Cancer Dataset Analysis

This project analyzes a cancer dataset to explore features and build models that aid in predicting cancer diagnoses. It includes data preprocessing, feature extraction, model training, and evaluation.

Table of Contents

Introduction

This project leverages a cancer dataset to develop and evaluate predictive models. The goal is to identify important features associated with cancer diagnoses and to build a model capable of distinguishing between cancer and healthy samples. The notebook explores the data, performs preprocessing, and applies machine and deep learning models to achieve high AUC in cancer prediction.

Google Colab Link

You can run the notebook directly in Google Colab using the following link: https://colab.research.google.com/drive/1gj_ju9XR4ikiv9nLmTAII38QxP-iJcXx?usp=sharing

Requirements

The following libraries are required to run the notebook, and they are typically available in Google Colab:

- - **Python** 3.10
- - **Libraries**: `numpy`, `pandas`, `scikit-learn`, `matplotlib`, `seaborn`, `tensorflow`

Additional libraries can be installed within the notebook using `!pip install`.

Usage

Running the Notebook

1. **Open the Notebook in Colab** - Click on the Colab link above to open the notebook in Google Colab.
2. **Run Cells Sequentially** - Run each cell in sequence. Some cells may prompt for data upload or require specific configurations, depending on the dataset and tasks.

Data Uploads

If the notebook requires data files:
- You may upload the cancer dataset directly to Colab, or
- Use Google Drive to load large or persistent files:

```python
from google.colab import drive
drive.mount('/content/drive')
```

Notebook Overview

This notebook is organized into the following main sections:

Data Loading:

- Loads the cancer dataset, typically from a CSV file or a similar format.
- Provides an initial preview of the data, examining its shape, feature names, and the first few rows to understand its structure.

Data Preprocessing:

- Standardizes or normalizes features as needed for optimal model performance.
- Class Imbalance Handling
- Exploratory Data Analysis (EDA):
    - Visualizes the distribution of key features, using histograms, scatter plots, and box plots to highlight potential patterns.
    - Examine correlations between features and the target variable to identify important predictors.

Feature Extraction

- Feature Selection using GBM
- Performs dimensionality reduction or feature selection methods to retain only the most relevant features using PCA and LDA

Model Training:

- Trains multiple machine learning and deep learning models (e.g., logistic regression, support vector machine, neural networks) on the preprocessed data.
- Configures model parameters and applies validation to assess stability and accuracy.

Model Evaluation:

- Evaluates models on the test data using metrics such as accuracy, precision, recall, F1-score, and AUC-ROC.
- Visualizes model performance with confusion matrices and ROC curves to help assess classification effectiveness.

Predictions and Interpretation:

- Make predictions on test data.

Hyperparameter Tuning

- Optimizes model parameters (manual hyperparameter tuning for SVM and MLP and RandomizedSearchCV for Logistic Regression and Voting Classifier) to achieve the best possible performance.