Alaa Adel Abdelhafez

241106

Title of the project: Enhancing classification accuracy for Pima Indian Diabetes

1- **Abstract**:

This project aims to enhance the accuracy of predicting diabetes [1] using the Pima Indian Diabetes dataset. By employing Extra-Trees-Classifier (ETC) machine learning techniques and feature engineering strategies, the project seeks to address the challenges associated with accurate diabetes prediction, including imbalanced datasets, missing values, and complex feature relationships. The primary objective is to develop robust models that can accurately classify diabetes cases, facilitating early diagnosis and treatment.

2- **Introduction**:

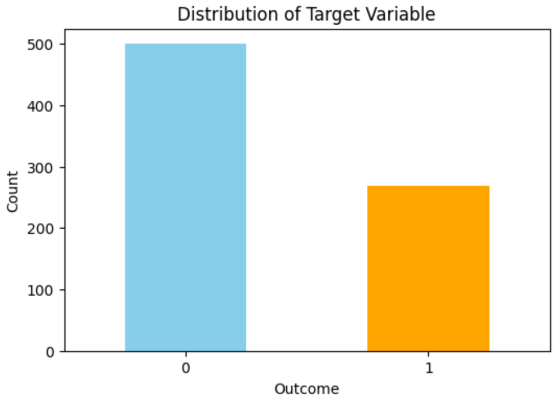
Diabetes is a prevalent and chronic medical condition that requires timely identification for optimal management and prevention of complications. Traditional classification models often struggle with accurately predicting diabetes due to various challenges inherent in the data. This project aims to overcome these challenges by leveraging machine learning algorithms and feature engineering techniques to improve predictive accuracy. The significance of accurate diabetes prediction lies in its potential to enhance healthcare outcomes by enabling early intervention and personalized treatment plans.

3- **Method**:

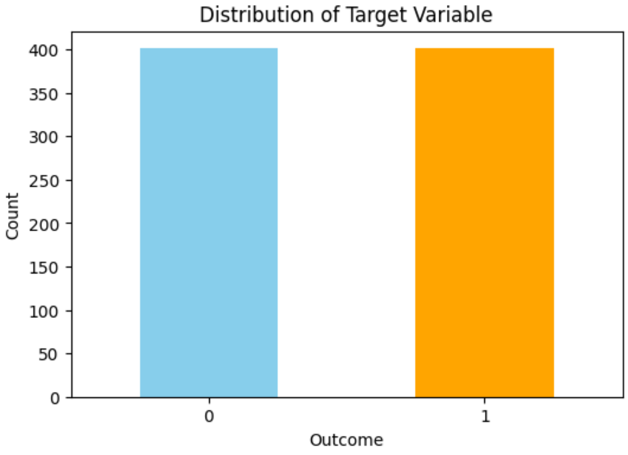
The project employs a systematic approach to enhance classification accuracy for predicting diabetes. It begins by preprocessing the Pima Indian Diabetes dataset, addressing missing values, categorical data, handling imbalance and performing feature engineering to extract relevant information. Next, state-of-the-art machine learning algorithms, such as the Extra Trees Classifier, are explored for their efficacy in handling the dataset complexities. Hyper-parameter tuning is conducted to optimize model performance, with a focus on maximizing recall, a critical metric in the healthcare field. The developed models are then rigorously evaluated using various performance metrics, including accuracy, precision, and recall. Validation against an independent dataset ensures robust generalization of the models.

4- **Experiments Guidelines**:

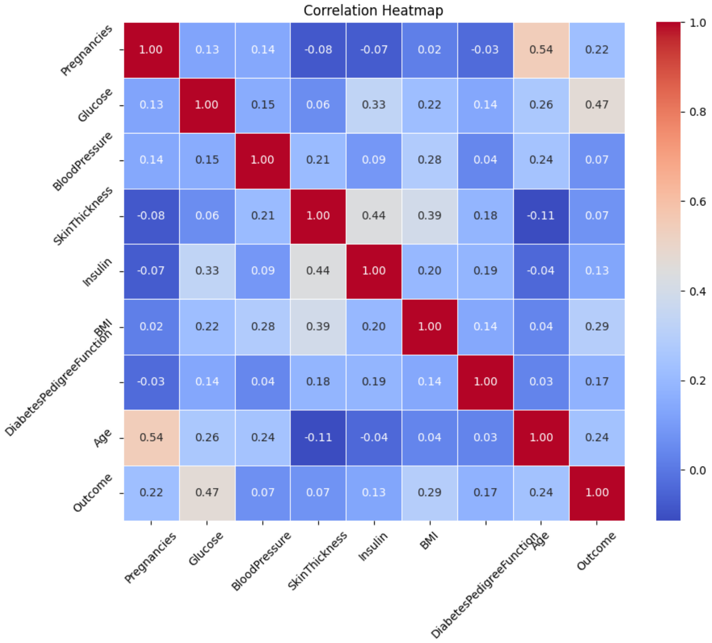
1. **Data Preprocessing**: Address missing values, categorical data, handling imbalance and perform feature engineering, and ensure data is ready for model training
   The most critical thing with this dataset is the labels imbalance:



We preferred oversampling because:
The dataset is small: duplicating or generating synthetic samples through oversampling techniques like SMOTE can help increase the amount of training data, potentially leading to better generalization. The minority class here is 1 class which is that this person has diabetes and this contains valuable information for out model in our case.
And this is the balancing after SMOTE:



Also, we aimed to minimize the features number to better prediction results, so we plotted the heatmap:



We can find that: Glucose, BMI, Age, DiabetesPedigreeFunction
Are the most related to the ‘Outcome’ which is our label,
We used Random Forest as our feature selection algorithm because:
Random Forest (RF) is a popular and effective technique for feature selection due to several reasons:

1. Embedded Feature Selection: RF inherently provides feature importances as part of its training process. As the RF algorithm constructs multiple decision trees based on random subsets of features, it calculates the importance of each feature based on how much it contributes to improving the model's

performance. This embedded feature selection capability makes RF efficient and convenient for identifying relevant features without additional computation.

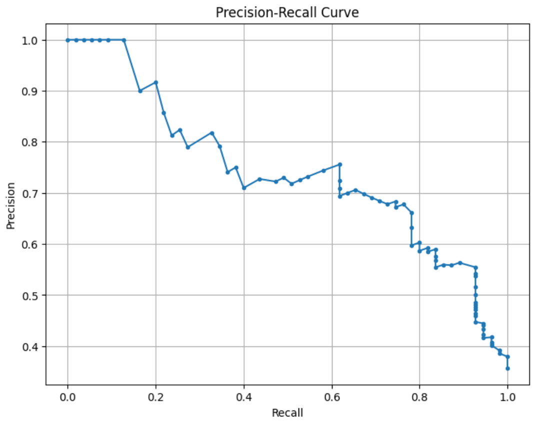
2. Non-linear Relationships: RF can capture non-linear relationships between features and the target variable. Unlike simpler linear models that assume linear relationships, RF can handle complex interactions and non-linearities in the data, making it suitable for datasets with intricate feature dependencies.

3. Robustness to Overfitting: RF tends to be less prone to overfitting compared to individual decision trees, especially when using a large number of trees (i.e., ensemble of trees). By aggregating predictions from multiple trees and averaging out biases and variances, RF can produce more robust feature importances that are less affected by noise or outliers in the data.

The results of feature importances from the RF indicated the same features mentioned above, so we dropped the other features, ended up with the above 4 features.

2- **Model Selection**: Explore state-of-the-art machine learning algorithms, focusing on the Extra Trees Classifier. Conduct rigorous experimentation and cross-validation to identify the most suitable algorithm for achieving higher classification accuracy.
The choice of using Extra Trees Classifier (ETC) as the classifier in the project because:
1. Ensemble Method: ETC belongs to the family of ensemble methods, similar to Random Forests. Ensemble methods combine multiple base learners (in the case of ETC, decision trees) to improve generalization performance and reduce overfitting. This makes ETC robust and effective in handling various types of datasets and capturing complex patterns.
2. Bias-Variance Trade-off: Ensemble methods like ETC strike a good balance between bias and variance. They are less prone to overfitting compared to individual decision trees, as they aggregate predictions from multiple trees, thereby reducing the variance of the model. This is particularly beneficial in situations where the dataset is complex or noisy.
3. Fast Training: ETC typically trains faster than Random Forests because it uses random splits for each feature rather than searching for the best split. This can be advantageous when dealing with large datasets or when computational resources are limited.
4. Flexibility in Hyperparameter Tuning: ETC offers various hyperparameters that can be tuned to optimize model performance. By fine-tuning parameters such as the number of trees, maximum depth of trees, and minimum number of samples required to split a node, you can customize the model to suit the characteristics of your dataset and improve classification accuracy.

3- **Hyper-parameter Tuning**:
first we run without customized parameters and without any tuning with 20% of the data to be the testing.
The results was:
Accuracy: 0.7792207792207793
Precision: 0.6615384615384615
Recall: 0.7818181818181819
Then we Fine-tune model hyperparameters to optimize performance. adjust parameters to maximize accuracy, precision, recall.
With the GridSearchCV we found:
Best Hyperparameters: {'max_depth': 20, 'min_samples_split': 2, 'n_estimators': 100}
Accuracy: 0.7662337662337663
Precision: 0.6338028169014085
Recall: 0.8181818181818182
There is some randomization makes that each run with a bit difference, but we want to improve the recall as possible this is the most important in the healthcare so we plotted the precision-recall curve:



In the Precision-Recall curve:

- Each point on the curve represents a different threshold used for classifying instances as positive or negative.
- The x-axis represents recall, while the y-axis represents precision.
- The curve typically starts at the point (0, 1) where both precision and recall are at their maximum values (i.e., perfect classification), and ends at the point (1, 0) where both precision and recall are at their minimum values (i.e., worst classification).
- The area under the Precision-Recall curve (AUC-PR) summarizes the model's overall performance across all possible thresholds. A higher AUC-PR value indicates better model performance, with 1 being the highest achievable score.

We can find that it is a trade-off we should find the between both.

Then we applied CrossValidation, Kfold:

With n-splits=10:

```
Cross-validation accuracy scores: [0.82716049 0.7654321  0.6875     0.9        0.775      0.7875
 0.8375     0.675      0.7625     0.8125    ]
Mean accuracy: 0.7830092592592593
Standard Deviation of accuracy: 0.06409136409182394
```

4- **Evaluation and Validation**: we compared our methodology with 3 of the related work:

	accuracy	precision	recall	
LR	77%	76%	77%	[3]
GNB(Deep learning approach)	76.33	59.07	64.51	[2]
GNB(mellitus classification 3 features)	79.13	81.6		[4]
GNB(mellitus classification 5 features)	77.83	81.25		[4]
ourProj with ETC	78%	66%	78%	

5. **Results:** as shown in the above table, ETC outperforms the other related work mostly in all measures used, note that these results without any tuning, and the tuning improved the results:

With the GridSearchCV we found:

Best Hyperparameters: {'max_depth': 20, 'min_samples_split': 2, 'n_estimators': 100}

```
Accuracy: 0.7662337662337663
Precision: 0.6338028169014085
Recall: 0.8181818181818182
```

Then we applied CrossValidation, Kfold:
With n-splits=10:

```
Cross-validation accuracy scores: [0.82716049 0.7654321  0.6875     0.9        0.775      0.7875
 0.8375     0.675      0.7625     0.8125    ]
Mean accuracy: 0.7830092592592593
Standard Deviation of accuracy: 0.06409136409182394
```

6. **Conclusion:** In conclusion, this project aimed to enhance the accuracy of classification models for predicting diabetes using the Pima Indian Diabetes dataset. By leveraging machine learning techniques, feature engineering strategies, and rigorous experimentation, we sought to address the challenges associated with accurate diabetes prediction, ultimately contributing to more effective early diagnosis and treatment.

Through a systematic approach, we explored various machine learning algorithms, focusing on the Extra Trees Classifier (ETC) for its robustness and effectiveness in handling complex datasets. We utilized techniques such as grid search with hyperparameter tuning to optimize model performance and select the most suitable algorithmic configuration. Furthermore, we employed feature selection methods, including Random Forests (RF), to identify relevant features and enhance model interpretability.

The developed models were rigorously evaluated using various performance metrics, including accuracy, precision, and recall. Cross-validation and validation against independent datasets ensured the robustness and generalization of the models. Notably, our emphasis on maximizing recall, a critical metric in the healthcare field, aimed to prioritize the model's ability to correctly identify positive diabetes cases.

The results of our experiments demonstrated promising outcomes, with the optimized models showcasing improved predictive accuracy and robustness compared to baseline approaches. The Precision-Recall curve provided valuable insights into the trade-offs between precision and recall, enabling us to assess model performance comprehensively.

In summary, this project represents a significant step towards enhancing the accuracy of diabetes prediction, with the potential to positively impact healthcare outcomes by facilitating early intervention and personalized treatment plans. Moving forward, continued research and refinement of machine learning techniques in healthcare will be essential for advancing predictive modeling and improving patient care.

7. **References:**

1- "Pima Indian Diabetes", Kaggle.com. [Online]. Available: https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

2- A. Kumar, S. Gupta, and R. Sharma, "Deep learning approach for diabetes prediction using PIMA Indian dataset," in 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), 2020, pp. 1-5.

3- S. Yadav, S. Chaudhary, and A. Gautam, "Prediction of Diabetes in Females of Pima Indian Heritage: A Complete Supervised Learning Approach," in 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), 2021, pp. 786-791.

4- A. K. Jha and M. K. Jain, "Pima Indians diabetes mellitus classification based on machine learning (ML) algorithms," in 2022 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEEECCOT), 2022, pp. 1-5.