USA SASE Hackathon

Introduction

Stroke is a medical condition where poor blood flow to the brain results in cell death. It is a leading cause of disability and death worldwide. Early detection and prevention are crucial to reducing the adverse outcomes of stroke. Your task is to build a machine learning model that predicts the occurrence of stroke based on various health and demographic factors. The model should help in identifying high-risk individuals, which can aid in early intervention and personalized healthcare planning.

- Develop a predictive model using machine learning to assess stroke risk.
- Analyze risk factors such as age, BMI, glucose levels, smoking status, and hypertension.
- Create a dashboard or visualization tool to help medical professionals interpret results.
- Suggest actionable interventions for at-risk individuals based on predictions.
- Use feature importance analysis to explain which factors contribute most to stroke risk.

Link to the CSV dataset can be found and downloaded here:
https://tinyurl.com/yp7u8rj6

Attribute Information
1) id: unique identifier
2) gender: "Male", "Female" or "Other"
3) age: age of the patient
4) hypertension: 0 if the patient doesn't have hypertension, 1 if the patient has hypertension
5) heart_disease: 0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease
6) ever_married: "No" or "Yes"
7) work_type: "children", "Govt_jov", "Never_worked", "Private" or "Self-employed"
8) Residence_type: "Rural" or "Urban"
9) avg_glucose_level: average glucose level in blood
10) bmi: body mass index
11) smoking_status: "formerly smoked", "never smoked", "smokes" or "Unknown"*
12) stroke: 1 if the patient had a stroke or 0 if not
*Note: "Unknown" in smoking_status means that the information is unavailable for this patient

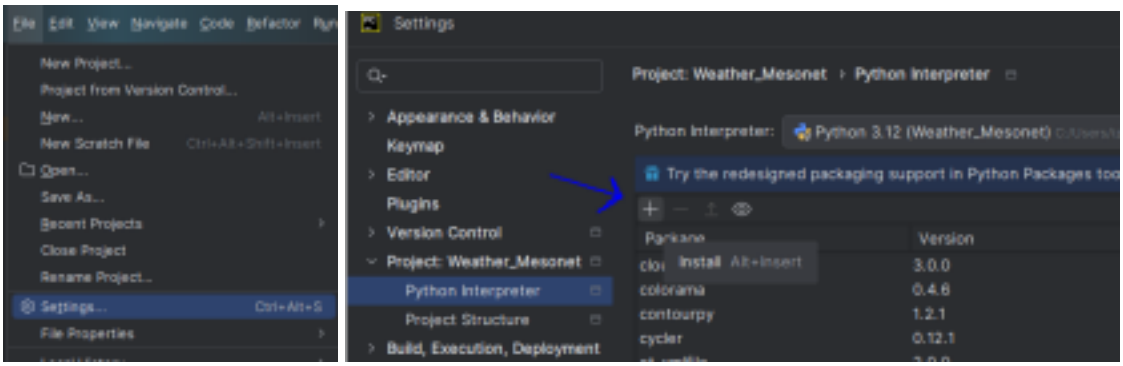
Environment Setup
If you have your own preferred IDE you'd like to use, you're able to use it

1) Download and install Python 3.9+ from https://www.python.org/downloads/. (If you already have Python installed, you can skip this step.
2) Install PyCharm
3) Download PyCharm **Community Edition** from https://www.jetbrains.com/pycharm/download/. Make sure you are installing the Community edition that can be found on the bottom of the page or else you'll be asked to pay for a license. 4) Install it and open PyCharm.
5) Open PyCharm → Click on New Project.
6) Name your project of your own choice.
7) Choose Virtual Environment → Ensure venv is selected.
8) Select Python Interpreter → Use the installed Python version.
9) Click Create

Import Setup

1) Inside PyCharm, create a new file: right-click ML_Demo → New → Python File → Name it anything of your choice
2) Install Required Packages: Install *pandas, numpy, scikit-learn, seaborn*, *imbalanced-learn*, *shap*, and *matplotlib, (*More packages may be needed to be added in depending on the approach you take in coding)

3) You can download these packages in PyCharm by clicking File -> Project: -> Python Interpreter -> Clicking the "+" sign and installing the packages from there.



Challenge #1 Data Preprocessing & Initial Model Training
1) Load and explore the dataset. Handle missing values in the BMI column.
3) Normalize numerical features (age, glucose level, BMI) for better model performance.
4) Use train_test_split (stratified) from sklearn.model_selection to create training and test sets (80-20 split). 5)
Train a baseline classification model (Logistic Regression) and evaluate using accuracy, F1-score, and precision

Challenge #2: Handling Imbalanced Data & Training More Models
1) Check the class distribution of the target variable (stroke) to see if it is imbalanced.
2) Apply SMOTE (Synthetic Minority Over-sampling Technique) to balance the training set. SMOTE helps by generating synthetic data points for the minority class.
3) Train a DecisionTreeClassifier and evaluate using precision, recall, and F1-score.
3) Train a RandomForestClassifier and evaluate using precision, recall, and F1-score.
3) Train a XGBClassifier and evaluate using precision, recall, and F1-score.
4) Compare the performances between these classifiers.

Challenge #3 Visual Implementation
1) Implement SHAP: Use the shap library to explain model predictions, identifying key health and demographic factors contributing to stroke predictions.
2) Implement Feature Correlation Heatmap: Use Seaborn to visualize correlations between different health factors and stroke occurrences.

Challenge #4 Presentation
Create a PowerPoint summarizing the most important factors influencing stroke predictions. Present findings with visualizations and feature importance analysis. Suggest real-world applications, such as early intervention, personalized healthcare, and public health strategies.

Bonus Challenge
Wanna do more and above for more points? Implement additional advanced machine learning techniques and visual models.

Judging Criteria
Model Performance (40%) – Accuracy and overall effectiveness.
Code Quality (20%) – Readability, modularity, and documentation.
Data Insights (10%) – Quality of feature engineering, and SHAP analysis.
Questions (10%) – Ability to answer questions
Presentation (20%) – Clear explanation of findings and approach.