CAPSTONE PROJECT

AI-Based Stroke Risk Prediction Using Machine Learning

PRESENTED BY

STUDENT NAME: BENEDICT CHACKO MATHEW

COLLEGE NAME: VISWAJYOTHI COLLEGE OF ENGINEERING AND TECHNOLOGY , VAZHAKULAM

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING

EMAIL ID: BENEDICTCM1@GMAIL.COM

AICTE STUDENT ID: STU681b5eb7b6b1c1746624183

OUTLINE

- Problem Statement
- Proposed System/Solution
- System Development Approach
- Algorithm & Deployment
- Result
- Conclusion
- Future Scope
- References

PROBLEM STATEMENT

Stroke stands as one of the most devastating medical emergencies—often striking without warning and leaving behind a trail of permanent disability or death. Every second matters, yet many high-risk patients remain undetected until it's too late, simply because early warning signs go unnoticed in traditional systems.

With the explosion of health data in the digital age, there's a transformative opportunity: using **machine learning to predict stroke risk before it happens**. By analyzing key health indicators like **age, hypertension, heart disease, blood glucose levels, and BMI**, we can build smart systems that flag potential stroke cases in advance—enabling doctors to take action when it matters most.

PROPOSED SOLUTION

This project introduces an intelligent machine learning–based classification system designed to assess a patient’s medical profile and accurately predict their risk of experiencing a stroke.

Key Features:

- **Dataset:**

Stroke Prediction Dataset (Kaggle)

- **Preprocessing:**

Missing value handling, one-hot encoding

- **Algorithms:**

Logistic Regression & Random Forest

- **Evaluation:**

Accuracy, precision, recall, F1-score, confusion matrix

Result:

Model	**Accuracy**
Logistic Regression	~95%
Random Forest	~95%

SYSTEM APPROACH

Tools Used:

- Python 3 (Jupyter Notebook, Google Colab)
- pandas, numpy, seaborn, matplotlib
- sklearn.model_selection
- RandomForestClassifier, LogisticRegression
- classification_report, accuracy_score, confusion_matrix

- **Steps:**
- Load and explore the data
- Handle missing values and encode categories
- Split data into train sets
- Train models and evaluate performance
- Compare results using classification metrics

ALGORITHM & DEPLOYMENT

Algorithm Selection:

To tackle the binary classification problem of stroke prediction, we selected two core supervised learning algorithms: **Logistic Regression** and **Random Forest Classifier**.

- **Logistic Regression** was chosen as a baseline model due to its simplicity, interpretability, and effectiveness on linearly separable data. It provides a solid benchmark for comparing more complex models and offers direct probability outputs for risk interpretation.
- **Random Forest Classifier** was selected for its ability to model complex, non-linear relationships between health variables and stroke risk. It is robust to overfitting, handles both numerical and categorical data efficiently, and naturally ranks feature importance—making it ideal for medical datasets with mixed types of input.

Both models were trained on features such as age, hypertension, heart disease, average glucose level, and BMI—selected based on medical relevance to stroke risk.

ALGORITHM & DEPLOYMENT

Data Input:

- The machine learning models were trained on a structured medical dataset containing multiple health-related attributes of patients. These features serve as predictors for determining the likelihood of a stroke.

Key Input Features Used:

- **Age**: Stroke risk increases significantly with age.
- **Hypertension:** Binary indicator of high blood pressure.
- **Heart Disease**: Binary indicator of pre-existing heart conditions.
- **Average Glucose Level**: Continuous measure; high glucose levels are linked to stroke risk.
- **Body Mass Index (BMI)**: Measures obesity, which correlates with cardiovascular issues.
- **Gender**: Male/female/other—certain trends in stroke risk vary with gender.
- **Smoking Status**: Categorized as 'smokes', 'formerly smoked', 'never smoked', or 'unknown’.

These features were selected based on clinical relevance and exploratory data analysis. Categorical variables were **encoded numerically**, and missing values (especially in BMI) were **imputed** before training.

ALGORITHM & DEPLOYMENT

Training Process:

- The classification models were trained on a labeled dataset containing historical health data of patients, including whether or not they had experienced a stroke. The process involved several critical steps to ensure effective learning and generalization:

Train-Test Split

The dataset was divided into:

- **Training Set** (80%) — used to fit the model
- **Test Set** (20%) — used to evaluate performance on unseen data

Handling Class Imbalance

Stroke cases represented a **small minority of the dataset**, creating an imbalanced classification problem. To address this:

- The stroke class distribution was analyzed
- The model's performance was evaluated using **precision, recall**, and **F1-score**, not just accuracy

ALGORITHM & DEPLOYMENT

Prediction Process:

- Once trained, the machine learning models can predict the **likelihood of stroke** for new or unseen patient data by analyzing their medical features.

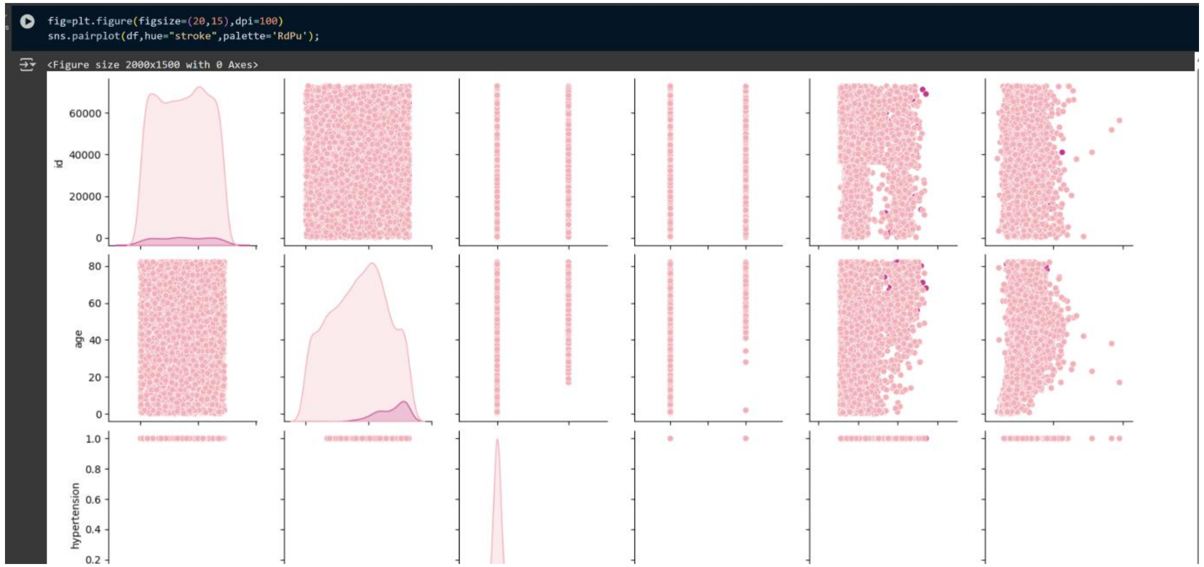
How the Prediction Works:

- A new patient's input data (age, glucose level, hypertension status, etc.) is passed to the model.
- The trained algorithm evaluates these features using learned patterns and decision boundaries from the training phase.
- The model outputs:
  - 0 → No stroke risk
  - 1 → High risk of stroke
  - Some models also output a **probability score** (e.g., 0.85), indicating confidence.

RESULT

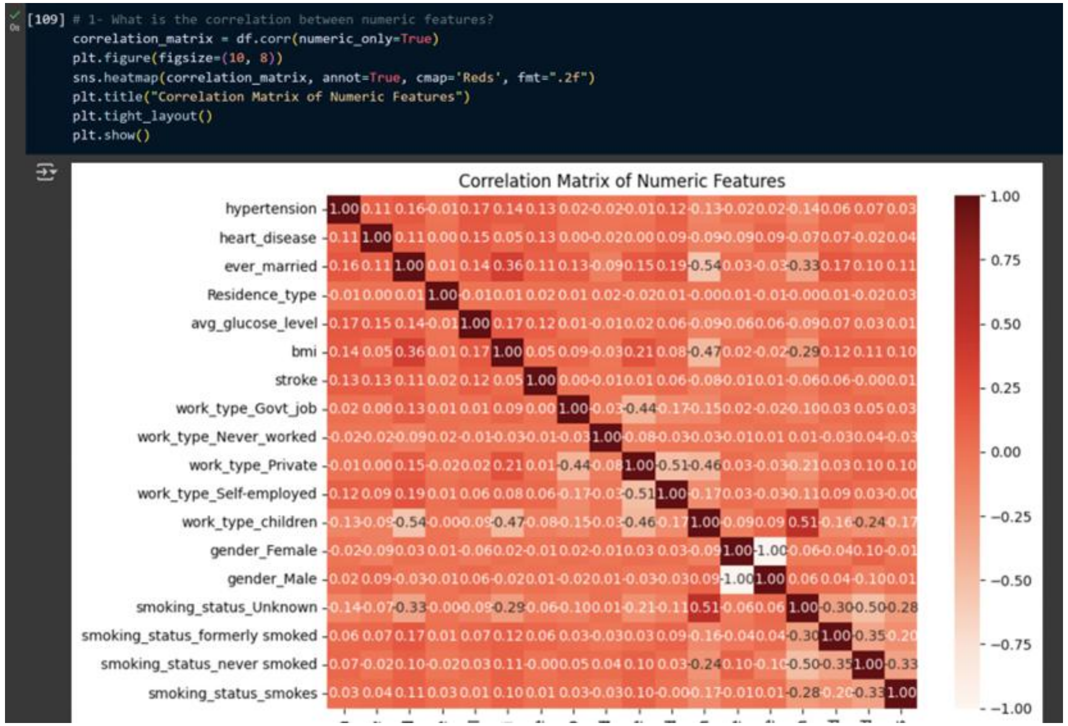
The performance of the machine learning models was evaluated using a test set, focusing on metrics such as **accuracy**, **precision**, **recall**, and **F1-score**. Both **Logistic Regression** and **Random Forest** demonstrated strong predictive capabilities, with Both Logistic Regression and Random Forest showing equally the best performance .

PROJECT LINK



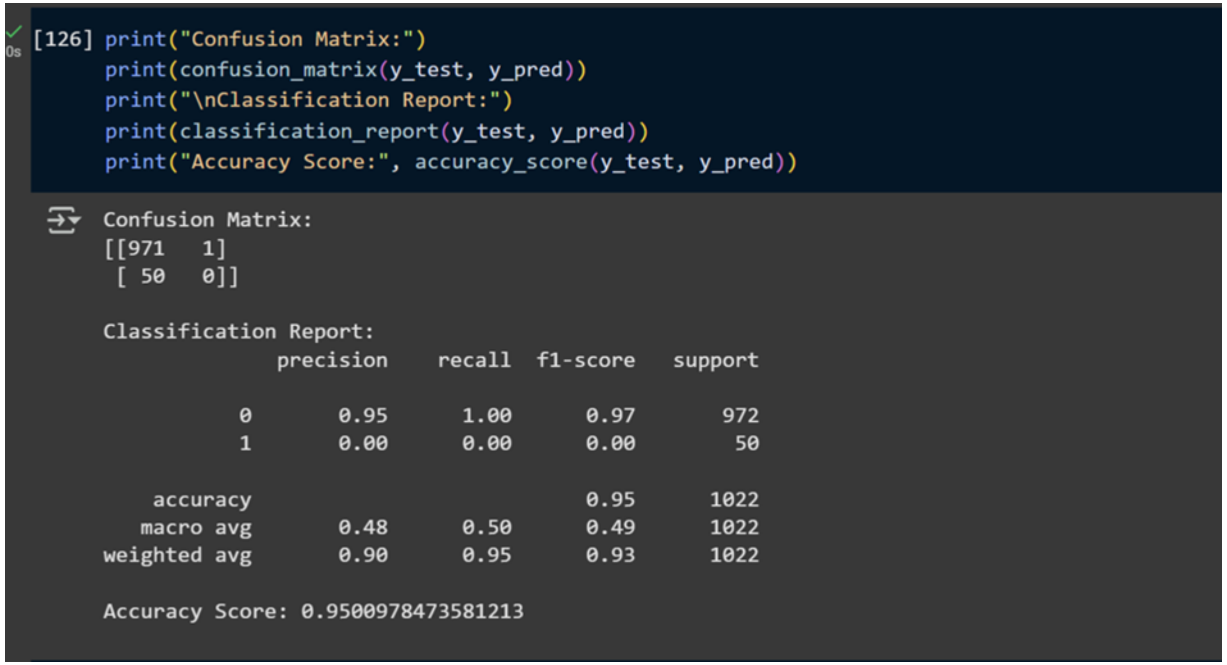
```
fig=plt.figure(figsize=(20,15),dpi=100)
sns.pairplot(df,hue="stroke",palette='RdPu');
```

```
<Figure size 2000x1500 with 0 Axes>
```



```
[109] # 1- What is the correlation between numeric features?
      correlation_matrix = df.corr(numeric_only=True)
      plt.figure(figsize=(10, 8))
      sns.heatmap(correlation_matrix, annot=True, cmap='Reds', fmt=".2f")
      plt.title("Correlation Matrix of Numeric Features")
      plt.tight_layout()
      plt.show()
```

RESULT



```
[126] print("Confusion Matrix:")
      print(confusion_matrix(y_test, y_pred))
      print("\nClassification Report:")
      print(classification_report(y_test, y_pred))
      print("Accuracy Score:", accuracy_score(y_test, y_pred))
```

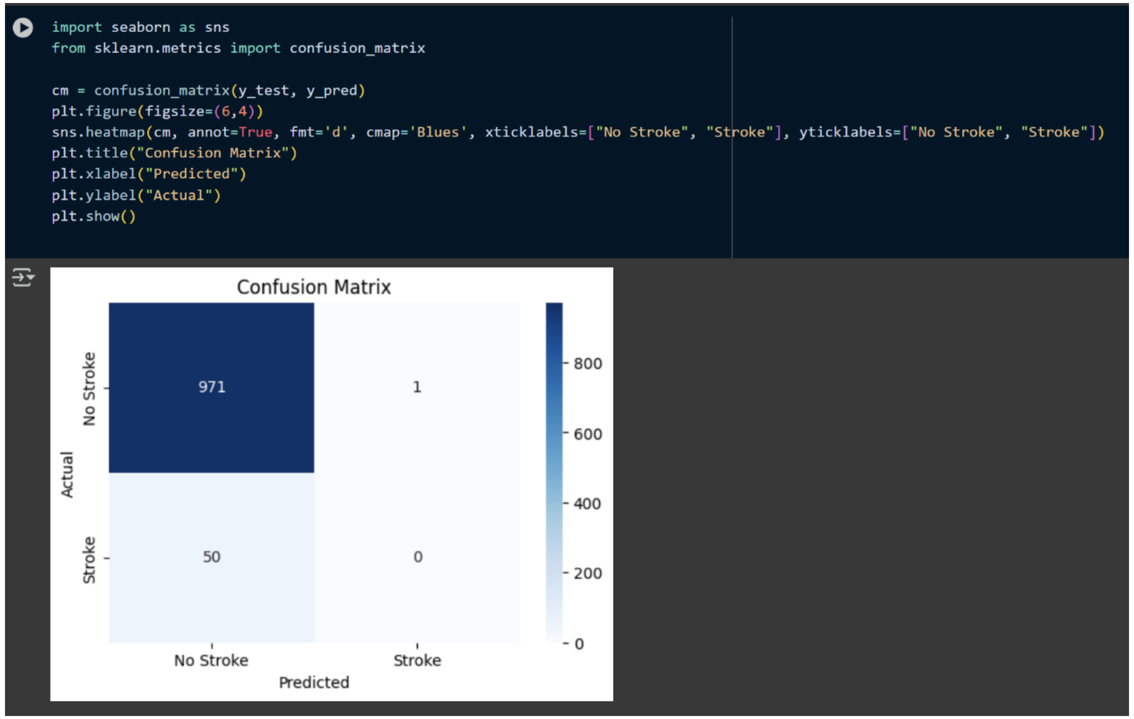
```
Confusion Matrix:
[[971   1]
 [ 50   0]]

Classification Report:
              precision    recall  f1-score   support

           0       0.95      1.00      0.97       972
           1       0.00      0.00      0.00        50

    accuracy                           0.95      1022
   macro avg       0.48      0.50      0.49      1022
weighted avg       0.90      0.95      0.93      1022

Accuracy Score: 0.9500978473581213
```

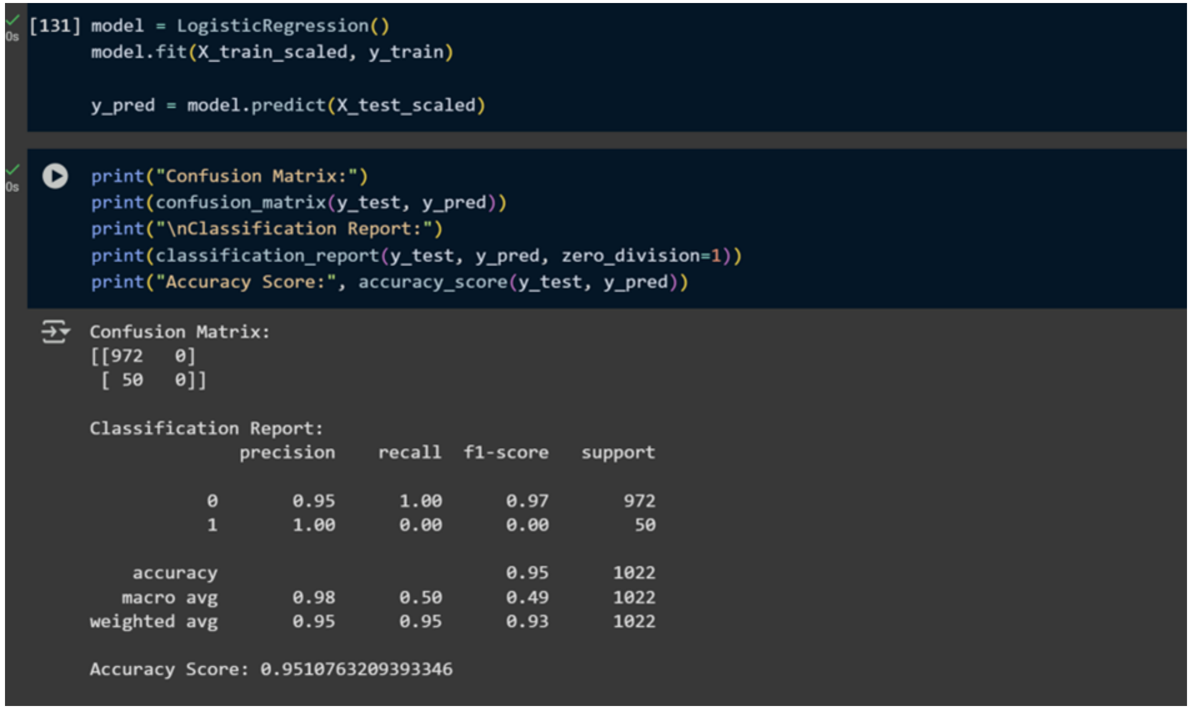


```
import seaborn as sns
from sklearn.metrics import confusion_matrix

cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(6,4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=["No Stroke", "Stroke"], yticklabels=["No Stroke", "Stroke"])
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

CLASSIFICATION REPORT FOR RANDOM FOREST

RESULT



```
[131] model = LogisticRegression()
      model.fit(X_train_scaled, y_train)

      y_pred = model.predict(X_test_scaled)
```

```
print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
print("\nClassification Report:")
print(classification_report(y_test, y_pred, zero_division=1))
print("Accuracy Score:", accuracy_score(y_test, y_pred))
```

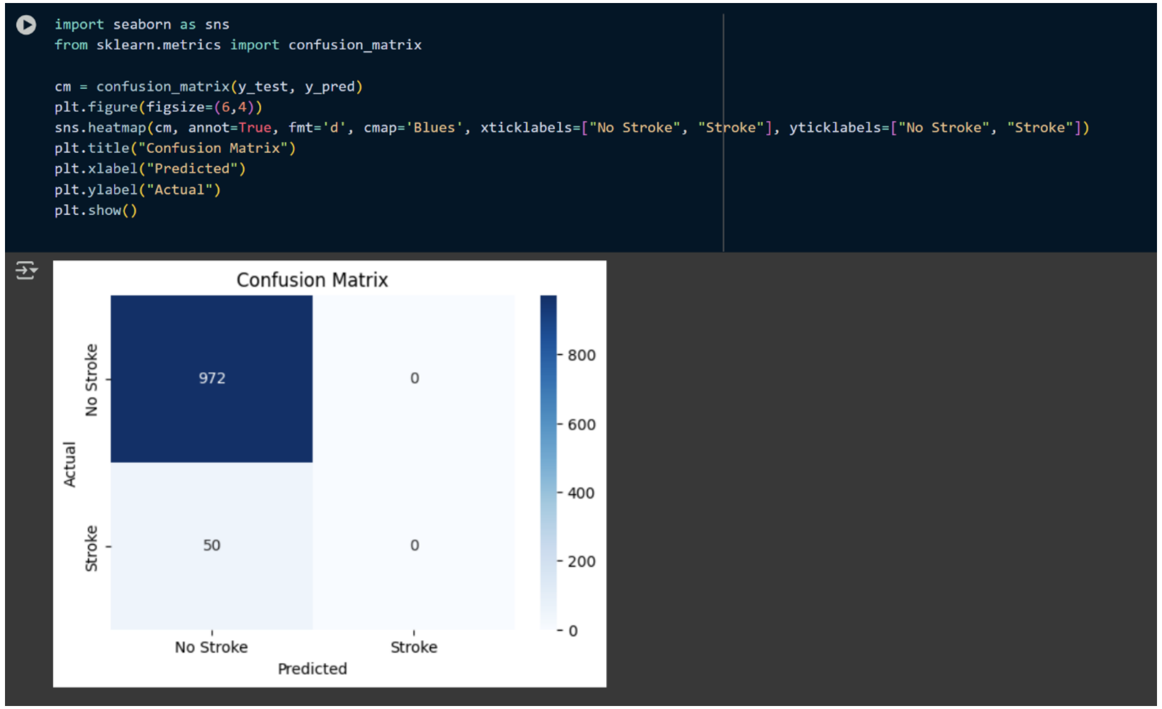
```
Confusion Matrix:
[[972   0]
 [ 50   0]]

Classification Report:
              precision    recall  f1-score   support

           0       0.95      1.00      0.97       972
           1       1.00      0.00      0.00        50

    accuracy                           0.95      1022
   macro avg       0.98      0.50      0.49      1022
weighted avg       0.95      0.95      0.93      1022

Accuracy Score: 0.9510763209393346
```

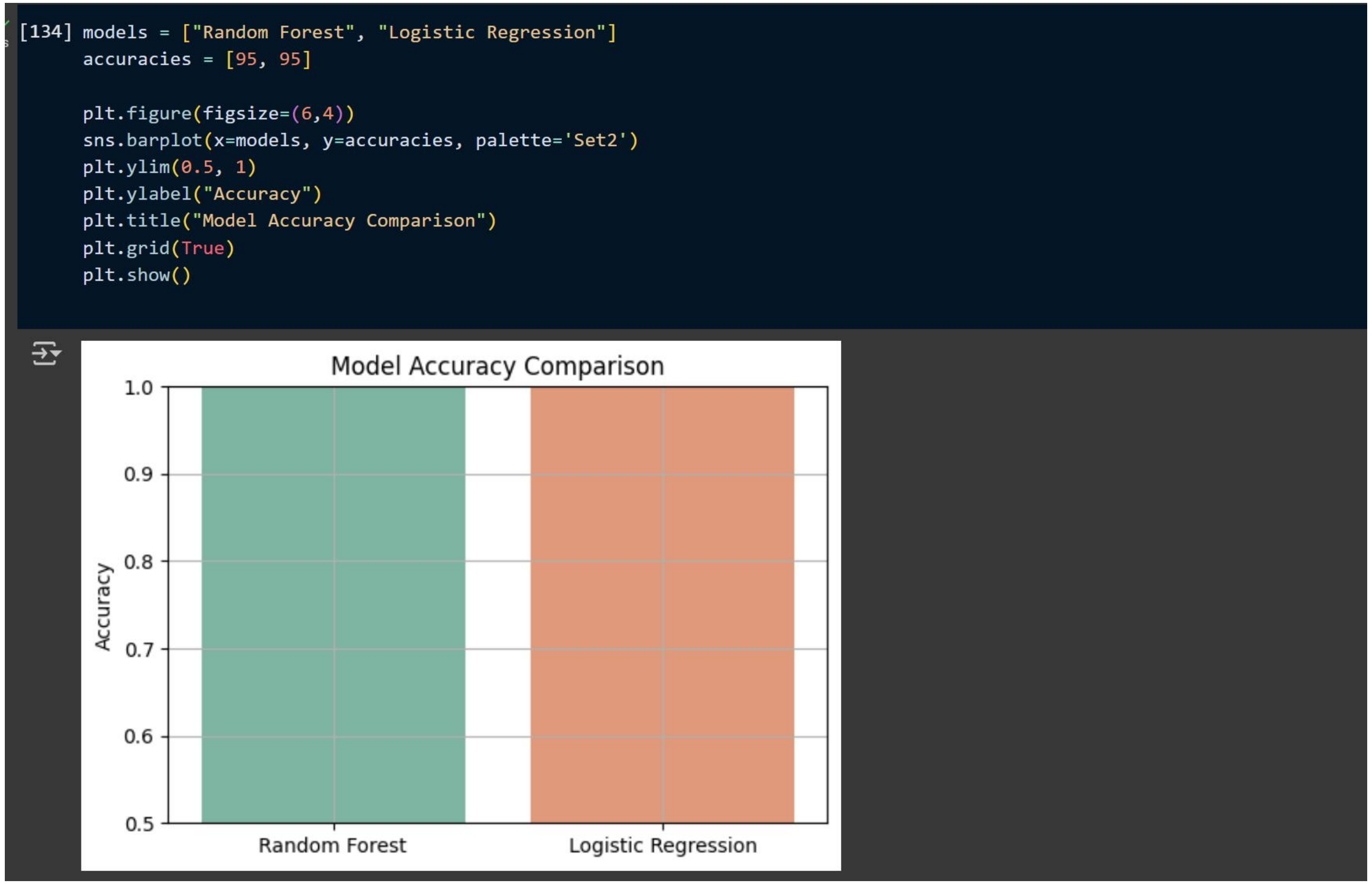


```
import seaborn as sns
from sklearn.metrics import confusion_matrix

cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(6,4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=["No Stroke", "Stroke"], yticklabels=["No Stroke", "Stroke"])
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

CLASSIFICATION REPORT FOR LOGISTIC REGRESSION MODEL

RESULT



```
[134] models = ["Random Forest", "Logistic Regression"]
      accuracies = [95, 95]

      plt.figure(figsize=(6,4))
      sns.barplot(x=models, y=accuracies, palette='Set2')
      plt.ylim(0.5, 1)
      plt.ylabel("Accuracy")
      plt.title("Model Accuracy Comparison")
      plt.grid(True)
      plt.show()
```

MODEL ACCURACY COMPARISION

Model	Accuracy
Logistic Regression	95.1%
Random Forest	95.0%

CONCLUSION

This project demonstrated the effectiveness of machine learning in predicting stroke risk using patient health data. By training models like **Random Forest** and **Logistic Regression**, the system was able to identify patterns that correlate strongly with stroke occurrence, such as age, hypertension, and glucose levels.

Going forward, the model can be enhanced with **real-time data integration**, **feature importance visualization**, and deployment as a clinical tool. Such AI-driven systems can play a vital role in **preventive healthcare**, enabling early intervention and reducing stroke-related fatalities.

FUTURE SCOPE

- In the future, the system can be enhanced by incorporating **larger and more diverse datasets** from hospitals, wearable devices, and real-time electronic health records. This would improve generalizability across different populations and healthcare environments.
- The model's performance can be further improved by applying **advanced algorithms** like **XGBoost, LightGBM**, or even **deep learning** architectures. Additionally, **automated hyperparameter tuning** and **model interpretability tools** can help optimize accuracy while maintaining clinical trust.
- To enable real-world deployment, the solution could be integrated with **edge computing devices** for on-site predictions in remote clinics or mobile health units. Expanding this system into a **web or mobile-based platform** would make it accessible to healthcare workers in underserved regions, ultimately contributing to early intervention and reduced stroke mortality.

REFERENCES

Project GitHub Link: LINK (Uploaded By Myself)

Stroke Prediction Dataset, Kaggle : LINK

Predictive modelling and identification of key risk factors for stroke using machine learning : LINK

Predicting stroke risk: An effective stroke prediction model based on neural networks : LINK

Stroke Prediction using Machine Learning Methods : LINK

Predicting Risk of Stroke From Lab Tests Using Machine Learning Algorithms: Development and Evaluation of Prediction Models : LINK

Thank you