Stroke Prediction Model

This repository contains code for a stroke prediction model based on the "Stroke Prediction" Dataset, it includes various features related to individuals health and lifestyle; whether they have experienced a stroke or not. our aim is to develop a model that can predict the likelihood of a stroke based on these features. It contains a number of features such as pertaining to people's health, lifestyle choices, and stroke history. The objective is to create a model that uses these characteristics to forecast the chance of a stroke.

Dataset

https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset

Model map

https://excalidraw.com/#json=iKSVwVE4PLkk6UUCCI43,ZlZiiqib_mDHmrRUDv2wvA

Context

According to the World Health Organization stroke is the second leading cause of death in the world, responsible for approximately 11% of deaths. This dataset is used to predict whether a patient is likely to get stroke based on the input parameters like gender, age, various diseases, and smoking status.

Libraries

The following libraries are required to run the code in this repository:

- Python
- Pandas
- Seaborn
- Scipy
- Numpy
- Matplotlib
- Scikit-learn
- Imbalanced-learn
- Xgboost
- Lightgbm
- Catboost

Code Structure

The code is organized into the following sections:

1. **Data Preprocessing**: In this step, the code reads and combines the training and testing data, handles missing values, encodes categorical variables, and adds additional features.
2. **Feature Selection**: The importance of features is determined using a Random Forest Classifier. The code selects the top-k features for further analysis.
3. **Model Training**: Two models, Logistic Regression and XGBoost, are trained on the selected features. The models are evaluated on the validation set using various performance metrics, including accuracy, precision, recall, F1-score, ROC AUC score, and PR AUC score.
4. **Model Evaluation**: The trained models are evaluated on the validation set, and the code displays their performance metrics.
5. **Model Testing**: The final trained models, Logistic Regression and XGBoost, are used to make predictions on the test data. The code stores the predictions in a CSV file named "results_ronaldinho.csv".
6. **Ensemble Model**: An ensemble model is implemented using a voting classifier. It combines the trained Logistic Regression, XGBoost, and LightGBM models. The code evaluates and displays the performance of the ensemble model.
7. **CatBoost Classifier**: The code trains and evaluates a CatBoost Classifier on the validation set.

Results

METRIC	SCORE
Accuracy	0.8222
Precision	0.9562
Recall	0.7628
F1-score	0.8745
ROC AUC score	0.8769
PR AUC score	0.1875
Kaggle Submission	0.814