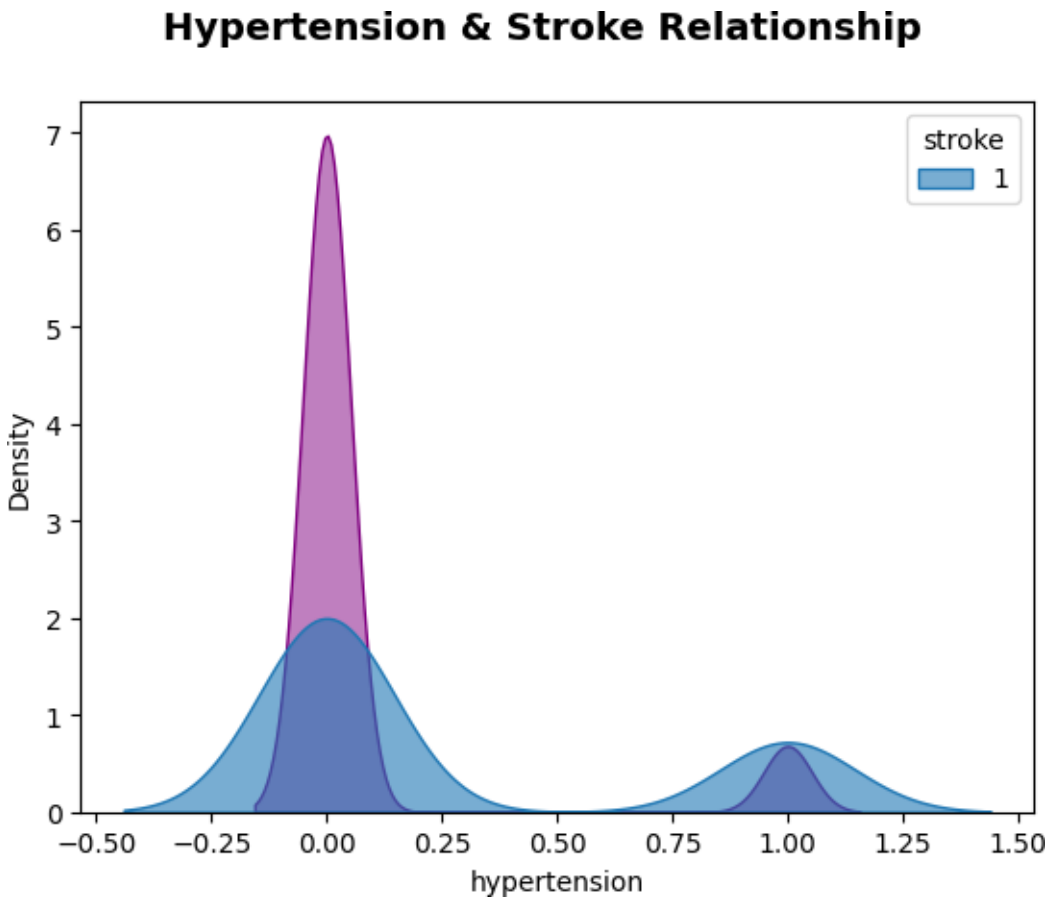
4. Heart_Disease-Stroke:



- There is a positive correlation between heart disease and stroke. This means that as the level of heart disease increases, the density of stroke also increases.
- Heart disease is a major risk factor for stroke. People with heart disease are more likely to have a stroke than people without heart disease.

5. Hypertension-Stroke:

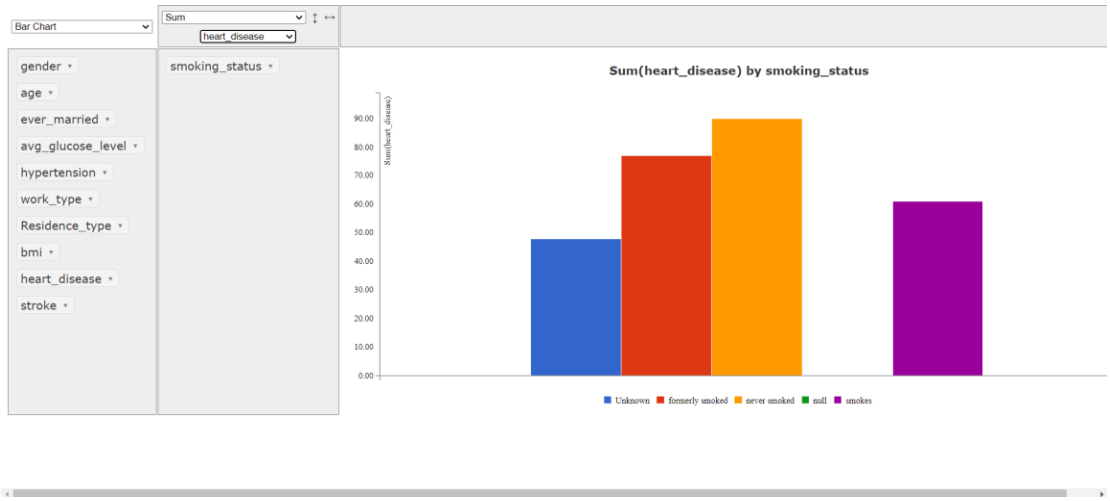
- There is a strong correlation between hypertension and stroke. This means that as hypertension levels increase, the likelihood of stroke also increases.
- Hypertension is a major risk factor for stroke. People with hypertension are more likely to have a stroke than people with normal blood pressure.



To create some visualizations without writing code:

```
from pivottablejs import pivot_ui
pivot_ui(df)
```

Use pivottable, this allows you to create different basic visualizations to save time and help you analyse clearly.



Model Development

Before proceeding with model development, it's essential to preprocess the data and ensure that all features are in a suitable format for analysis. In our brain stroke prediction project, we convert categorical variables into a numerical format using a process called encoding. We replace categorical labels with corresponding numerical values to facilitate model training.

For example:

a) Gender: 'Male' is encoded as 0, 'Female' as 1, and 'Other' as -1.

Residence Type: 'Rural' is encoded as 0 and 'Urban' as 1.

Work Type: 'Private' as 0, 'Self-employed' as 1, 'Govt_job' as 2, 'children' as -1, and 'Never_worked' as -2.

We perform this encoding using the replace() function and ensure that the encoded values are of the appropriate data type, typically unsigned integers (np.uint8).

Data Splitting for Training and Testing:

After encoding the categorical variables, we proceed to split the dataset into training and testing sets. This split allows us to train the predictive model on a portion of the data and evaluate its performance on unseen data. We use the train_test_split() function from the sklearn.model_selection module to randomly divide the dataset into training and testing sets, following an 80-20 ratio (80% for training and 20% for testing). Additionally, we set a random state (random_state=42) to ensure reproducibility of the split.

Once the data is split, we verify the dimensions of the training and testing sets using the shape attribute and print the results to confirm the correctness of the split. This step ensures that the data is properly partitioned and ready for model training and evaluation.

1. Logistic Regression:

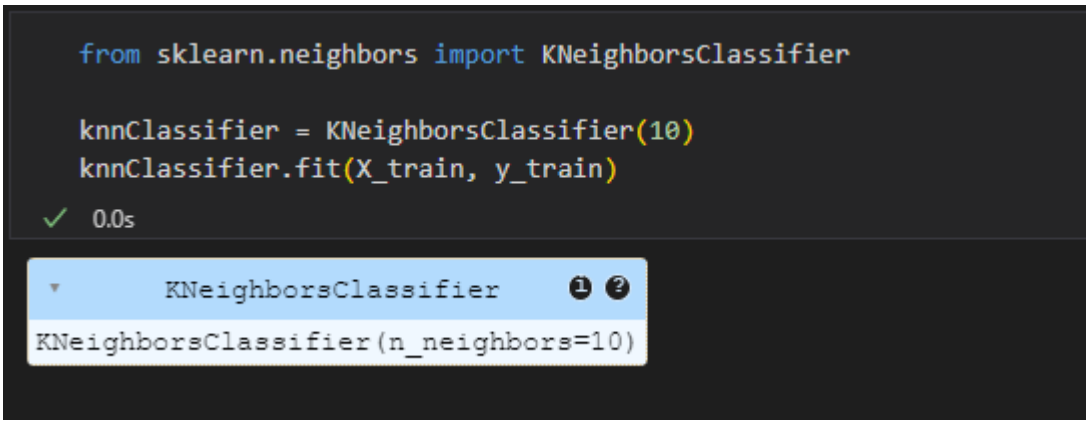
Description: Logistic regression is a simple yet powerful algorithm suitable for binary classification tasks like stroke prediction. It provides interpretable results and works well with linearly separable data.

Accuracy: 0.939335

2. K-Nearest Neighbors (KNN):

Description: KNN is a non-parametric algorithm that classifies data points based on the majority class among their k nearest neighbors. It is intuitive, easy to understand, and can capture local patterns in the data.

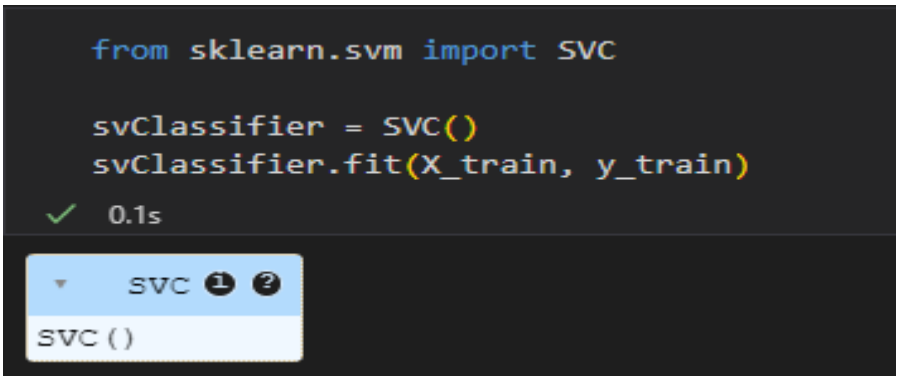
Accuracy: 0.939335



3. Support Vector Machine (SVM):

Description: SVM is effective in separating classes by finding the optimal hyperplane that maximizes the margin between them. It works well with high-dimensional data and is robust to overfitting.

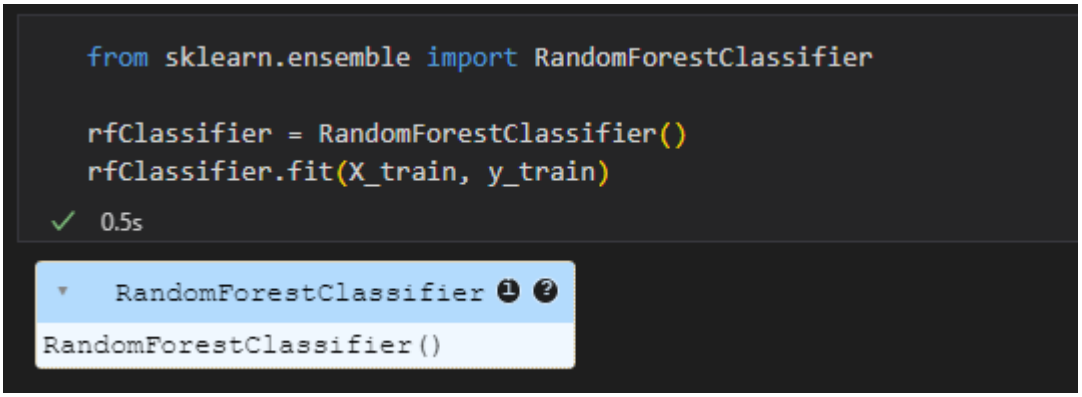
Accuracy: 0.939335



4. Random Forest:

Description: Random Forest is an ensemble learning algorithm that combines multiple decision trees to improve predictive performance. It can handle non-linear relationships and interactions between features and is robust to overfitting.

Accuracy: 0.940313



5. Naive Bayes:

Description: Naive Bayes is a probabilistic classifier based on Bayes' theorem. It assumes independence between features, making it computationally efficient and robust to irrelevant features.

Accuracy: 0.853229

```
from sklearn.naive_bayes import GaussianNB

nbClassifier = GaussianNB()
nbClassifier.fit(X_train, y_train)
```

✓ 0.0s

```
GaussianNB()
```

6. Decision Tree:

Description: Decision trees are interpretable models that partition the feature space into regions based on feature thresholds. They are robust to outliers and can capture non-linear relationships in the data.

Accuracy: 0.917808

```
from sklearn.tree import DecisionTreeClassifier
dtClassifier = DecisionTreeClassifier()
dtClassifier.fit(X_train, y_train)
```

✓ 0.0s

```
DecisionTreeClassifier()
```

7. Gradient Boosting Machines (GBM):

Description: GBM builds trees sequentially, correcting errors of the previous one. It generally provides better accuracy compared to single decision trees and is robust to outliers.

Accuracy: 0.940313

```
from sklearn.ensemble import GradientBoostingClassifier

gbm_classifier = GradientBoostingClassifier(random_state=42)
gbm_classifier.fit(X_train, y_train)
```

✓ 0.6s

```
GradientBoostingClassifier
GradientBoostingClassifier(random_state=42)
```

8. XGBoost (Extreme Gradient Boosting):

Description: XGBoost is an advanced implementation of gradient boosting known for its efficiency and scalability. It automatically handles missing values and is robust to outliers.

Accuracy: 0.933464

```
import xgboost as xgb

xgb_classifier = xgb.XGBClassifier(random_state=42)
xgb_classifier.fit(X_train, y_train)
```

✓ 6.7s

```
XGBClassifier
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=None, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=None, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              multi_strategy=None, n_estimators=None, n_jobs=None,
              num_parallel_tree=None, random_state=42, ...)
```

9. AdaBoost (Adaptive Boosting):

Description: AdaBoost sequentially corrects mistakes of weak classifiers, combining multiple weak classifiers to build a strong one. It is effective in binary classification problems like stroke prediction.

Accuracy: 0.939335

```
from sklearn.ensemble import AdaBoostClassifier

adaboost_classifier = AdaBoostClassifier(random_state=42)
adaboost_classifier.fit(X_train, y_train)
```

✓ 0.2s

```
c:\Users\LENOVO\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\ensemble\_weight_boosting.py:519:
  warnings.warn(
```

```
AdaBoostClassifier(random_state=42)
```

10. Neural Networks:

Description: Neural networks can capture complex non-linear relationships in the data and are suitable for large datasets with high-dimensional features. They can automatically learn feature representations from the data, making them powerful for stroke prediction tasks.

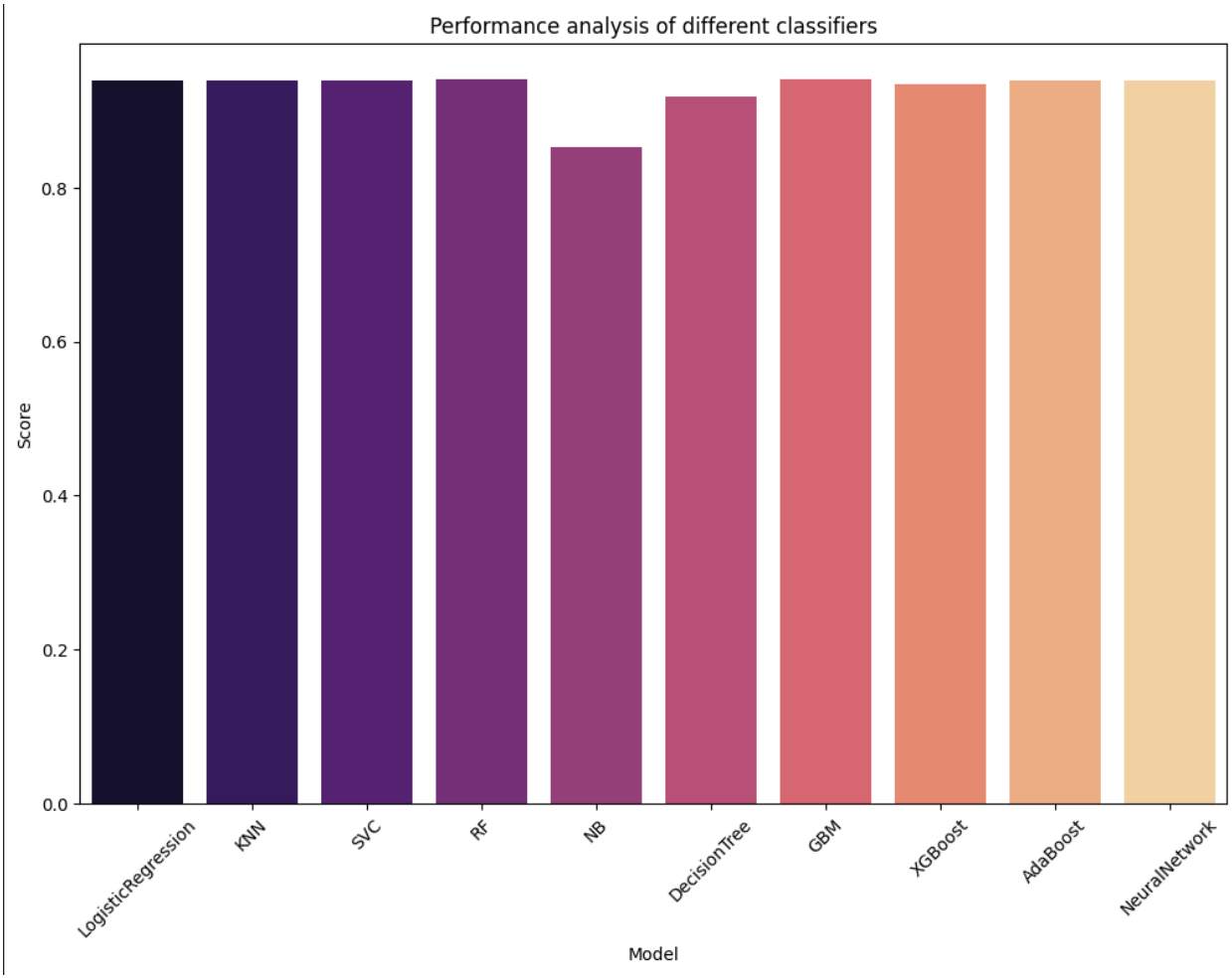
Accuracy: 0.939335

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

# Define the number of features
num_features = X_train.shape[1]  # Number of features is equal to the number of columns in X_train

# Define the model architecture
model = Sequential([
    Dense(64, activation='relu', input_shape=(num_features,)),
    Dense(64, activation='relu'),
    Dense(1, activation='sigmoid')
])

# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Train the model
model.fit(X_train, y_train, epochs=10, batch_size=32)
```

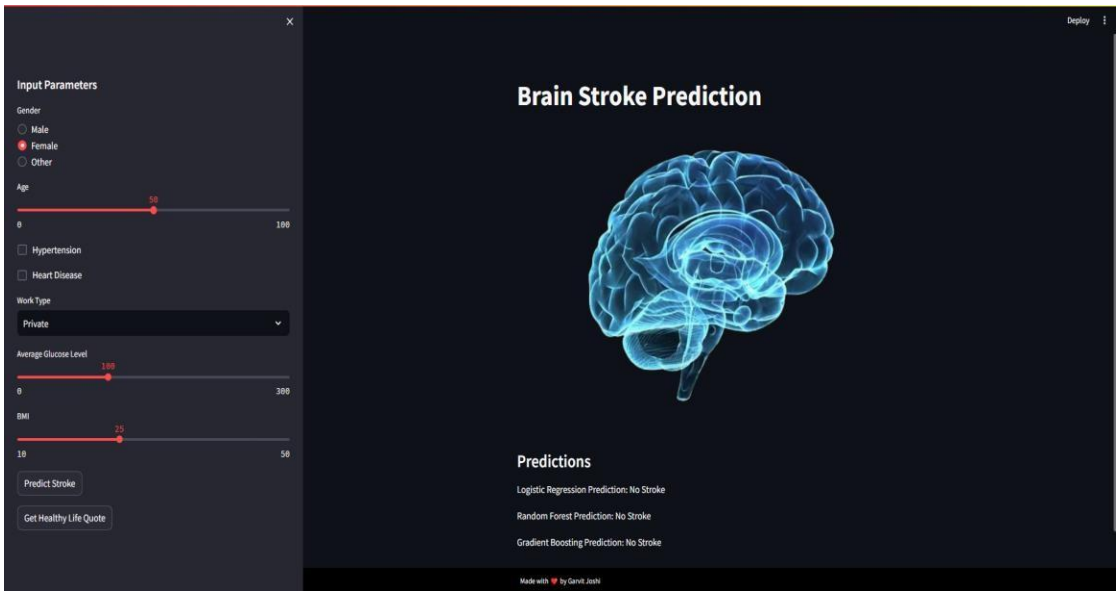
✓ 3.8s

```
Epoch 1/10
C:\Users\LENOVO\AppData\Roaming\Python\Python311\site-packages\keras\src\layers\core\dense.py:86: UserWarning: Do not pass
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
128/128 ━━━━━━━━━━━━━━━━━━━━ 2s 1ms/step - accuracy: 0.9519 - loss: 0.2655
Epoch 2/10
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.9569 - loss: 0.1844
Epoch 3/10
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.9479 - loss: 0.2045
Epoch 4/10
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.9480 - loss: 0.2450
Epoch 5/10
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.9497 - loss: 0.2148
Epoch 6/10
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.9521 - loss: 0.1986
Epoch 7/10
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.9528 - loss: 0.1924
Epoch 8/10
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.9487 - loss: 0.2032
Epoch 9/10
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.9555 - loss: 0.1903
Epoch 10/10
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.9564 - loss: 0.1877
```

Visualization for Summary of Scores of Each Model



Deployment



What is Streamlit?

Streamlit is an open-source Python library specifically designed to simplify the creation of data apps. It allows you to rapidly build user interfaces for your machine learning models and data analysis projects entirely in Python, without needing expertise in web development frameworks like HTML, CSS, or Javascript.

Why Streamlit for this project?

Here's why Streamlit is a great choice for deploying your stroke prediction model:

- **Ease of Use:** Streamlit focuses on simplicity. You can write Python code with Streamlit functions to create interactive elements like buttons, text boxes, and charts. This makes it ideal for data scientists and machine learning engineers who may not have extensive web development experience.
- **Fast Prototyping:** Streamlit allows you to quickly build a functional prototype of your web app. This lets you get feedback from users and iterate on the design efficiently.
- **Lightweight Deployment:** Streamlit apps are lightweight and can be deployed on various platforms, including the free Streamlit Community Cloud. This makes it easy to share your app with others without needing complex server setups.

Features in the Stroke Prediction Page:

Based on the image you described, here are some potential Streamlit features in your stroke prediction app:

- **User Input Widgets:** Streamlit offers various widgets like text boxes, dropdown menus, and radio buttons. These allow users to input data points like age, gender, and health conditions, which are then fed into the machine learning models.
- **Model Predictions:** Streamlit lets you integrate your machine learning models into the app. When the user submits their data, the models generate predictions for stroke risk (e.g., "Stroke" or "No Stroke") and display them in the app.

Future Directions

1. Integration into Wearable Devices:

Deploying the stroke prediction model into wearable devices, such as smartwatches, can provide real-time monitoring and alerts for individuals at risk. By developing an application that continuously monitors vital signs and other relevant parameters, users can receive alerts when certain parameters indicate a potential stroke risk. This proactive approach empowers individuals to take timely preventive measures and seek medical attention if necessary.

2. Healthcare Software Solutions:

Extending the model into a more precise software solution for healthcare institutions, insurance companies, and hospitals can enhance stroke risk assessment and management. By integrating the predictive model into electronic health record (EHR) systems or dedicated healthcare software platforms, healthcare providers can leverage the model's insights to identify high-risk patients, tailor treatment plans, and allocate resources more efficiently. Insurance companies can also utilize the model to assess risk profiles accurately and offer personalized insurance plans.

3. Telemedicine and Remote Monitoring:

Incorporating the stroke prediction model into telemedicine platforms and remote monitoring systems enables remote consultations and proactive healthcare management. Patients can use mobile applications or web-based platforms to input relevant health data, and the model can analyze this data to assess stroke risk remotely. Healthcare providers can then intervene as necessary, providing guidance, medication adjustments, or referrals for further evaluation.

4. Continuous Model Improvement:

Continuously refining and updating the predictive model based on new data, feedback, and advancements in machine learning techniques is essential for maintaining its accuracy and effectiveness over time.
Collaborating with medical experts, researchers, and data scientists to gather additional data, incorporate new features, and optimize model

performance ensures that the model remains relevant and reliable in real-world healthcare settings.

5. Education and Awareness Campaigns:

Conducting education and awareness campaigns to promote stroke prevention, early detection, and intervention is crucial for reducing the burden of stroke-related morbidity and mortality. By disseminating information about stroke risk factors, symptoms, and preventive measures, individuals can make informed lifestyle choices and seek medical attention promptly if needed. Integrating the stroke prediction model into educational materials and public health initiatives can amplify its impact and empower communities to prioritize stroke prevention and management.

By pursuing these deployment and future directions, the stroke prediction model can have a transformative impact on healthcare delivery, enabling proactive risk assessment, personalized interventions, and improved outcomes for individuals at risk of stroke.

Conclusion

The project aimed to rigorously assess the performance of various machine learning models on a specific dataset, seeking insights into their predictive capabilities. Across models like Logistic Regression, Support Vector Machine, K-Nearest Neighbors, AdaBoost, and Neural Network, a consistent performance emerged, with accuracies averaging around 93-94%. Notably, ensemble methods such as Random Forest, Gradient Boosting Machines, and AdaBoost showcased a slight edge over individual models like Decision Tree and Naive Bayes, highlighting the effectiveness of collective wisdom in enhancing predictive accuracy. However, caution is warranted due to potential overfitting detected in certain models, particularly Decision Tree and Naive Bayes, suggesting a need for further evaluation beyond simple accuracy metrics. Metrics like precision, recall, and F1-score provide a more nuanced understanding of a model's generalization performance beyond the training data. These findings underscore the iterative nature of model evaluation, emphasizing the importance of continual refinement and the adoption of ensemble techniques to unlock new frontiers in predictive modeling. In navigating the complexities of model selection and evaluation, practitioners are guided by a commitment to rigor and a quest for insights, paving the way for transformative advancements in the field of machine learning.

Bibliography

1. World Health Organization. (2020). Stroke. Retrieved from https://www.emro.who.int/health-topics/stroke-cerebrovascular-accident/index.html

2. Kaggle. (n.d.). Stroke Prediction Dataset. Retrieved from https://www.kaggle.com/datasets/zzettrkalpakbal/full-filled-brain-stroke-dataset

3. www.google.com

4. Learn about Stroke. [(accessed on 25 May 2022)]. Available online: https://www.world-stroke.org/world-stroke-day-campaign/why-stroke-matters/learn-about-stroke

5. 2. Elloker T., Rhoda A.J. The relationship between social support and participation in stroke: A systematic review. *Afr. J. Disabil.* 2018;**7**:1–9. doi: 10.4102/ajod.v7i0.357. [PMC free article] [PubMed] [CrossRef] [Google Scholar]

6. 3. Katan M., Luft A. *Seminars in Neurology.* Volume 38. Thieme Medical Publishers; New York, NY, USA: 2018. Global burden of stroke; pp. 208–211. [PubMed] [Google Scholar]

7. 4. Bustamante A., Penalba A., Orset C., Azurmendi L., Llombart V., Simats A., Pecharroman E., Ventura O., Ribó M., Vivien D., et al. Blood biomarkers to differentiate ischemic and hemorrhagic strokes. *Neurology.* 2021;**96**:e1928–e1939. doi: 10.1212/WNL.0000000000011742. [PubMed] [CrossRef] [Google Scholar]

8. 5. Xia X., Yue W., Chao B., Li M., Cao L., Wang L., Shen Y., Li X. Prevalence and risk factors of stroke in the elderly in Northern China: Data from the National Stroke Screening Survey. *J. Neurol.* 2019;**266**:1449–1458. doi: 10.1007/s00415-019-09281-5. [PMC free article] [PubMed] [CrossRef] [Google Scholar]