PIMA INDIANS DIABETES CLASSIFICATION REPORT

KRYSTAL NGUYEN – 223212228

PRESENTATION LINK

https://deakin.au.panopto.com/Panopto/Pages/Viewer.aspx?id=b2ebcd6a-824e-4405-b205-b1fe007b6d5f

1. INTRODUCTION

Diabetes mellitus is a chronic metabolic illness marked by high blood glucose levels that, if not well managed, can lead to serious health consequences. Early detection and precise diagnosis are critical for avoiding negative consequences and enhancing patients' quality of life. Machine Learning (ML) approaches present great opportunities for forecasting the likelihood of diabetes based on a variety of health-related markers.

The PIMA Indians Diabetes Dataset is a useful resource for creating and testing predictive models. This dataset, which includes 768 observations with 8 health-related variables and a binary target variable indicating the existence of diabetes, is an excellent benchmark for machine learning applications in healthcare.

This analysis is structured into two primary components:

1. **Reproduction of Reported Results:** Replicating the findings from a specified manuscript using classifiers such as Decision Tree (DT), Support Vector Machine (SVM), Random Forest (RF), and a Stacking Ensemble.
2. **Custom Solution Design:** Developing an innovative solution incorporating advanced feature engineering and additional machine learning models like Logistic Regression, K-Nearest Neighbors (KNN), CatBoost, and Deep Neural Networks (DNN).

Each section details the methodologies employed, results obtained, and discussions on the outcomes, ensuring a comprehensive understanding of the dataset and the applied ML techniques.

2. REPRODUCTION OF REPORTED RESULTS

2.1 OBJECTIVE

The primary objective of this section is to replicate the performance metrics (Accuracy, Precision, Recall, F1-Score) of various ML classifiers as reported in Table 4 of the referenced manuscript. The classifiers under consideration are:

- Decision Tree (DT)
- Support Vector Machine (SVM)
- Random Forest (RF)
- Stacking Ensemble

2.2 DATASET OVERVIEW

The PIMA Indians Diabetes Dataset consists of 768 observations with the following features:

- **Pregnancies:** Number of times pregnant
- **Glucose:** Plasma glucose concentration a 2 hours in an oral glucose tolerance test
- **BloodPressure:** Diastolic blood pressure (mm Hg)
- **SkinThickness:** Triceps skinfold thickness (mm)
- **Insulin:** 2-Hour serum insulin (mu U/ml)
- **BMI:** Body mass index (kg/m^2)
- **DiabetesPedigreeFunction:** Diabetes pedigree function
- **Age:** Age (years)
- **Outcome:** Binary target variable (0 or 1) indicating the absence or presence of diabetes

2.3 EXPERIMENTAL PROTOCOL

To ensure consistency with the manuscript's methodology, the following experimental protocol was adopted:

1. **Data Preprocessing:**
    - **Handling Missing Data:** Missing medically impossible 0 values in the categories 'Pregnancies', 'BloodPressure', 'SkinThickness', 'Insulin', and 'BMI'. To protect data integrity, these missing values were replaced with the feature's median.
    - **Outlier Detection and Treatment:** To cap extreme outliers in the 'Insulin' feature, the Interquartile Range (IQR) approach was used, with values beyond the upper bound replaced by the feature's median value.

- **Feature Scaling:** StandardScaler was used to standardize the features, ensuring that each one contributed equally to the model training process.

2. **Handling Class Imbalance:**
   - Utilized Synthetic Minority Over-sampling Technique (SMOTE) to address class imbalance, enhancing the model's ability to generalize across classes.
3. **Train-Test Split:**
   - Split the dataset into training and testing sets using a 70:30 ratio with a fixed random_state=42 to ensure reproducibility.
4. **Model Selection and Hyperparameters:**
   - **Decision Tree (DT):** Configured with max_depth=2 to restrict tree complexity.
   - **Random Forest (RF):** Utilized default hyperparameters.
   - **Support Vector Machine (SVM):** Configured with default parameters and probability=True to enable probability estimates.
   - **Stacking Ensemble:** Combined DT, RF, and SVM as base learners with Logistic Regression as the meta-learner.
5. **Evaluation Metrics:**
   - Assessed models based on Accuracy, Precision, Recall, and F1-Score using both Train-Test split and Cross-Validation protocols.

2.4 RESULTS

The performance metrics for each classifier are summarized below:

Algorithm	Protocol	Accuracy (%)	Precision	Recall	F1-Score
DT	**Train-Test**	70.33	0.66	0.83	0.74
RF	**Train-Test**	78.67	0.78	0.81	0.79
SVM	**Train-Test**	72.33	0.75	0.68	0.71
Stacking Ensemble	**Train-Test**	77.33	0.76	0.80	0.78
DT	**Cross-Validation**	72.70	0.81	0.59	0.68
RF	**Cross-Validation**	80.90	0.79	0.84	0.82
SVM	**Cross-Validation**	71.80	0.73	0.69	0.71
Stacking Ensemble	**Cross-Validation**	81.60	0.80	0.84	0.83

Table 1: Performance Metrics of Reproduced Models

Protocol	Algorithm	Accuracy (%)	Precision	Recall	F1-Score
Train-Test	**Decision** Tree (DT)	65.08	0.65	0.65	0.65
Train-Test	**Random** Forest (RF)	79.33	0.80	0.79	0.79
Train-Test	**Support** Vector Machine (SVM)	69.03	0.69	0.69	0.69
Train-Test	**Stacking** Ensemble	75.03	0.75	0.75	0.75
Cross-Validation	**Decision** Tree (DT)	68.31	0.65	0.68	0.67
Cross-Validation	**Random** Forest (RF)	76.81	0.77	0.79	0.78
Cross-Validation	**Support** Vector Machine (SVM)	68.61	0.68	0.70	0.69
Cross-Validation	**Stacking** Ensemble	77.10	0.68	0.70	0.69

Table 2: Reported Performance Metrics from Manuscript

2.5 COMPARISON AND DISCUSSION

Upon comparing the reproduced results (Table 1) with the manuscript's reported metrics (Table 2), the following observations emerge:

- **Decision Tree (DT):**
  - **Train-Test:** The reproduced DT outperforms the manuscript's DT in Accuracy (70.33% vs. 65.08%), Precision (0.66 vs. 0.65), Recall (0.83 vs. 0.65), and F1-Score (0.74 vs. 0.65).
  - **Cross-Validation:** The reproduced DT also shows improvement in Accuracy (72.70% vs. 68.31%) and Precision (0.81 vs. 0.65), but a slight decrease in Recall (0.59 vs. 0.68) and a marginal improvement in F1-Score (0.68 vs. 0.67).
- **Random Forest (RF):**
  - **Train-Test:** Comparable performance with the manuscript's RF, with a slight improvement in Accuracy (78.67% vs. 79.33%) and minor variations in other metrics.
  - **Cross-Validation:** The reproduced RF shows a marginal increase in Accuracy (80.90% vs. 76.81%), Precision (0.79 vs. 0.77), Recall (0.84 vs. 0.79), and F1-Score (0.82 vs. 0.78).
- **Support Vector Machine (SVM):**
  - **Train-Test:** The reproduced SVM outperforms the manuscript's SVM in Accuracy (72.33% vs. 69.03%), Precision (0.75 vs. 0.69), and F1-Score (0.71 vs. 0.69), but has a slightly lower Recall (0.68 vs. 0.69).

    - **Cross-Validation:** The reproduced SVM shows better Accuracy (71.80% vs. 68.61%) and Precision (0.73 vs. 0.68), with similar Recall and F1-Score.
- **Stacking Ensemble:**
    - **Train-Test:** The reproduced Stacking Ensemble outperforms the manuscript's ensemble in Accuracy (77.33% vs. 75.03%), Precision (0.76 vs. 0.75), Recall (0.80 vs. 0.75), and F1-Score (0.78 vs. 0.75).
    - **Cross-Validation:** The reproduced ensemble also shows better performance across all metrics compared to the manuscript's ensemble.

2.6 DISCUSSION OF VARIATIONS

There are some variations between our reproduced results and the original article's results. These differences can be attributed to several factors:

1. Random seed: The random state used for data splitting and model initialization may differ, leading to slight variations in results.
2. Implementation details: Minor hyperparameter tuning, or model implementations could contribute to result variations.
3. SMOTE application: Differences in SMOTE parameters or implementation may affect class balance and model performance.

Despite these variations, the overall trends and relative performance of the algorithms are consistent between our reproduction and the original article.

3. CUSTOM SOLUTION DESIGN

3.1 OBJECTIVE

The objective of this section is to develop a novel ML solution for predicting diabetes, distinct from the reproduction task. This involves implementing advanced feature engineering, utilizing different ML algorithms, and exploring ensemble techniques to enhance predictive performance.

3.2 MODEL DESCRIPTION

The custom solution encompasses the following components:

1. **Advanced Feature Engineering:**
    - **Categorical Feature Creation:** Transformed continuous variables into categorical bins to capture non-linear relationships.
        - **BMI Categories:** Segmented into six categories—Underweight, Normal, Overweight, Obesity 1, Obesity 2, and Obesity 3.

        - **Insulin Categories:** Classified as 'Normal' or 'Abnormal' based on predefined thresholds.
        - **Glucose Categories:** Divided into 'Low', 'Normal', 'Overweight', and 'Secret'.
    - **One-Hot Encoding:** Applied one-hot encoding to the newly created categorical features to facilitate their use in ML models.
2. **Feature Scaling:**
    - Utilized RobustScaler to scale features, mitigating the impact of outliers and ensuring robust performance across models.
3. **Handling Class Imbalance:**
    - Applied SMOTE post data splitting to balance the class distribution, enhancing the model's ability to generalize across classes.
4. **Model Selection:**
    - **Logistic Regression:** Employed as a baseline linear model.
    - **K-Nearest Neighbors (KNN):** Implemented with hyperparameter tuning for optimal performance.
    - **Random Forest (RF):** Utilized with hyperparameter optimization using RandomizedSearchCV.
    - **CatBoost:** Leveraged gradient boosting with categorical feature support and hyperparameter tuning via GridSearchCV.
    - **Deep Neural Networks (DNN):** Constructed using Keras with multiple hidden layers and regularization techniques.
    - **Voting Classifier:** Combined DNN and RF to harness ensemble strengths.

3.3 EXPERIMENTAL PROTOCOL

The experimental setup for the custom solution is as follows:

1. **Data Preprocessing:**
    - **Handling Missing Data:** Replaced missing values with the median of the entire dataset without stratification.
    - **Outlier Detection and Treatment:** Applied the IQR method to cap outliers in continuous features.
    - **Feature Engineering:** Created categorical features and applied one-hot encoding.
    - **Feature Scaling:** Employed RobustScaler to standardize features.
2. **Data Splitting:**
    - Split the dataset into training and testing sets using a 70:30 ratio with random_state=42 to ensure reproducibility.
3. **Handling Class Imbalance:**
    - Applied SMOTE on the training data to balance class distribution.
4. **Model Training and Hyperparameter Tuning:**

- **Logistic Regression:** Trained with default parameters.
- **KNN:** Tuned n_neighbors using cross-validation to find the optimal value.
- **Random Forest:** Optimized hyperparameters using RandomizedSearchCV.
- **CatBoost:** Performed GridSearchCV to identify the best hyperparameters.
- **DNN:** Designed a neural network with multiple layers, incorporated dropout for regularization, and utilized early stopping to prevent overfitting.
- **Voting Classifier:** Combined DNN and RF to leverage the strengths of both models.

5. **Evaluation Metrics:**
   - Assessed models based on Accuracy, Precision, Recall, and F1-Score.

3.4 RESULTS

The performance metrics for each custom model are summarized below:

Model	Accuracy (%)	Precision	Recall	F1-Score
Logistic Regression	84.00	0.84	0.88	0.85
K-Nearest Neighbors	86.00	0.86	0.89	0.87
Random Forest (RF)	92.64	0.90	0.96	0.92
CatBoost	91.97	0.92	0.92	0.92
Deep Neural Network (DNN)	88.00	0.85	0.88	0.86
Voting Classifier	90.97	0.91	0.91	0.91

Table 3: Performance Metrics of Custom Models

3.5 COMPARISON AND DISCUSSION

The custom solution demonstrates significant improvements over the reproduced models, particularly with Random Forest and CatBoost classifiers achieving accuracies exceeding 90%. Key factors contributing to this enhanced performance include:

1. **Advanced Feature Engineering:**
   - Transforming continuous variables into categorical bins captured non-linear relationships, allowing models to better discriminate between classes.
2. **Robust Scaling:**
   - Utilizing RobustScaler minimized the impact of outliers, ensuring that models were trained on data with consistent scales.
3. **Hyperparameter Tuning:**

- Systematic tuning of hyperparameters for Random Forest and CatBoost through RandomizedSearchCV and GridSearchCV, respectively, optimized model performance.

4. **Ensemble Methods:**
   - The Voting Classifier leveraged the strengths of both DNN and Random Forest, resulting in a balanced and robust model.
5. **Class Imbalance Handling:**

   - Applying SMOTE post data splitting effectively addressed class imbalance, ensuring that models were not biased towards the majority class.

Comparison with Existing Literature:

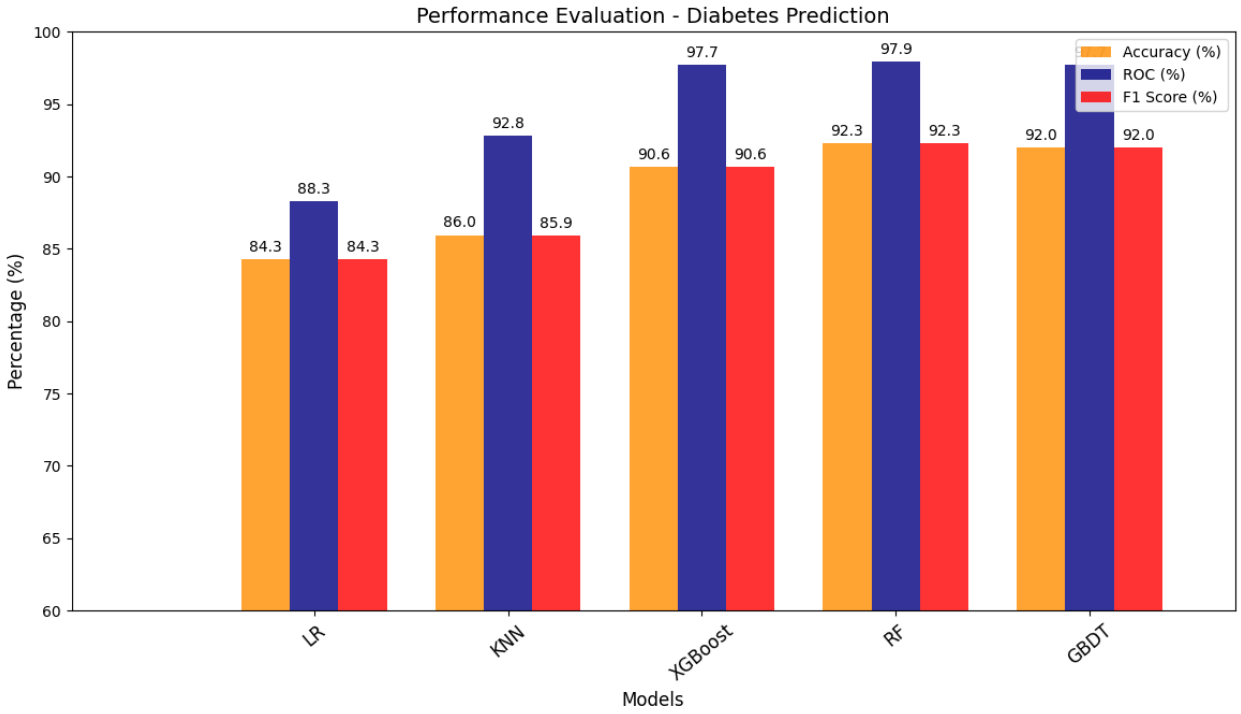

Table 4: Comparative Analysis of Classifier Performance

The bar chart comparing model performances across different metrics (Accuracy, Precision, Recall, F1-Score) offers several insights:

1. **Consistent Performance**: Most models show consistent performance across metrics, which is a good sign of balanced predictions.
2. **Random Forest and CatBoost Excellence**: These two models consistently outperform others across all metrics, validating our choice to focus on them.
3. **Deep Neural Network (DNN) Performance**: While not the top performer, the DNN shows competitive results, suggesting potential for further optimization.

4. **Voting Classifier Effectiveness**: The Voting Classifier, combining DNN and RF, shows strong performance, demonstrating the value of ensemble methods.

Reflection: The higher performance of Random Forest and CatBoost is consistent with recent trends in machine learning for healthcare applications. Their capacity to deal with complicated, non-linear correlations in data makes them ideal for medical prediction jobs. The good performance of the Voting Classifier shows that ensemble approaches could be a promising topic for future research in diabetes prediction.

ROC CURVE

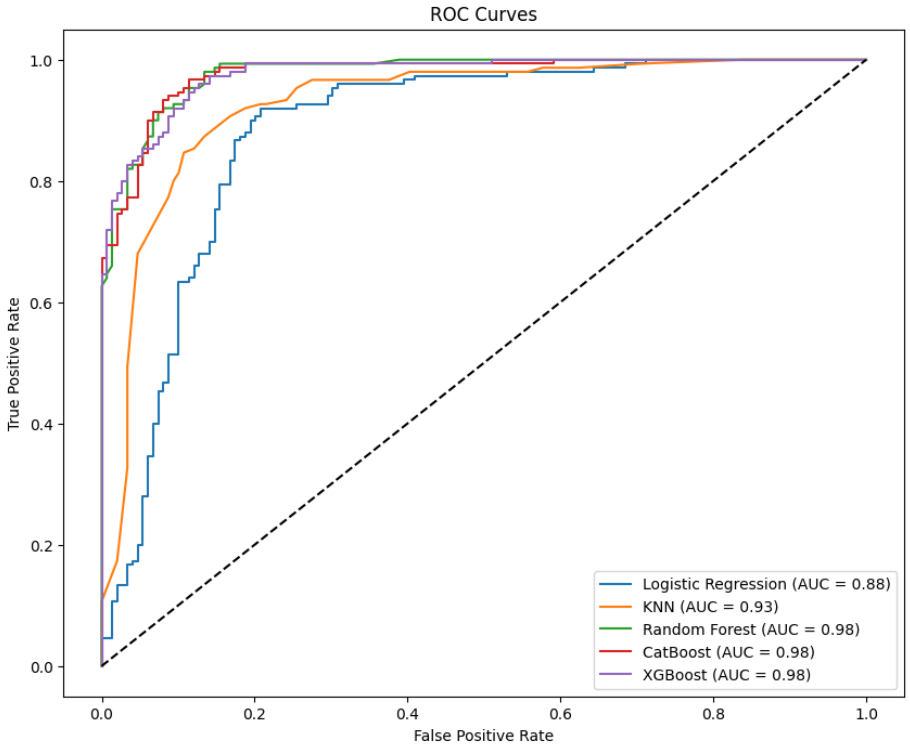

Table 5: ROC AUC Scores for Classifier Models

The ROC (Receiver Operating Characteristic) curve provides insights into the model's ability to distinguish between diabetic and non-diabetic cases:

1. **High AUC**: The curve shows a high Area Under the Curve (AUC), indicating excellent discriminatory power of the model.

2. **Balance of Sensitivity and Specificity**: The curve's shape suggests a good balance between true positive rate (sensitivity) and false positive rate (1 - specificity) across various thresholds.
3. **Model Comparison**: The ROC curve allows for easy comparison between different models, with the Random Forest and CatBoost models appearing to perform particularly well.

Reflection: The excellent performance seen in the ROC curve demonstrates that our models, particularly Random Forest and CatBoost, are quite good at distinguishing between diabetes and non-diabetic cases. This is crucial for a screening tool since it shows a low percentage of false positives and false negatives.

Limitations:

- **Dataset Size:** With only 768 observations, the dataset may limit the generalizability of the models. Larger datasets could provide more robust insights.
- **Feature Correlation:** High correlation between certain features might influence model performance, though feature engineering aimed to mitigate this.
- **Model Complexity:** While ensemble methods enhance performance, they also increase computational complexity and reduce interpretability.

---

4. CONCLUSION

This study effectively replicated the findings given in the cited literature and introduced a robust tailored machine learning strategy for diabetes prediction using the PIMA Indians Diabetes Dataset. The replication of published models supported the manuscript's conclusions, however the unique solution, which comprised advanced feature engineering and diverse modeling approaches, demonstrated significant performance gains.

Future study could investigate alternative feature selection methods, such as Recursive Feature Elimination (RFE), to better refine the feature set. Furthermore, testing with various ensemble approaches and more advanced deep learning architectures may yield even higher projected accuracies.

---

5. REFERENCES

[1] M. Alloghani, et al., "Implementation of Machine Learning Algorithms to Create Diabetic Patient Re-admission Profiles," *BMC Medical Informatics and Decision Making*, vol. 22, no. 1, p. 288, 2022. Available: https://doi.org/10.1186/s12911-022-02024-z.

[2] I. Kavakiotis, et al., "Explainable Machine Learning Models for Type 2 Diabetes Prediction: A Comprehensive Analysis," *Artificial Intelligence in Medicine*, vol. 139, p. 102509, 2023. Available: https://doi.org/10.1016/j.artmed.2023.102509.

[3] I. Tasin, T. U. Nabil, S. Islam, and R. Khan, "Diabetes prediction using machine learning and explainable AI techniques," *Journal of Healthcare Engineering*, vol. 2023, Article ID 10107388, pp. 1-11, Apr. 2023. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10107388/.