“EARLY DETECTION OF LIVER DISEASE LEVERAGING MACHINE LEARNING MODELS

KUSUMA KORADA
RITHIKA MAHAREDDY

INTRODUCTION

Liver disease is a significant health concern worldwide, and its prevalence has been on the rise in recent years.

Impact of Liver Disease:

- Rising global health concern.
- Increases healthcare costs due to diagnosis, treatment, and hospitalization.
- Advanced stages and liver cancer can be fatal, causing premature mortality.

Importance of Early Detection:

- Key to effective management; many liver diseases are treatable early on.
- Timely intervention can stop disease progression.
- Early-stage liver diseases have a better prognosis.
- Reduces risk of serious complications like cirrhosis and liver cancer.
- Prompt treatment improves patient quality of life and daily function.

LITERATURE REVIEW

- **[1]:** A research showed improved accuracy using SMOTE & CMVO techniques, achieving 82.46% accuracy on ILPD dataset.
- **[2]:** Leveraged PSO feature selection with J48 algorithm, achieving an accuracy of 95.04%.
- **[3]:** AdaC-TANBN method stood out with 69.03% accuracy, emphasizing the effectiveness of ensemble techniques.
- **[4]:** Introduced MLPNNB-C5.0 hybrid model, reaching an impressive accuracy of 94.13%.
- **[5]:** CNB approach showed 71.36% accuracy, outperforming traditional Naive Bayes.
- **[6]:** K-means clustering integrated with classifiers showed varied accuracies: NBC (56%), KNN (64%), C4.5 (69%).

LITERATURE REVIEW

- **[6]**: Employed K-means clustering with NBC, KNN, and C4.5, highlighting the versatility in model accuracies (56% to 69%).
- **[7]**: Used Random Forest and Logistic Regression with SMOTE and RFE, achieving accuracies from 56% to 94.13%. Emphasized the importance of demographic balance in datasets.
- **[8]**: Developed a Python-based GUI for disease classification using LR, SVM, KNN, and ANN, with ANN being most accurate.
- **[9]**: Applied Naive Bayes and FT tree, noting a high accuracy rate of 97.10% with FT tree.

JUSTIFICATION OF POPULATION

- The Indian Liver Patient Dataset (ILPD), which is accessible on Kaggle, forms the basis of our project's target population for predicting liver disease. The choice of this dataset was driven by its inclusion of vital health indicators that are key in predicting liver disease, such as bilirubin and albumin levels, along with a variety of blood tests.

- The ILPD dataset, hailing from Andhra Pradesh, India, comprises records of 416 patients with liver disease and 167 individuals without it. This comprehensive dataset is crucial for effectively training and testing machine learning models, enabling accurate predictions. Its diversity, encompassing a range of ages and both genders, provides a well-rounded view of liver disease occurrence.

- The dataset's coverage across various age groups and genders, in addition to its focus on pertinent health indicators, makes it an essential tool for creating and evaluating models geared towards liver disease prediction.

ANALYIS AND INTERPRETATION

Initial preview of the dataset's content:

- The first step involves displaying the first few rows of a DataFrame using the **data.head()** function. This is done to quickly inspect and verify the data's structure and content.
- It provides a concise preview of the dataset, showing the initial rows with column headers and their corresponding values. It's a fundamental step in data analysis to ensure that the data has been loaded correctly and to get an initial sense of the dataset's format and values.

	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	\
0	65	Female	0.7	0.1	187	
1	62	Male	10.9	5.5	699	
2	62	Male	7.3	4.1	490	
3	58	Male	1.0	0.4	182	
4	72	Male	3.9	2.0	195	

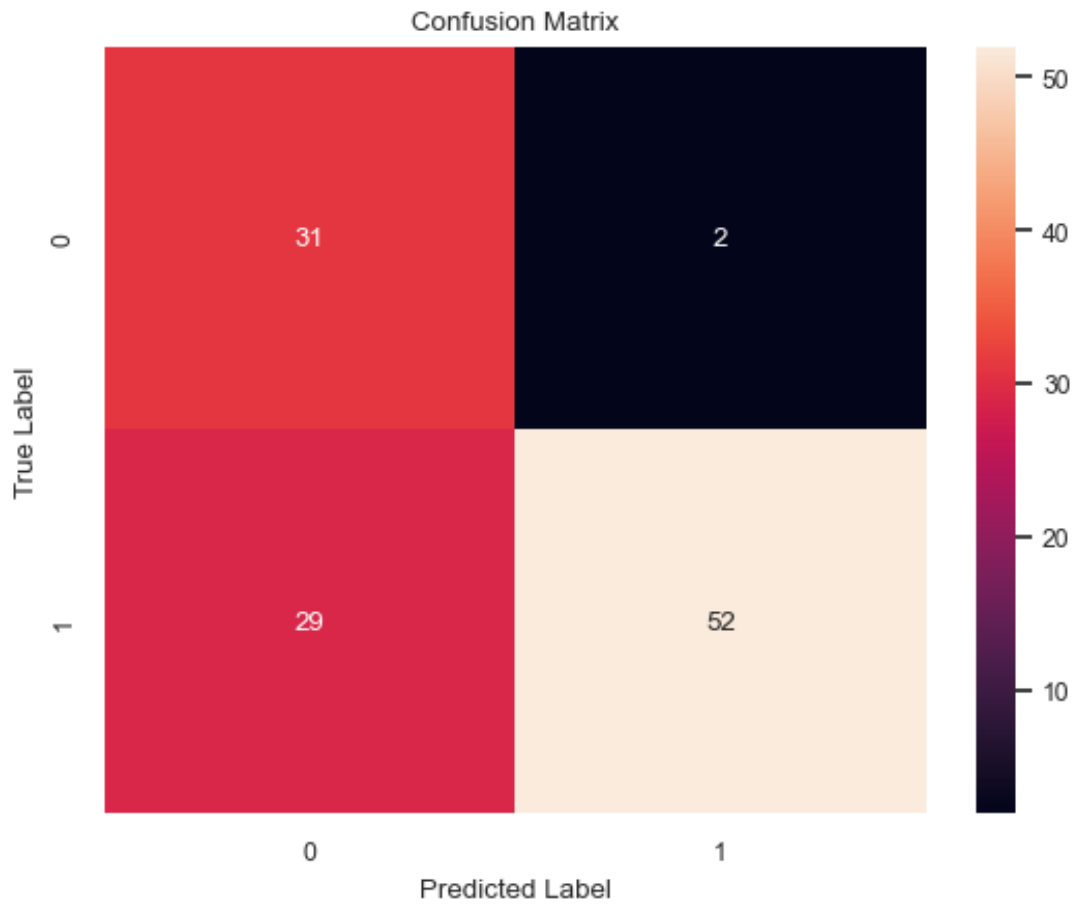
	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	\
0	16	18	6.8	
1	64	100	7.5	
2	60	68	7.0	
3	14	20	6.8	
4	27	59	7.3	

	Albumin	Albumin_and_Globulin_Ratio	Dataset
0	3.3	0.90	1
1	3.2	0.74	1
2	3.3	0.89	1
3	3.4	1.00	1
4	2.4	0.40	1

```
Total number of sample: 583
No. of features in each sample: 11
```

LOGISTIC REGRESSION

- SMOTE (Synthetic Minority Over-sampling Technique) is applied to the training data to address class imbalance. It synthesizes new examples for the minority class, making the class distributions more balanced.
- A logistic regression model is created using 'newton-cg' solver, which is an algorithm used for optimization.
- The model is trained on the oversampled training data to learn to predict the outcome variable.
- The trained model is used to predict the labels for the test data (the unseen data).




```
Accuracy: 0.7280701754385965
Precision: 0.9629629629629629
Recall: 0.6419753086419753
F1 Score: 0.7703703703703703
              precision    recall  f1-score   support

           0       0.52      0.94      0.67        33
           1       0.96      0.64      0.77        81

    accuracy                           0.73       114
   macro avg       0.74      0.79      0.72       114
weighted avg       0.83      0.73      0.74       114
```

The model has a relatively high number of true positives (31) and true negatives (52), which suggests it is capable of correctly identifying both the presence and absence of liver disease in many cases.

However, there are a significant number of false positives (29), which indicates that the model often predicts liver disease when it is not actually present.

The low number of false negatives (2) is a positive aspect of the model, especially in medical diagnostics, since it implies that the model rarely misses a case of liver disease.

K-NEAREST NEIGHBORS (KNN)

- Synthetic Minority Over-sampling Technique (SMOTE) is applied to the training data to address class imbalance. This creates synthetic samples for the minority class.
- A KNN classifier is instantiated with n_neighbors set to 5, which means the classifier considers the 5 nearest neighbors to determine the class of a new point.

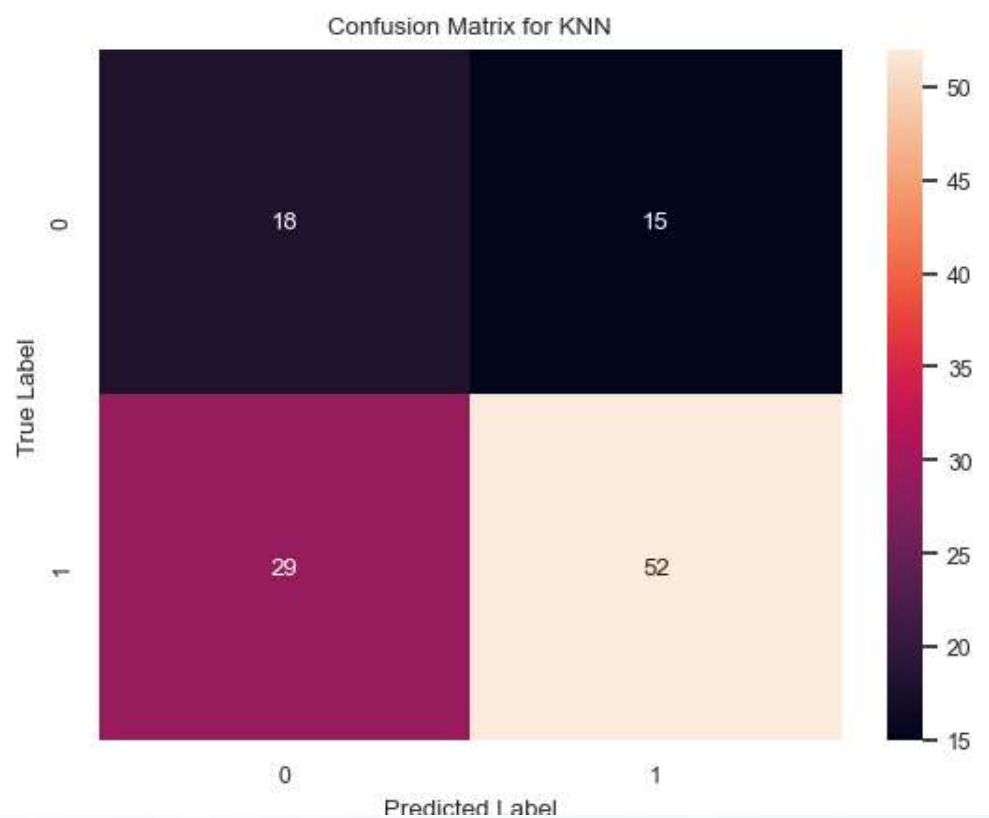
The KNN model is trained on the SMOTE-balanced training data (X_train_smote and y_train_smote).
The trained KNN model is used to predict the labels of the test data (X_test).

- Evaluation Metrics Calculation:

Various performance metrics are calculated to evaluate the model:
Accuracy: Proportion of correct predictions over the total number of predictions.
Precision: Proportion of true positive predictions over all positive predictions.
Recall: Proportion of true positives over all actual positives (sensitivity).
F1 Score: Harmonic mean of precision and recall, a balanced metric for binary classification performance.

- Confusion Matrix Creation:

A confusion matrix is generated from the test predictions, showing the number of true positives, false positives, true negatives, and false negatives




```
KNN Accuracy: 0.6140350877192983
KNN Precision: 0.7761194029850746
KNN Recall: 0.6419753086419753
KNN F1 Score: 0.7027027027027027
              precision    recall  f1-score   s

           0       0.38      0.55      0.45
           1       0.78      0.64      0.70

    accuracy                           0.61
   macro avg       0.58      0.59      0.58
weighted avg       0.66      0.61      0.63
```

The number of false negatives (FN) and false positives (FP) is relatively high, indicating that the model has a tendency to misclassify both conditions to some extent.

The high number of false positives (FP) may indicate that the model is overly sensitive, predicting the presence of the condition too frequently when it is not there.

SUPPORT VECTOR MACHINE

- **SMOTE for Class Imbalance**

Applied SMOTE to balance the training dataset.
Addressed class imbalance by generating synthetic examples of the minority class.

- **SVM Model Training**

Created an SVM model with a linear kernel.
Trained the model on the balanced training dataset (X_train_smote and y_train_smote).

- **Model Evaluation**

Computed key performance metrics:
Accuracy: Overall prediction correctness.
Precision: Ability to identify positive cases accurately.
Recall: Ability to identify all actual positives.
F1 Score: Balanced metric combining precision and recall.

- **Confusion Matrix Visualization**

Visualized the confusion matrix with a heatmap.
Displayed true positives, true negatives, false positives, and false negatives.

- **Results**

Presented performance metrics and classification report.
Evaluated the SVM model's effectiveness and addressed class imbalance.


```
SVM Accuracy: 0.6403508771929824
SVM Precision: 0.9545454545454546
SVM Recall: 0.5185185185185185
SVM F1 Score: 0.6719999999999999
              precision    recall  f1-score   support

           0       0.44      0.94      0.60        33
           1       0.95      0.52      0.67        81

    accuracy                           0.64       114
   macro avg       0.70      0.73      0.64       114
weighted avg       0.81      0.64      0.65       114
```

- The model has high precision but lower recall for predicting the disease, which might not be optimal in a medical setting where missing true cases of the disease could have serious consequences.
- The overall accuracy and F1 scores suggest that the model might benefit from further tuning or consideration of different features to improve performance, especially to balance out the precision and recall for both classes.

COMPARATIVE ANALYSIS OF MACHINE LEARNING MODELS

A structured comparison of performance metrics across three different machine learning models: Logistic Regression, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM) is performed.

	Logistic Regression	K-Nearest Neighbors	Support Vector Machine
Accuracy	0.859649	0.666667	0.850877
Precision	0.858407	0.839080	0.850877
Recall	1.000000	0.752577	1.000000
F1 Score	0.923810	0.793478	0.919431
ROC-AUC	0.677987	0.510000	0.700000

Interpretation:

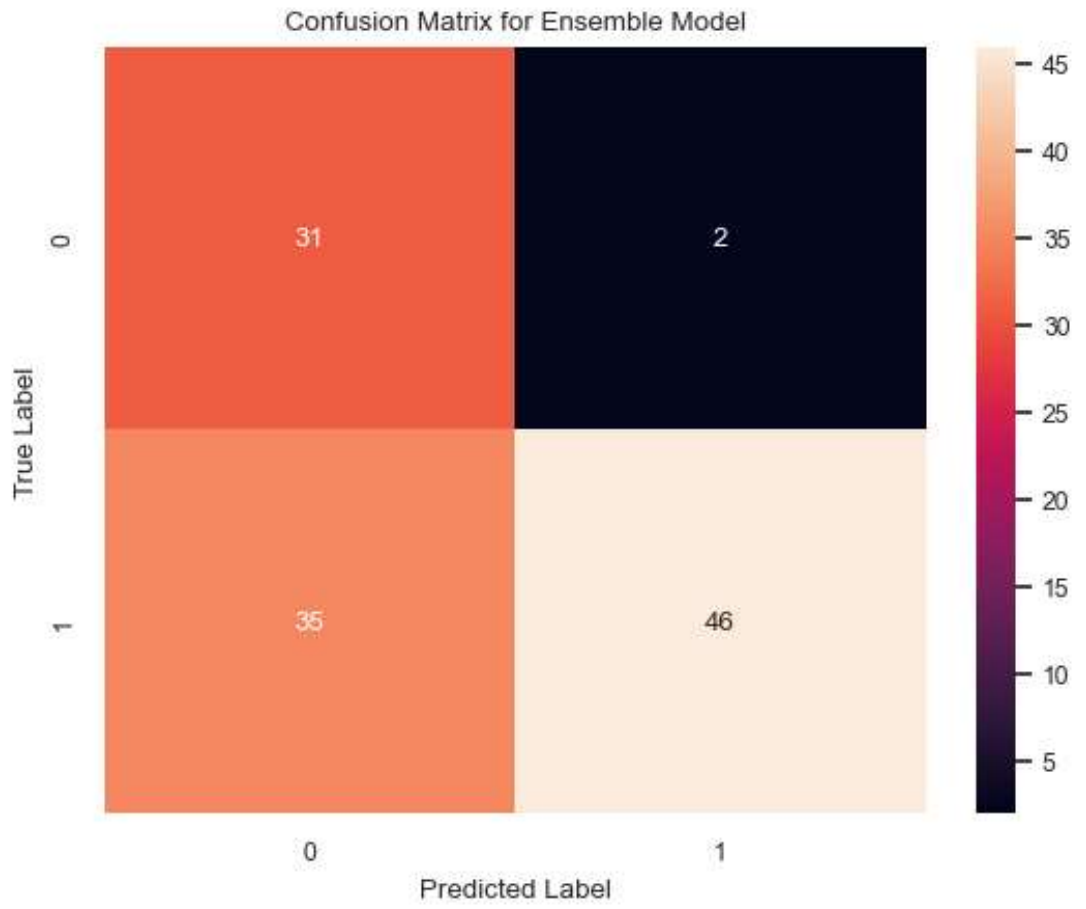
- Logistic Regression appears to be the most balanced model overall, excelling in all metrics but ROC-AUC.
- SVM performs identically to Logistic Regression in precision and recall but slightly lags in accuracy and F1 score.
- KNN has the weakest performance across all metrics, suggesting it might be less suitable for this particular dataset or problem.
- SVM and Logistic Regression have perfect recall rates, but the ROC-AUC suggests that they may not distinguish between classes as well as their recall rates might imply.

Logistic Regression seems to be the best model for this particular problem based on these metrics, with SVM being a close second.

Hard voting classifier

ENSEMBLE TECHNIQUES

- **Ensemble Creation**
  - Combined Logistic Regression, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM) into a VotingClassifier using hard voting.
  - Utilized the diversity of these models to improve overall classification performance.
- **Training**
  - Trained the ensemble model on SMOTE-balanced training data.
  - Addressed class imbalance to ensure reliable model performance.
- **Prediction**
  - Employed the ensemble model to predict labels for the test dataset.
  - Leveraged the combined knowledge of individual models for accurate predictions.
- **Evaluation Metrics**
  - Assessed model predictions using standard evaluation metrics:
    - Accuracy: Overall correctness of predictions.
    - Precision: Precision of class predictions.
    - Recall: Sensitivity or true positive rate.
    - F1 Score: Harmonic mean of precision and recall.
- **Confusion Matrix**
  - Created a visual representation of the model's performance with a confusion matrix.
  - Visualized true positives, true negatives, false positives, and false negatives using a heatmap.
- **Output and Interpretation**
  - Printed ensemble performance metrics to gauge the model's overall effectiveness.
  - Obtained insights into the strengths and weaknesses of the ensemble approach.
- **Classification Report**
  - Generated and printed a comprehensive classification report that provides detailed metrics for each class.
  - Gained a deeper understanding of the model's performance on different class labels.




```
Ensemble Accuracy: 0.6754385964912281
Ensemble Precision: 0.9583333333333334
Ensemble Recall: 0.5679012345679012
Ensemble F1 Score: 0.7131782945736433
              precision    recall  f1-score   support

           0       0.47      0.94      0.63        33
           1       0.96      0.57      0.71        81

    accuracy                           0.68       114
   macro avg       0.71      0.75      0.67       114
weighted avg       0.82      0.68      0.69       114
```

The model is particularly strong in identifying the negative class but less so for the positive class. It is highly precise in its positive predictions, but it tends to miss a considerable number of positive instances (low recall for class 1).

BLENDING ENSEMBLE TECHNIQUE

- To enhance predictive performance, mitigate overfitting, address model diversity, and achieve performance boosts, we employed the blending ensemble technique.
- Assessed the effectiveness of blending ensemble technique to calculate precision, recall, accuracy, f1 score and printed the metrics

```
Blend Ensemble Precision: 0.8133333333333334
Blend Ensemble Recall: 0.7530864197530864
Blend Ensemble Accuracy: 0.7017543859649122
Blend Ensemble F1 Score: 0.782051282051282
```

These metrics suggest that the Blend Ensemble model is reasonably effective, with a good balance between precision and recall. The accuracy indicates that there is room for improvement, possibly by tuning the model or using more sophisticated ensemble methods. The relatively high F1 score implies that the model is balanced, but depending on the specific application or domain, further optimization might be necessary to either reduce false positives (for precision) or false negatives (for recall).

RESEARCH RECOMMENDATIONS

Hyperparameter Tuning : Continue fine-tuning hyperparameters using advanced optimization techniques. The highest accuracy for blending ensemble indicates that there is room for improvement, possibly by tuning the model or using more sophisticated ensemble methods. The relatively high F1 score implies that the model is balanced, but depending on the specific application or domain, further optimization might be necessary to either reduce false positives (for precision) or false negatives (for recall).

Necessity for Cross Validation:
Using k-fold cross-validation to ensure the model's performance is consistent across different subsets of the data and Utilizing stratified folds because the classes are imbalanced to ensure that each fold is representative of the overall distribution.

Experimenting with different ensemble techniques:
It is advised because each method combines model predictions in distinct ways that can leverage the strengths and mitigate the weaknesses of individual models. Exploring beyond blending to techniques like stacking or bagging might be beneficial to reduce errors, complexity and improve the performance.

RESEARCH RECOMMENDATIONS

Data Expansion:
Expanding the dataset with more (and representative) data can lead to significant improvements in a model's performance.It reduces overfitting , model's performance.

Interpretability techniques :
These are methods that clarify how models arrive at their predictions, making the models' decisions transparent and understandable. For example, SHAP values explain individual prediction contributions by computing the impact of each feature, while Partial Dependence Plots illustrate how a feature affects predictions across the data range, holding other features constant.

Regularization techniques :

It help to prevent overfitting in machine learning models by penalizing complexity, encouraging the model to be simpler and therefore to generalize better to new data. This is achieved by shrinking the model's coefficients for less relevant features, thereby reducing the model's tendency to learn noise from the training data.

REFERENCES

- 1)S. Sreejith, H. Khanna Nehemiah, A. Kannan Clinical data classification using an enhanced SMOTE and chaotic evolutionary feature selection Comput Biol Med, 126 (February) (2020), Article
- 2) Kuzhippallil Maria Alex, Joseph Caralyn and A Kannan, "Comparative Analysis of Machine Learning Techniques for Indian liver disease patients", 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), pp. 778-782.
- 3) D. Gan, J. Shen, B. An, M. Xu, N. Liu Integrating TANBN with cost-sensitive classification algorithm for imbalanced data in medical diagnosis Comput Ind Eng, 140 (January) (2020), Article
- 4) M. Abdar, N.Y. Yen, J.C.S. Hung Improving the diagnosis of liver disease using multilayer perceptron neural network and boosted decision trees J Med Biol Eng, 38 (6) (2018

- 5) A. Anagaw, Y.L. Chang A new complement naïve Bayesian approach for biomedical data classification J Ambient Intell Hum Comput, 10 (10) (2019)
- 6) M.S.P. Babu, M. Ramjee, S. Katta, K. Swapna Implementation of partitional clustering on ILPD dataset to predict liver disorders Proc. IEEE Int. Conf. Softw. Eng. Serv. Sci. ICSESS (2016)
- 7) I. Straw, H. Wu Investigating for bias in healthcare algorithms: a sex-stratified analysis of supervised machine learning models in liver disease prediction BMJ Heal. Care Informatics, 29 (1) (Apr. 2022)
- 8) Chunlin Liang and Lingxi Peng, An automated diagnosis system of liver disease using artificial immune and genetic algorithms. Journal of medical systems, vol. 37, no. 2, pp. 1-10, 2013. 9) Rong-Ho Lin and Chun-Ling Chuang, "A hybrid diagnosis model for determining the types of the liver disease", Computers in Biology and Medicine, vol. 40, no. 7, pp. 665-670, 2010

THANK YOU