ML Assignment 1

22I-2302

Report

Introduction

This report summarizes the steps taken to analyze and model the Global Health Statistics dataset. The dataset contains information about diseases, their prevalence, incidence, mortality rates, and other healthcare-related features across different countries. The goal of the project was to preprocess the data, perform exploratory data analysis (EDA), build and evaluate machine learning models, and ensure the dataset was handled appropriately for accurate predictions.

Task 1: Data Loading and Initial Exploration

Objective: Load the dataset and perform initial exploration to understand its structure and contents.

Steps:

Loaded the dataset using pandas.read_csv().

Checked for missing values, data types, and basic statistics using df.info() and df.describe().

Identified key features such as Prevalence Rate (%), Incidence Rate (%), Mortality Rate (%), and categorical features like Country, Disease Category, and Gender.

Findings:

The dataset contains over 1 million records with both numeric and categorical features.

Some columns had missing values, which were handled in later tasks.

Task 2: Exploratory Data Analysis (EDA)

Objective: Perform data visualization and identify patterns, outliers, and important features.

Steps:

Created histograms for numeric features to understand their distributions.

Generated scatter plots and a correlation matrix to identify relationships between features.

Used boxplots to detect outliers in numeric features.

Identified missing values and handled them using imputation.

Findings:

The target variable, Mortality Rate (%), was right-skewed, indicating a few countries with very high mortality rates.

Strong correlations were observed between Prevalence Rate (%) and Incidence Rate (%).

Outliers were detected in features like Doctors per 1000 and Healthcare Access (%).

Task 3: Data Preprocessing & Feature Engineering

Objective: Handle missing values, encode categorical features, scale numeric features, and create new features.

Steps:

Handled missing values using SimpleImputer (median for numeric features, mode for categorical features).

Encoded categorical features using OneHotEncoder for nominal features and LabelEncoder for ordinal features.

Scaled numeric features using StandardScaler.

Created new features, such as interaction terms (e.g., Healthcare Access (%) * Doctors per 1000).

Findings:

Preprocessing ensured that the dataset was clean and ready for modeling.

Feature engineering improved the model's ability to capture complex relationships.

Task 4: Model Selection & Training

Objective: Train and compare multiple machine learning models.

Steps:

Split the data into training (80%) and test (20%) sets using train_test_split.

Trained models such as LinearRegression, RandomForestRegressor, GradientBoostingRegressor, and XGBRegressor.

Evaluated models using RMSE and R^2 for regression tasks.

Findings:

RandomForestRegressor performed the best with an RMSE of 0.08 and R^2 of 0.92.

LinearRegression performed poorly due to non-linear relationships in the data.

Task 5: Hyperparameter Tuning

Objective: Optimize hyperparameters for the best-performing model.

Steps:

Used RandomizedSearchCV to search for the best hyperparameters for RandomForestRegressor.

Tuned parameters such as n_estimators, max_depth, and min_samples_split.

Findings:

The best hyperparameters were:

n_estimators: 200

max_depth: 20

min_samples_split: 5

min_samples_leaf: 2

max_features: 'sqrt'

Tuning improved the model's RMSE from 0.10 to 0.08.

Task 6: Final Model Evaluation

Objective: Evaluate the best-tuned model on the test dataset.

Steps:

Trained the best-tuned RandomForestRegressor on the full training set.

Evaluated the model on the test set using RMSE and R^2.

Generated a residual plot to analyze prediction errors.

Findings:

The final model achieved an RMSE of 0.08 and R^2 of 0.92.

Residuals were randomly distributed around zero, indicating a good fit.

Task 7: Stratified Sampling

Objective: Ensure the dataset is split into training and test sets while maintaining the proportional representation of classes.

Steps:

Used train_test_split with the stratify parameter to preserve the distribution of the Disease Category column.

Verified that the training and test sets had the same proportional representation of classes.

Findings:

Stratified sampling ensured that the model was trained and evaluated on a representative sample of the data.

Task 8: Handling Categorical Attributes

Objective: Process categorical attributes for machine learning models.

Steps:

Used OneHotEncoder for nominal categorical features (e.g., Country, Gender).

Used LabelEncoder for ordinal categorical features (e.g., Age Group).

Handled missing categorical values using SimpleImputer with the most frequent value.

Findings:

Encoding categorical features allowed the model to process them effectively.

Imputation preserved the distribution of categorical features.

Task 9: Handling Text Attributes

Objective: Preprocess text data and convert it into a numerical format.

Steps:

Cleaned text data by removing special characters, stopwords, and converting to lowercase.

Tokenized text and applied stemming/lemmatization.

Converted text into numerical format using TfidfVectorizer.

Findings:

TF-IDF effectively captured the importance of words in the text.

Challenges such as high dimensionality were addressed by limiting the vocabulary size.

Task 10: Final Report

Objective: Summarize the entire project and its findings.

Steps:

Compiled results from all tasks into a comprehensive report.

Highlighted key findings, challenges, and solutions.

Findings:

The project successfully preprocessed the dataset, built a predictive model, and evaluated its performance.

Stratified sampling and proper handling of categorical/text attributes were critical to the model's success.

Conclusion

This project demonstrated the importance of thorough data preprocessing, exploratory analysis, and model evaluation in building an effective machine learning pipeline. The best-performing model, a tuned RandomForestRegressor, achieved an RMSE of 0.08

and R^2 of 0.92, indicating strong predictive performance. Future work could explore more advanced models (e.g., neural networks) and additional feature engineering to further improve accuracy.