**DIAGNOSIS OF POLYCYSTIC OVARY SYNDROME**

Problem Statement:

- Polycystic ovarian syndrome (PCOS) is one of the most common reproductive endocrinological disorders with a broad spectrum of clinical manifestations affecting about 6- 8% of women of reproductive years.

- However, it is important to make an early diagnosis in order to prevent early and late sequel of the syndrome.

- PCOS a diagnosis of exclusion has been a topic of debate and many consensus definitions have evolved over time.

- PCOS causes in ovaries inappropriate growth of follicles that are prevented at a primary stage and miscarry to mature.

- This is one of the causes for infertility. Therefore, it is significant to screen the patients at a primary stage to prevent any serious moment of the PCOS disease.

**Proposed System:**

- This project focuses on the data-driven diagnosis of polycystic ovary syndrome (PCOS) in women.

- For this, machine learning algorithms are applied to a dataset freely available in Kaggle repository.

- This dataset has 43 attributes of 541 women, among which 177 are patients of PCOS disease.

- Firstly, univariate feature selection algorithm is applied to find the best features that can predict PCOS.

- The ranking of the attributes is computed and it is found that the most important attribute is the ratio of Follicle-stimulating hormone (FSH) and Luteinizing hormone (LH).

- Next, holdout and cross validation methods are applied to the dataset to separate the training and testing data.

- A number of classifiers such as gradient boosting, random forest, logistic regression, and hybrid random forest and logistic regression (RFLR) are applied to the dataset.

- Results show that the first 10 highest ranked attributed are good enough to predict the PCOS disease.

- Results also demonstrate that RFLR exhibits the best testing accuracy of 91.01% and recall value of 90% using 40-fold cross validation applied to the 10 most important features.

**Implementation:**

To implement this project we are using the following steps:

**Dataset:**

In this experiment, dataset is composed of Polycystic ovary syndrome is a disorder involving infrequent, irregular or prolonged menstrual periods, and often excess male hormone (androgen) levels. The ovaries develop numerous small collections of fluid called follicles and may fail to regularly release eggs dataset contains all physical and clinical parameters to determine PCOS and infertility related issues.

**Feature Selection:**

Feature selection is the procedure of taking some informative and significant features from a huge number of features. This can produce a better pattern characterization of multiple classes. Taking irrelevant features in the data can reduce the accuracy of the classification models. Feature selection can reduce overfitting and improve accuracy. One form of feature selection approach is the filtering-based univariate feature selection method which considers each feature independently that is with regard to the dependent variable. Each feature is scored individually on certain specified criteria and the features are then selected based on the higher scores or higher ranks.

**Model Training:**

For 10 features with 20-fold cross validation, a number of classifiers are used. With this condition, the classification accuracy obtained by k nearest neighbors (kNN), random forest, decision tree, light GBM (LGBM), adaptive boosting (AdaBoost), gradient boosting, logistic regression, multilayer perceptron (MLP), hybrid random forest and logistic regression (RFLR) are 69.87%, 87.83%, 85.18%, 88.58%, 85.96%, 88.75%, 86.52%, 83.22% and 90.01%, respectively. It can be seen that good accuracy scores are obtained by gradient boosting, random forest, logistic regression, and RFLR.

**Metrics used to evaluate our model**

In our project, we have chosen four key evaluation metrics to assess the performance of our regression models: Mean Squared Error (MSE), Mean Absolute Error (MAE), R-Squared Parameter, and Root Mean Squared Error (RMSE). These metrics are particularly pertinent to our problem because they provide comprehensive insights into the accuracy, precision, and goodness-of-fit of our regression models.

**Mean Squared Error (MSE):**

MSE quantifies the average squared disparity between actual and predicted values. We use MSE as it penalizes larger errors more substantially, making it sensitive to outliers, which is important for our dataset as outliers can significantly impact the performance of regression models.

**Mean Absolute Error (MAE):**

MAE measures the average absolute deviation between actual and predicted values. We include MAE as it provides a more balanced assessment of model accuracy compared to MSE, particularly in scenarios where outliers may have less influence on the overall performance.

**R-Squared Parameter:**

R-Squared Parameter indicates the proportion of variance in the dependent variable that is explained by the independent variables. This metric is crucial for understanding how well our regression models capture the variability in the target variable, providing insights into the goodness-of-fit.

**Root Mean Squared Error (RMSE):**

RMSE calculates the square root of the average squared difference between actual and predicted values. Similar to MSE, we use RMSE to assess the accuracy of our regression models, with the advantage of being in the same units as the target variable, facilitating easier interpretation.

We have experimented with various modeling techniques, including Logistic Regression, Random Forest, AdaBoost, Decision Tree, Gradient Boosting, LGBM, and KNN. Each modeling technique has its own trade-offs:

**Logistic Regression:**

 Simple and interpretable, but may not capture complex nonlinear relationships in the data.

**Random Forest:**

Robust to overfitting, handles high-dimensional data well, but may be computationally expensive and prone to overfitting with noisy data.

**Decision Tree:**

Simple and interpretable, but prone to overfitting and may not generalize well to unseen data.

**Gradient Boosting:**

Builds models sequentially, improving upon weaknesses of previous models, but may be computationally expensive and prone to overfitting with large datasets.

**LightGBM (LGBM):**

Optimized for speed and efficiency, handles large datasets well, but may require careful tuning of hyperparameters.

**K-Nearest Neighbors (KNN):**

Simple and intuitive, but computationally expensive and sensitive to the choice of distance metric and number of neighbors.

To prepare our data, we performed standard preprocessing steps such as handling missing values, encoding categorical variables, scaling numerical features, and splitting the data into training and testing sets. Additionally, we conducted feature engineering to create relevant features and address multicollinearity issues.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Parameter/Algorithm | MSE | MAE | R-Squared Parameter | RMSE | Accuracy |
| Logistics Regression | 0.10 | 0.10 | 0.50 | 0.30. | 0.85 |
| Random Forest | 0.10 | 0.10 | 0.60 | 0.25 | 0.90 |
| Decision Tree | 0.18 | 0.18 | 0.35 | 0.38 | 0.85 |
| Gradient Boosting | 0.15 | 0.15 | 0.40 | 0.38 | 0.80 |
| LGBM | 0.10 | 0.10 | 0.60 | 0.25 | 0.95 |
| KNN | 0.30 | 0.30 | -0.25 | 0.50 | 0.65 |
| Voting Classifier | 0.25 | 0.25 | -0.39 | 0.50 | 0.67 |

Upon evaluating our models using the selected metrics, we observed that LGBM achieved the highest accuracy of 95%, with low values for MSE, MAE, and RMSE, indicating superior performance compared to other models. However, the choice of the best model depends on the specific requirements and constraints of the problem, as well as considerations such as interpretability, computational efficiency, and scalability.

**In order to implement this project we have used the below algorithms:**

**Data Preprocessing and Visualization:**

This model processes and visualizes data related to Polycystic Ovary Syndrome (PCOS), focusing on individuals with and without infertility. It handles two distinct datasets: one for PCOS patients with infertility and another for those without, merging them on a common identifier.

The primary functionalities of the model can be outlined as follows:

**Step-1: Data Merging** - Combines two datasets into one while managing overlapping columns and removing unnecessary ones.

**Step-2: Data Cleaning** - Converts string data to numeric where applicable and fills missing values with the median of corresponding columns.

**Step-3: Exploratory Data Analysis** - Generates visualizations such as correlation heatmaps to explore relationships between various features and PCOS.

**Step-4: Statistical Visualization** - Produces plots to examine trends in age, BMI, menstrual cycle length, and other relevant clinical features.

**Step-5: Output**- The cleaned and processed dataset is saved to a CSV file for further analysis or modeling.

This model ensures that the data is not only thoroughly preprocessed but also ready for more detailed analysis or machine learning applications, making it a critical tool in PCOS research.

**K Nearest Neighbors (KNN):**

K-nearest neighbors (KNN) is a type of supervised learning algorithm used for both regression and classification. KNN tries to predict the correct class for the test data by calculating the distance between the test data and all the training points. Then select the K number of points which is closest to the test data. The KNN algorithm calculates the probability of the test data belonging to the classes of ‘K’ training data and class holds the highest probability will be selected. In the case of regression, the value is the mean of the ‘K’ selected training points.

The K-NN working can be explained on the basis of the below algorithm:

-  Step-1: Select the number K of the neighbors

- Step-2: Calculate the Euclidean distance of K number of neighbors.

-  Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

- Step-4: Among these k neighbors, count the number of the data points in each category.

-  Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.

- Step-6: Our model is ready.

**Gradient Boosting:**

Gradient boosting is an ensemble learning technique that builds a strong predictive model by iteratively adding weak learners to minimize the loss function. It belongs to the class of boosting algorithms, where each subsequent model corrects the errors of its predecessors, focusing on the instances that were previously misclassified or had high residuals.

Key Components:

- Weak Learners (Base Models): The weak learners used in gradient boosting are often decision trees, but other types of models can also be employed. Decision trees are typically shallow, which reduces the risk of overfitting and makes them suitable as base models.

- Loss Function: Gradient boosting can handle various loss functions depending on the type of problem being addressed, such as mean squared error (MSE) for regression and log loss (binary cross-entropy) for classification.

- Gradient Descent: The name "gradient boosting" stems from its use of gradient descent optimization to minimize the loss function. In each iteration, the algorithm calculates the gradient of the loss function with respect to the model's prediction and adjusts the parameters of the weak learner to move in the direction that reduces the loss.

- Boosting: Unlike bagging methods like Random Forest, where base learners are trained independently, boosting methods sequentially train weak learners, with each subsequent learner focusing more on the instances that were misclassified or had high residuals in the previous iterations.

**Random Forest:**

Random Forest classifier to predict Polycystic Ovary Syndrome (PCOS) from a dataset. This model is a form of supervised learning used primarily for classification tasks. The model outlines its functionality as follows:

**Step-1: Data Preparation** - The model loads a dataset from a specified path, processes it by dropping unnecessary identifiers, and separates the features (`X`) from the target variable (`y`).

**Step-2: Splitting the Data** - It divides the data into training and testing sets, with a 70-30 split, to prepare for model training and evaluation.

**Step-3: Model Training** - A Random Forest Classifier is instantiated and fitted to the training data.

**Step-4: Prediction and Evaluation** - The model predicts PCOS outcomes using the test data and calculates accuracy, Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared, and Root Mean Squared Error (RMSE) to evaluate model performance.

**Step-5: Output Metrics** - Results, including predictions and evaluation metrics, are saved to CSV files. Additionally, a bar chart visualizing the performance metrics is created and saved.

This structured approach not only predicts PCOS but also quantitatively assesses the model's performance, providing clear metrics for evaluating its accuracy and reliability.

**Decision Tree:**

Decision Tree classifier to predict Polycystic Ovary Syndrome (PCOS) based on various clinical features. This model is part of supervised learning techniques used for classification tasks. Here's a step-by-step summary of the model's operations:

-**Step-1: Data Preparation** - The model imports data from a specified path, processes it by excluding identifiers from the features, and isolates the features (`X`) and the target variable (`y`).

-**Step-2: Splitting the Data** - It divides the dataset into a 70-30 split between training and testing sets to prepare for training and evaluating the model.

-**Step-3: Model Training** - A Decision Tree Classifier is configured with specific criteria and depth parameters, then fitted to the training data.

-**Step-4: Prediction and Evaluation** - After training, the model is used to predict PCOS outcomes on the test set. It then calculates key performance metrics such as accuracy, Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared, and Root Mean Squared Error (RMSE).

-**Step-5: Output Metrics** - It records the predictions and performance metrics in CSV files. Additionally, a bar chart displaying these metrics is generated and saved.

This structured approach ensures that the Decision Tree model is thoroughly evaluated, providing detailed insights into its performance with metrics such as accuracy, MSE, MAE, R-squared, and RMSE. These evaluations are crucial for assessing the model's reliability and effectiveness in predicting PCOS, thereby supporting its potential use in medical diagnostic processes.

**Logistic Regression:**

Logistic regression is a foundational statistical method used for binary classification tasks. Despite its name, logistic regression is a linear model that predicts the probability of a binary outcome based on one or more predictor variables. It's widely used due to its simplicity, interpretability, and effectiveness in many real-world applications.

Key Components:

- Sigmoid Function (Logistic Function): Logistic regression applies the sigmoid function to the linear combination of input features and their corresponding weights. The sigmoid function transforms the output into a probability score between 0 and 1, representing the likelihood of the positive class.

- Linear Model: Logistic regression models the relationship between the independent variables and the logarithm of the odds of the dependent variable. It's called "linear" because the relationship between the input features and the output is linear, even though the output itself is not.

- Maximum Likelihood Estimation: Logistic regression estimates the model parameters (coefficients) using maximum likelihood estimation. The goal is to find the parameter values that maximize the likelihood of observing the given data under the assumed model. This estimation process is typically performed using optimization algorithms such as gradient descent.

- Decision Boundary: The decision boundary of logistic regression is a hyperplane that separates the feature space into regions corresponding to different class labels. In binary classification, the decision boundary is where the predicted probability equals 0.5, and points on one side of the boundary are classified as the positive class, while points on the other side are classified as the negative class.

**LightGBM Algorithm:**

LightGBM is a gradient boosting framework that uses tree-based learning algorithms. It is designed for efficiency, scalability, and improved accuracy. LightGBM is widely used in various machine learning tasks, including classification, regression, and ranking.

Key Components:

- Gradient Boosting Framework: LightGBM is based on the gradient boosting framework, which builds a predictive model by combining multiple weak learners, typically decision trees. It follows a boosting strategy where each new tree is trained to correct the errors of the previous ones.

- Leaf-Wise Tree Growth: LightGBM grows trees leaf-wise rather than level-wise. It chooses the leaf with the maximum delta loss to grow, which leads to a more balanced tree and reduces the number of levels, resulting in faster training.

- Gradient-Based One-Side Sampling: LightGBM uses a gradient-based strategy for data sampling during tree construction. It selects the instances with larger gradients, focusing on the regions where the model performs poorly, which improves the efficiency and accuracy of the training process.

- Histogram-Based Splitting: LightGBM uses histogram-based algorithms to find the best split points for each feature. It discretizes continuous features into discrete bins to speed up the calculation of split points and reduce memory usage.

**Voting Classifier:**

The Voting Classifier is an ensemble learning technique that combines multiple individual models to make predictions. It aggregates the predictions of each base model and outputs the most frequent prediction (in the case of hard voting) or the average predicted probabilities (in the case of soft voting), effectively leveraging the collective wisdom of diverse models.

Key Components:

- Base Models: The Voting Classifier combines the predictions of multiple base models, which can be of different types (e.g., decision trees, support vector machines, logistic regression). These base models can either be trained independently or on different subsets of the data.

- Voting Strategy: The Voting Classifier supports two main voting strategies: hard voting and soft voting.

- Hard Voting: In hard voting, the final prediction is determined by a simple majority vote among the base models.

- Soft Voting: In soft voting, the final prediction is based on the average predicted probabilities from all base models, weighted by their respective confidence levels.

**Conclusion:**

For this, the concepts of feature selection and machine learning algorithms are applied. A dataset of 541 patients obtained from Kaggle repository is used. Our results show that FSH/LH is the most important attribute among the 43 attributes in the dataset. Results indicate that good accuracy with lower computation time can be obtained when the best 10 features are used. A number of different classifiers are used on these 10 features. It is shown in this paper that gradient boosting, random forest, logistic regression and RFLR exhibit good accuracy and recall values. RFLR has the best testing accuracy of 91.01% and recall value of 90% when 40-fold cross validation is used to split the data into testing and training portions.