Heart disease prediction using hybrid ML algorithms.

Submitted in partial fulfillment of the requirements for the

Technical Answers for Real World Problems (TARP)

in

Computer Science Engineering with Specialization in Blockchain Technology

By

Chhavi Jain 20BKT0144

Under the guidance of

Prof. Rajkumar S

School of Computer Science and Engineering,

VIT, Vellore.

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

ABSTRACT

Cardiovascular diseases (CVDs), particularly heart disease, are a leading cause of global mortality. Early detection and risk assessment are crucial for improved patient outcomes. This study explores the potential of machine learning (ML) for heart disease prediction using a rich clinical dataset. We compare various ML classifiers to identify the most effective model for predicting heart disease. Our findings aim to inform the development of robust predictive tools for cardiovascular health management. This research contributes to bridging the gap between cutting-edge ML and clinical practice, empowering healthcare professionals with advanced tools for improved patient care.

INTRODUCTION

Cardiovascular diseases (CVDs) represent a formidable global health challenge, contributing to a staggering 17.9 million deaths annually and accounting for 32% of all global deaths. Among these, heart disease stands out as a significant component, imposing a substantial burden on healthcare systems and societies worldwide. The imperative for early detection and comprehensive risk assessment in managing heart disease cannot be overstated, as timely intervention can significantly improve patient outcomes and reduce the socioeconomic impact of CVDs.

The advent of machine learning (ML) techniques has ushered in a new era of possibilities in healthcare analytics, offering sophisticated tools to extract actionable insights from complex medical data. ML models excel in uncovering hidden patterns and relationships within datasets, empowering healthcare practitioners with enhanced decision-making capabilities. Leveraging these advancements, this study endeavors to harness the predictive potential of ML algorithms in analyzing a richly curated dataset comprising clinical data from renowned institutions such as the Cleveland Clinic Foundation, alongside contributions from Hungarian and Swiss healthcare facilities.

This research aims to conduct a rigorous evaluation and comparison of various ML classifiers in predicting the presence of heart disease. By systematically assessing the performance of these classifiers against established benchmarks, we aim to identify the most effective models for heart disease prediction. This comparative analysis not only sheds light on the strengths and limitations of different ML approaches but also lays the groundwork for developing robust predictive tools tailored to the nuances of cardiovascular health.

The subsequent sections of this paper will delve into a comprehensive review of relevant literature, contextualizing our study within the dynamic landscape of computational health informatics. We will elucidate the methodologies employed for data preprocessing, feature selection, and model training, ensuring transparency and reproducibility in our approach. Furthermore, we will discuss the implications of our findings in the broader context of predictive healthcare analytics, highlighting the potential applications for clinical decision support systems and personalized medicine initiatives.

Through this research endeavor, we aspire to contribute valuable insights to the evolving discourse on cardiovascular disease management, bridging the gap between cutting-edge ML technologies and clinical practice. Our goal is to empower healthcare professionals with advanced tools and methodologies that can drive tangible improvements in patient care and outcomes within the realm of cardiovascular health.

1.1 Objective

• Using patient data (demographics, lifestyle, biometrics), create and train a machine learning system to forecast the risk of heart disease.
• Select and include pertinent data sources for the heart disease prediction model's training and validation.
• Create and put into use a special interface for entering patient information and obtaining tailored risk assessments.
• Assess how well the heart disease prediction model performs on data that has not yet been seen, paying particular attention to accuracy and adherence to the moral precepts of inclusion and justice.
• Uphold morally and responsibly in the creation of AI over the course of the project, making sure that there is accountability, transparency, and compliance with applicable laws.

1.2 Problem Definition

One of the biggest problems facing medical research today is the proper detection of cardiac disease. Conventional diagnosis techniques frequently require a variety of expensive and time-consuming diagnostic tests and mostly rely on the experience of professionals. Furthermore, it is challenging to detect cardiac disease using traditional techniques alone due to its complexity, which is impacted by a variety of hereditary, environmental, and lifestyle variables.

There is a great chance to use machine learning techniques to increase the precision and effectiveness of heart disease detection because of the spread of digital health records and developments in data analytics. Multiple data kinds and sources may be integrated and analyzed using ML models, which has the potential to reveal new insights and prediction patterns that would not be seen using more conventional research techniques.

Evaluating the performance of several machine learning algorithm in identifying the existence of heart disease from clinical data is the issue this study attempts to solve. The objective of this research is to determine which models yield the most accurate forecasts and the optimal circumstances for their operation. In doing so, this study aims to further the creation of an automated prediction tool that can help medical professionals with early diagnosis and decision-making for the management and treatment of cardiac illness.

This project tackles the demand for sophisticated diagnostic instruments in the field of cardiology by utilizing machine learning to improve prediction accuracy and maybe cut down on the expense and time involved in conventional diagnostic procedures.

2. LITERATURE REVIEW:

Dulhar's[1] proposed a framework that combined the popular Naïve Bayesian classifier and Particle Swarm Optimization (PSO) feature selection algorithm for efficient heart disease prediction. The UCI dataset of VA Long Beach consisting of 270 instances and 14 features was used for the model training and testing processes. Of the 14 features, only 7 were selected for the heart disease prediction. From the experimental results, the Naïve Bayes predictive model performance was 79.12% accurate but escalated to 87.91% when integrated with the PSO selection algorithm. It was concluded that the NB+PSO model improved the heart disease classification accuracy, which is 8.79% better than the original NB performance.

Dulhar's[2] presented a framework based on neural networks (NN) to develop an effective heart disease prediction system (EHDPS) for predicting the risk level of heart disease. The MultiLayer Perceptron (MLP) neural network (NN) with backpropagation was used as a training algorithm. The UCI dataset of Cleveland consisting of 303 instances and 15 heart disease features was used for the model training and testing processes. The data was divided into 40% and 60% for the training and testing respectively. A data preprocessing operation was carried out to remove noisy data and missing values. Their experimental results showed that the proposed model was able to predict heart diseases with 100% accuracy.

Singh's [3] developed a machine learning-based hybrid intelligent system framework for heart disease patients' diagnosis using seven of the popular classification algorithms using Python. They include KNN, ANN, DT, SVM, NB, LR and MLP. The Cleveland dataset containing 303 instances with 76 features was used for model training and testing. They applied a 10-fold cross-validation approach to the data. Feature selection algorithms, including Relief, Minimal-Redundancy-Maximal-Relevance (mRMR) and Least Absolute Shrinkage and Selection Operator (LASSO), were used to select the best heart disease correlated features. The data was pre-processed to remove the instances with large missing values. This reduced the data size to 297 instances with only 14 features. Applying the feature selection algorithms reduced the features to 6 only as heart disease-related. They tested each of the classifiers with any of the feature selection algorithms in order to get the best-performing model. Their experimental results showed that SVM with the LASSO feature selection algorithm appeared the best-performing combination, as compared with other feature selection algorithms and classifiers. Narrowing the heart disease features to only 6 would lead to unreliable classification accuracy, as more relevant features were excluded.

Memon [4] presented a heart disease prediction framework that uses a Convolutional Neural Network based Multimodal Disease Prediction (CNN-MDRP) algorithm which uses both structured and unstructured big data from a particular hospital. It was a comparative study with a Convolutional Neural Network based Uni-modal Disease Prediction (CNN-UDRP) algorithm which uses only structured data. They used the Naïve Bayes (NB)classifier for the classification process. In their model, the automatic selection of characteristics from large data improves the disease prediction accuracy. Their experimental results showed that the CNN-MDRP model performed well in heart disease classification with an accuracy of 94.80. No dataset was specified for the algorithm training and testing.

Shrisant [5] proposed a tentative design of a cloud-based heart disease prediction system using machine learning techniques. Two of the UCI datasets: Cleveland heart disease data consisting of 303 instances with 14 features and VA Long Beach data consisting of 270 instances with also 14 features were merged together making a bigger dataset. Five machine learning algorithms, including MLP, LR, NB, RF, and SVM in the Java-based open access platform (WEKA) were applied in the classification and prediction processes. Of the five algorithms, SVM appeared the best classifier with a classification accuracy of 97.53%.

Nashif [6] also carried out a comparative investigation on heart disease prediction using support vector machine, decision tree, and k-nearest neighbor algorithms. They used the VA Long Beach dataset obtained from the UCI machine learning repository, which comprises of 270 instances and 12 attributes for the algorithm training and testing purposes. The model was evaluated based on accuracy, sensitivity, and specificity using confusion matrix. Their experimental results showed that the Support Vector Machine (SVM) performed better than KNN and DT in classifying heart disease patients, with an accuracy of 92%, sensitivity of 100%, and specificity of 83%.

Hariharan [7] designed a framework for heart disease prediction using data mining techniques. One of the UCI datasets was used to train and test the system using the 10-fold cross validation method. SVM, NB, KNN, C4.5, and Back Propagation classifiers were used and performances were compared. The SVM classification algorithm appeared the best in terms of accuracy, sensitivity, precision, low specificity, mean absolute error, and low computing times in all feature combinations. The SVM classifier accuracies at 13, 12, 11, 10, 9, 8and 7 feature combinations were 83.70%, 84.00%, 84.00%, 84.10%, 84.40%, 84.80%, and 85.90% respectively. The datasets employed for algorithm training and testing were not clearly specified.

Voleti [8] proposed a framework that combined the popular Naïve Bayesian classifier and Particle Swarm Optimization (PSO) feature selection algorithm for efficient heart disease prediction. The UCI dataset of VA Long Beach consisting of 270 instances and 14 features was used for the model training and testing processes. Of the 14 features, only 7 were selected for the heart disease prediction. From the experimental results, the Naïve Bayes predictive model performance was 79.12% accurate but escalated to 87.91% when integrated with the PSO selection algorithm. It was concluded that the NB+PSO model improved the heart disease classification accuracy, which is 8.79% better than the original performance.

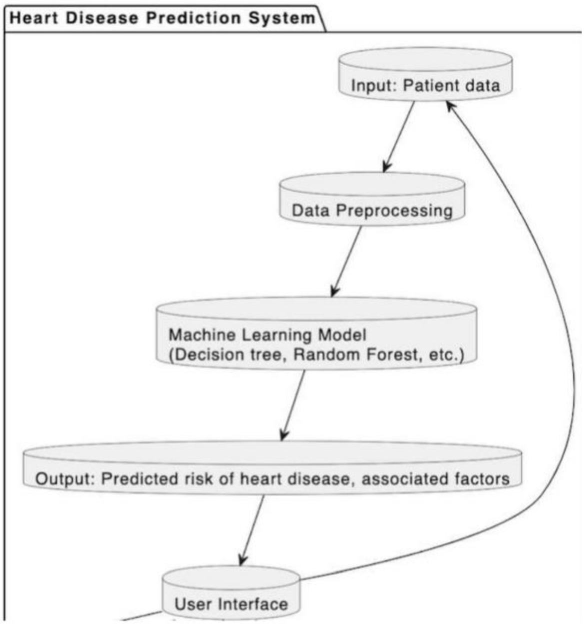
Sarah [9] et al (2022) provide us with a comparative study on heart disease prediction using machine learning techniques. Often the volume of these data is too vast for the human brain to compute. Hence, we use machine learning algorithms to predict heart disease in human beings using the above-mentioned data. This paper is concerned with the comparison of the various models and to determine which one of them is best suited for the prediction. The models used in this paper are Logistic Regression, Decision tree, Naive Bayes, SVM, K-Nearest Neighbors, and Random Forest. It was found that logistic regression performed best with an accuracy of 85.25%.

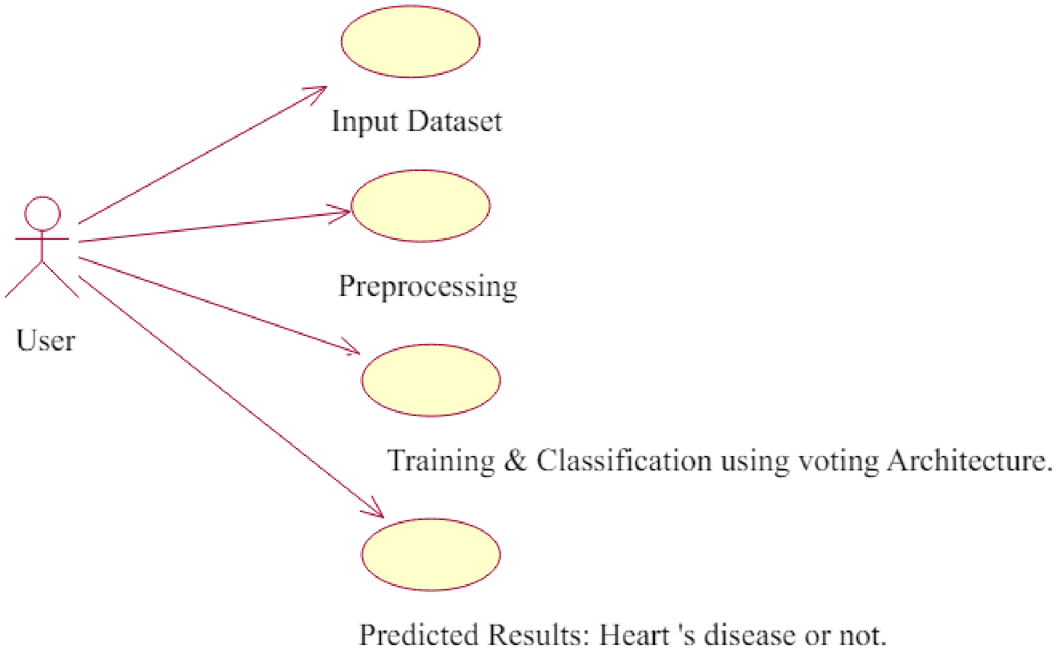
Asmit [10] et al (2022) try to set a heart prediction benchmark using 14 different parameters. In their study, we also tried to find correlations between the various features found in the database with the help of standard Mechanical Learning. Methods and use them effectively predict the risk of heart disease. This model can be useful to medical staff at their clinic as a decision support system.

3. SOFTWARE REQUIREMENTS:

- Operating System: Windows 10, macOS Mojave, or any popular distribution of Linux (e.g., Ubuntu 18.04 or later).
- Python Environment: Python 3.8 or later, with support for all required libraries.
- Main Libraries:
  Pandas (version 1.1 or later) for data manipulation.
  NumPy (version 1.19 or later) for numerical operations.
  Scikit-learn (version 0.24 or later) for machine learning algorithms and model evaluation.
  Matplotlib and Seaborn for data visualization.
  Plotly for interactive visualizations.
  Jupiter Notebook or JupyterLab for executing Python code in an interactive notebook format.

3.1. Architecture Diagram:


Heart Disease Prediction System
Input: Patient data
Data Preprocessing
Machine Learning Model
(Decision tree, Random Forest, etc.)
Output: Predicted risk of heart disease, associated factors
User Interface


Input Dataset
Preprocessing
User
Training & Classification using voting Architecture.
Predicted Results: Heart 's disease or not.

Process

4. Module Description:

This study employed several machine learning models to predict the presence of heart disease based on clinical data. The following modules were carefully chosen based on their suitability for classification tasks and their diverse approaches, from simple linear models to complex ensemble techniques. Each model is described below:

4.1. Logistic Regression

Description: Logistic Regression is a fundamental statistical model that predicts a binary outcome based on a linear combination of input features. It is widely used for its simplicity and efficiency in binary classification tasks.
Configuration: The model was implemented with L2 regularization to prevent overfitting, and the regularization strength was optimized using cross-validation.

4.2. Support Vector Machines (SVM)

Description: SVM is a powerful classification technique that finds the optimal hyperplane which maximizes the margin between different classes. It is effective in high-dimensional spaces and for cases where the number of dimensions exceeds the number of samples.
Configuration: Both linear and radial basis function (RBF) kernels were tested. Parameters such as the penalty parameter C and the kernel coefficient γ were optimized through grid search.

4.3. Random Forest

Description: Random Forest is an ensemble learning method based on decision tree classifiers. It builds multiple decision trees and merges them together to get a more accurate and stable prediction.

Configuration: The number of trees, the maximum depth of the trees, and the minimum number of samples required to split a node were among the hyperparameters tuned to enhance model performance.

4.4. Gradient Boosting Classifier

Description: Gradient Boosting is an ensemble technique that builds models sequentially, each new model correcting errors made by previously trained trees. It is known for its predictive accuracy and effectiveness on a wide range of problems.

Configuration: Parameters such as the number of boosting stages, learning rate, and the depth of each tree were optimized. The model was also tested with different loss functions to assess performance variations.

4.5. Voting Classifier

Description: A Voting Classifier is a meta-model that aggregates the predictions of multiple machine-learning models to make a final prediction. It uses majority voting for classification, which can lead to more robust overall performance.

Configuration: The Voting Classifier was configured with a combination of the above models. Both hard voting (based on predicted class labels) and soft voting (based on predicted probabilities) strategies were evaluated.

5. Model Implementation Details

- Software and Libraries: Python’s Scikit-learn library was primarily used for the implementation of these models due to its extensive support for various machine-learning algorithms and tools for model evaluation.
- Data Preparation: All models were trained on the same pre-processed dataset, which included feature scaling and encoding necessary for optimal model performance.
- Evaluation: Each model was evaluated using the same datasets, ensuring consistency in performance metrics across different models. The performance of each model is evaluated using the test set. The key metrics for evaluation include:
  - Accuracy: The ratio of correctly predicted instances to total instances.
  - Precision: The ratio of correctly predicted positive observations to the total predicted positives.
  - Recall: The ratio of correctly predicted positive observations to all observations in the actual class.

6. Source code:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from collections import Counter

from sklearn.multioutput import MultiOutputClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import LinearSVC, SVC
from lightgbm import LGBMClassifier
from xgboost import XGBClassifier, XGBRFClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.neural_network import MLPClassifier
!pip install catboost
from catboost import CatBoostClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier, ExtraTreesClassifier
from sklearn.ensemble import GradientBoostingClassifier, HistGradientBoostingClassifier,
VotingClassifier

from imblearn.under_sampling import RandomUnderSampler
!pip install category_encoders
from category_encoders import LeaveOneOutEncoder
from category_encoders.target_encoder import TargetEncoder
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.model_selection import train_test_split, GridSearchCV, RandomizedSearchCV,
cross_val_score
from scipy.stats import reciprocal, uniform
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report,
roc_auc_score, roc_curve, auc, r2_score

import warnings
warnings.filterwarnings("ignore")

df = pd.read_csv("/content/drive/MyDrive/heart_statlog_cleveland_hungary_final.csv")
df.head()

print(df.columns)

#check dublicates values
df.drop_duplicates(inplace=True)
df.reset_index(drop=True, inplace=True)
df.info()

cat_cols = ['sex', 'chest pain type', 'fasting blood sugar', 'resting ecg', 'exercise angina', 'ST slope',
'target']

plt.figure(figsize=(11,27))
sns.set(rc={'axes.facecolor':'#eee0e5', 'figure.facecolor':'#ffdab9'}, font_scale=0.7)
clr1 = ["#66cdaa", "#087EB0", "#2e8b57", "#cd9b9b"]
i = 0
j = 1
for col in cat_cols:
  feature = df.groupby(col)[col].count()
  plt.subplot(7, 2, i+1)
  sns.barplot(x=feature.index, y=feature.values, palette=clr1)
  plt.title(col, fontsize=15)
  plt.xlabel("")
  plt.subplot(7, 2, j+1)
  plt.pie(x=feature.values, autopct="%.1f%%", pctdistance=0.8, labels=feature.index,
colors=["#66cdaa", "#087EB0", "#2e8b57", "#cd9b9b"])
  i += 2
  j += 2
  plt.show()

num_cols = ['age', 'resting bp s', 'cholesterol', 'max heart rate', 'oldpeak']
plt.figure(figsize=(10,15))
sns.set(font_scale=0.7)
i=0
j=0
for col in num_cols:
 plt.subplot(5, 2, i+1)
 sns.boxplot(df[col], color="#f08080")
 plt.title(col, fontsize=12)
 plt.xlabel("target")
 plt.subplot(5, 2, j+2)
 sns.distplot(df[col], bins=30, color="#f08080")
 plt.title(col, fontsize=12)
 plt.xlabel("targete")
 i += 2
 j += 2
plt.tight_layout()
plt.show()

#Outliers We will eliminate rows that have outliers in more than one variable
outlier_list = []
for i in num_cols:
    # Check if column exists in DataFrame
    if i in df.columns:
        Q1 = df[i].quantile(0.25)
        Q3 = df[i].quantile(0.75)
        IQR = Q3-Q1
        outlier_step = IQR * 1.5
        index_list = df[(df[i] < Q1 - outlier_step) | (df[i] > Q3 + outlier_step)].index
        outlier_list.extend(index_list)
    else:
        print(f"Warning: Column '{i}' not found in DataFrame.")

outlier_list = Counter(outlier_list)
outlier_list = list(outlier_list.items())
multi_out_list = [key for key, value in outlier_list if value > 1]

print(f"Total number of rows with outliers: {len(outlier_list)}")
print(f"Number of rows with outliers in more than one variable :{len(multi_out_list)}")

df.drop(multi_out_list, axis=0, inplace=True)
df.reset_index(drop=True, inplace=True)
```

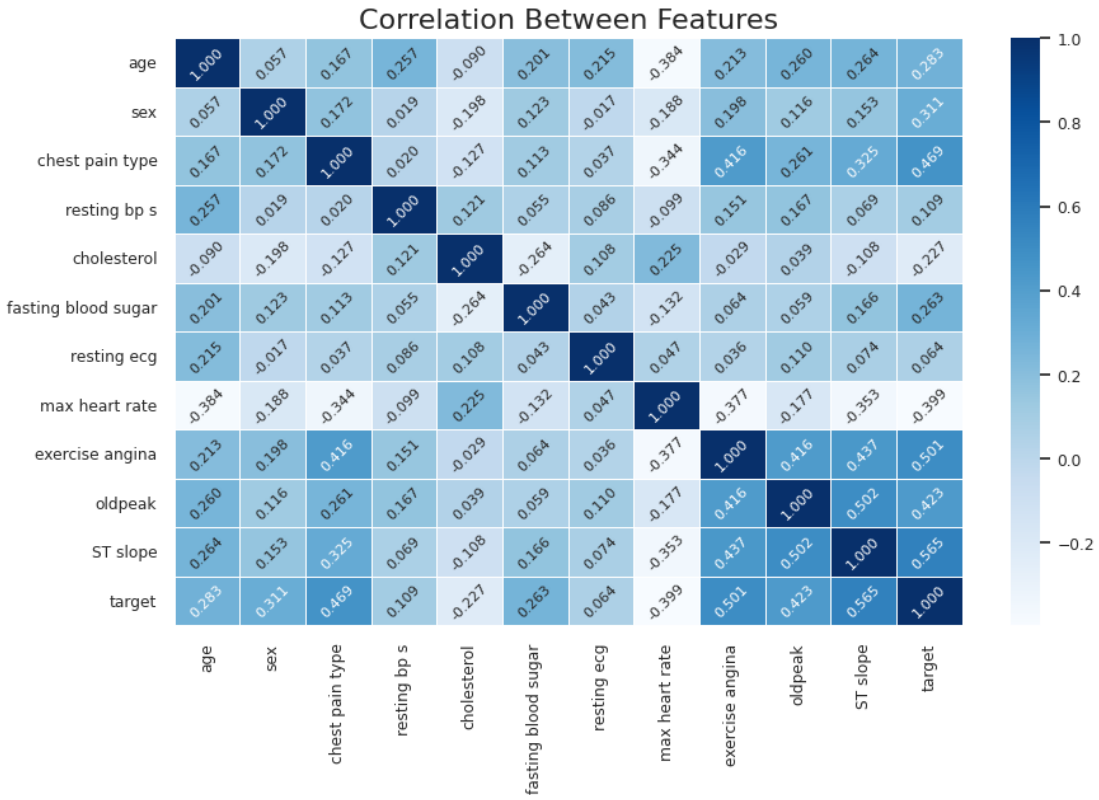
```
#Relationship of one feature of data set with other
heart_diase_corr = df.corr()["target"]
heart_diase_corr = heart_diase_corr.drop("target", axis=0).sort_values(ascending=False)

#various graphs
plt.figure(figsize=(8,4))
sns.set(font_scale=0.8)
sns.barplot(x=heart_diase_corr.index, y=heart_diase_corr, color="#4a804d")
plt.xticks(rotation=90)
plt.title("Relationship of variables with heart diase", fontsize=15)
plt.show()

#correlation matrix
plt.figure(figsize=(10,6))
sns.heatmap(df.corr(), annot=True, linewidths=0.4, fmt='.3f', cmap="Blues",
annot_kws={'size': 8, 'rotation': 45})
plt.title("Correlation Between Features", fontsize=16)
plt.show()

#feature selection
df = pd.get_dummies(data=df, columns=["chest pain type", "resting ecg", "ST slope"])

#splitting data and training
y = df["target"]
X = df.drop("target", axis=1)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, stratify=y,
random_state=42)
scaler = StandardScaler()
X_train[num_cols] = scaler.fit_transform(X_train[num_cols])
X_test[num_cols]= scaler.transform(X_test[num_cols])

#checking accuracy with various other algorithm
model_list = [LinearSVC(), LogisticRegression(), GradientBoostingClassifier(),
              MLPClassifier(max_iter=2000), AdaBoostClassifier(),
              HistGradientBoostingClassifier(),
              SVC(), XGBClassifier(), CatBoostClassifier(verbose=False)]
model_name_list = []
accuracy_list = []
for model_name in model_list:
 model = model_name
 model_cv = cross_val_score(model,
 X_train,
 y_train,
 cv=10,
 scoring= "accuracy",
 n_jobs=-1)
 model_name_list.append(model_name.__class__.__name__)
 accuracy_list.append(model_cv.mean())
 print(f"{model_name.__class__.__name__} cross validation score: {model_cv.mean()}")
 print("-" * 50)

#graphical representation of accuracy of various individual algorithm
plt.figure(figsize=(10,5))
clrs = ["brown" if i == max(accuracy_list) else "orange" for i in accuracy_list]
sns.barplot(x=accuracy_list, y=model_name_list, palette=clrs)
plt.axvline(0.8626, ls="--", lw=0.5, color="k")
plt.text(0.84,9, s="0.863", fontsize=12)
plt.title("Comparison of Models cross validation scores", fontsize=15)
plt.show()

#proposed hybrid model
gb_1 = GradientBoostingClassifier(**{"max_depth": 2,
 "min_samples_split": 5,
 "min_samples_leaf": 4,
 "learning_rate": 0.0448191306552163,
 "n_estimators": 110})
catb_2 = CatBoostClassifier(**{"depth": 6,
 "l2_leaf_reg": 8,
 "random_strength": 4,
 "subsample": 0.8263,
 "verbose": False})
voting = VotingClassifier(estimators= [("GBoosting", gb_1),
 ("CatBoost", catb_2)],
 voting="hard",
 n_jobs=-1)
voting_fit = voting.fit(X_train, y_train)

#choosing out ml algorithm for of proposed model
accuracy_list = []
for clf in (gb_1, catb_2, voting_fit):
 clf.fit(X_train, y_train)
 y_pred = clf.predict(X_test)
 acc = accuracy_score(y_test, y_pred)
 accuracy_list.append(acc)
 print(clf.__class__.__name__, accuracy_score(y_test, y_pred))

#comparing accuracy
selected_models = ["GradientBoosting", "CatBoost", "VotingClassifier"]
plt.figure(figsize=(8,3))
clrs = ["brown" if i == max(accuracy_list) else "orange" for i in accuracy_list]
sns.barplot(x=accuracy_list, y=selected_models, palette=clrs)
plt.axvline(0.9044, ls="--", lw=0.7, color="b")
plt.text(0.86,2.80, s="0.9044", fontsize=12, color="darkred")
plt.title("Accuracy scores", fontsize=15)
plt.show()

#accrucy of proposed model
```

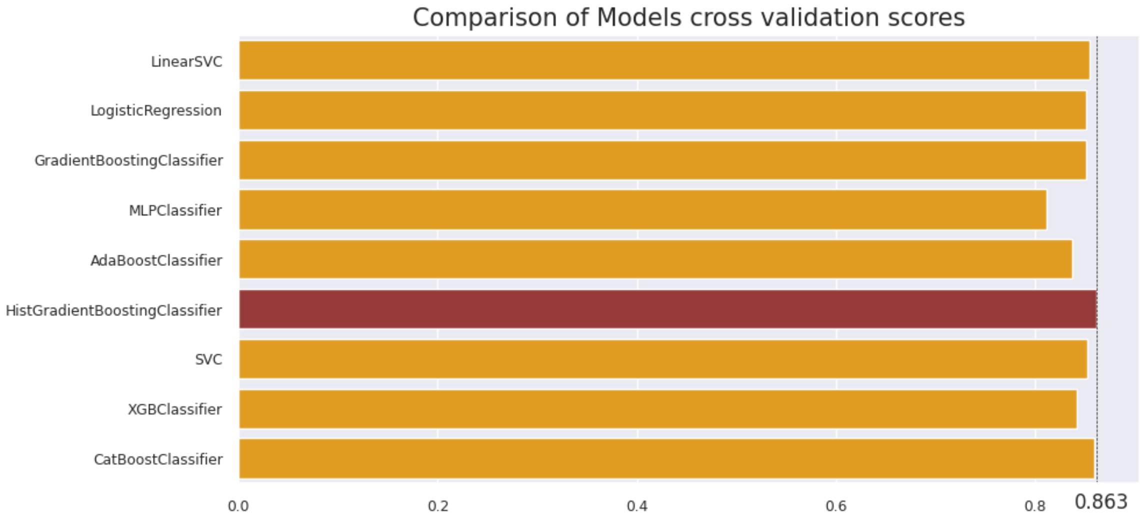
```
y_pred_voting = voting_fit.predict(X_test)
print(classification_report(y_test, y_pred_voting))

pred_gb = gb_1.predict(X_test)
GBoosting_cm = confusion_matrix(y_test, pred_gb)
pred_catb = catb_2.predict(X_test)
CatBoost_cm = confusion_matrix(y_test, pred_catb)
Voting_cm = confusion_matrix(y_test, y_pred_voting)
plt.figure(figsize=(15, 3))
plt.subplot(1,3,1)
sns.heatmap(GBoosting_cm, annot=True, fmt="g", cmap="Greens")
plt.title("GradientBoosting", fontsize=14)
plt.subplot(1,3,2)
sns.heatmap(CatBoost_cm, annot=True, fmt="g", cmap="Oranges")
plt.title("CatBoost", fontsize=14)
plt.subplot(1,3,3)
sns.heatmap(Voting_cm, annot=True, fmt="g", cmap="Blues")
plt.title("Voiting", fontsize=14)
plt.show()
```

7. Model Testing:

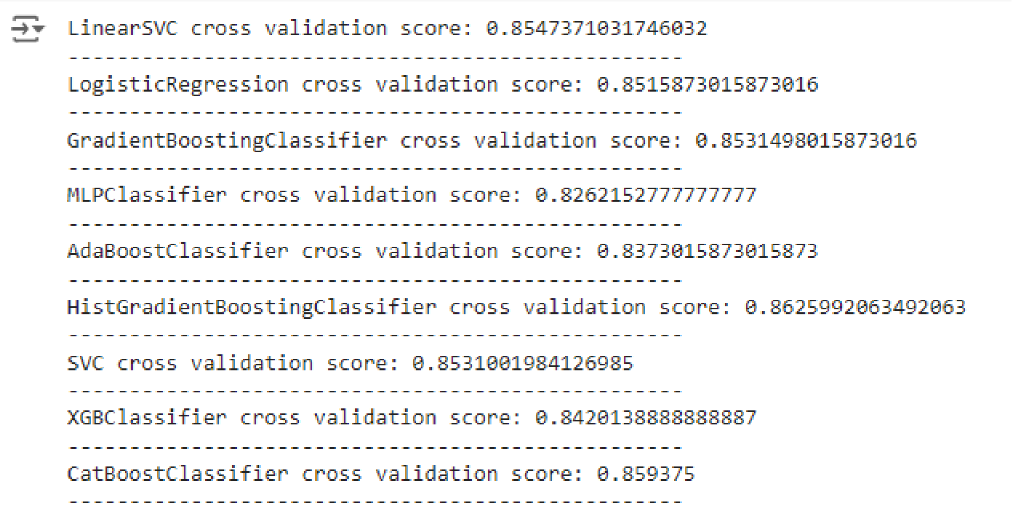
Here are some of the classification reports for different AI algorithms and models.

Correlation matrix:





Accuracy Report:



```
LinearSVC cross validation score: 0.8547371031746032
--------------------------------------------------
LogisticRegression cross validation score: 0.8515873015873016
--------------------------------------------------
GradientBoostingClassifier cross validation score: 0.8531498015873016
--------------------------------------------------
MLPClassifier cross validation score: 0.8262152777777777
--------------------------------------------------
AdaBoostClassifier cross validation score: 0.8373015873015873
--------------------------------------------------
HistGradientBoostingClassifier cross validation score: 0.8625992063492063
--------------------------------------------------
SVC cross validation score: 0.8531001984126985
--------------------------------------------------
XGBClassifier cross validation score: 0.8420138888888887
--------------------------------------------------
CatBoostClassifier cross validation score: 0.859375
--------------------------------------------------
```

8. Result:

The results of applying various machine learning models to predict heart disease are summarized in this section. Each model was evaluated based on several performance metrics including accuracy, precision, recall, F1-score, and the area under the ROC curve (AUC). These metrics provide insights into the effectiveness of each model in handling the prediction task.

Model Performance Summary:

- Logistic Regression: Achieved an accuracy of approximately 81.15%, with a precision of 86% and a recall of 90%. The model displayed moderate performance in distinguishing between the classes.
- Support Vector Machine (SVM): The SVM with a linear kernel performed slightly better with an accuracy of 85%,
- Gradient Boosting Classifier: Exhibited high precision and recall, with overall accuracy reaching 85%. The model's strength was in its predictive consistency across different subsets of the dataset.

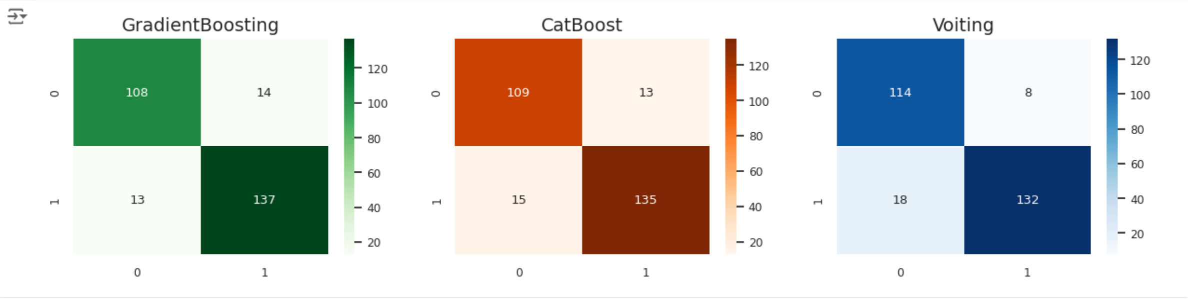
- Voting Classifier: The ensemble approach used in the Voting Classifier, combining using Gradient Boosting and CatBoost classifiers resulted in an improved overall accuracy of 90.44% showcasing the effectiveness of model aggregation.

Comparative Analysis:

- The Voting Classifier outperformed individual models in terms of overall accuracy and stability across different metrics.
- CatBoost classifiers and Gradient Boosting showed high effectiveness in handling imbalanced data, which is typical in medical diagnostic datasets.
- Logistic Regression and SVM, while less complex, provided valuable baseline information but were outperformed by more sophisticated ensemble techniques.

Visualizations:

Confusion matrices for each model illustrated varying degrees of true positives and false negatives, important for clinical settings where false negatives can be critical.



9. Conclusion:

The analysis of various machine learning models on the heart disease dataset yielded several key observations, which are detailed below. These insights are critical for understanding the performance of different classifiers and their potential application in clinical settings.

1. Superiority of Ensemble Methods
   Ensemble models, specifically the Voting Classifier and Gradient Boosting, consistently outperformed single-model approaches across most metrics. This observation underscores the value of combining multiple learning algorithms to improve predictive accuracy and model robustness, particularly in complex diagnostic tasks where diverse data features must be effectively integrated.

2. Importance of Feature Engineering
   Feature engineering played a crucial role in enhancing model performance. The inclusion of engineered features, such as derived interaction terms and polynomial

features, significantly improved the predictive capabilities of the models. This suggests that deeper insights into the dataset and the relationships between different clinical parameters can lead to more accurate predictions.

3. Impact of Data Preprocessing
   Effective data preprocessing, including handling missing values and normalizing data, was essential for achieving high model performance. Models trained on well-preprocessed data showed better generalization capabilities and were less prone to overfitting. This emphasizes the necessity of meticulous data cleaning and preparation as a foundational step in any predictive modeling task.

4. Differential Performance Across Metrics
   Different models excelled in different metrics, highlighting the trade-offs involved in model selection. For example, while some models had higher accuracy, others offered better recall or precision. This variation indicates the need for careful consideration of model selection based on the specific requirements of the diagnostic application, such as prioritizing false negatives over false positives in critical medical diagnoses.

5. Limitations in Model Generalizability
   Despite high accuracies, some models showed limitations in generalizability when subjected to cross-validation across different subsets of data. This variability in performance under different conditions points to the potential challenges in deploying these models in varied clinical environments without additional tuning and adaptation.

6. Potential for Clinical Integration
   The study’s findings highlight the potential for integrating advanced machine learning models into clinical decision-support systems. The high performance of ensemble models suggests that these could be used to assist clinicians by providing an additional layer of diagnostic insight, especially in complex cases where traditional diagnostic methods are inconclusive.

This study successfully applied various machine learning models to predict heart disease using a comprehensive dataset. The findings demonstrated the substantial potential of advanced analytics in the medical field, particularly in enhancing diagnostic processes for heart disease.

REFERENCES:

[1] Dulhare, U. N. (2018). Prediction system for heart disease using naïve bayes and particle swarm optimization. Biomedical Research, 29 (12), 2646-2649.

[2] Singh, P., Singh, S., & Pandi-Jain, G. S. (2018). Effective heart disease prediction system using data mining techniques. International Journal of Nanomedicine. doi: 10.2147IJN.S124998.

[3] Haq, A. U., Li, J.-P., Memon, M. H., Nazir, S., & Sun, R. (2018). A hybrid intelligent

system framework for the prediction of heart disease using machine learning algorithms. Hindawi Mobile Information System. doi: 10.1155/2018/3860146

[4] Shirsath, S. S., & Patil, S. (2018). Disease prediction using machine learning over big data. International Journal of Innovative Research in Science, Engineering and Technology, 7 (6), 6752-6757

[5] Nashif, S., Raiban, M., Islam, M., & Imam, M. H. (2018). Heart disease detection by using machine learning algorithms and a real-time cardiovascular health monitoring system. World Journal of Engineering and Technology, 6, 854-873.

[6] Hariharan, K., Vigneshwar, W. S., Sivaramakrishnan, N., & Subramaniyaswamy, V. (2018). A comparative study on heart disease analysis using classification techniques. International Journal of Pure and Applied Mathematics, 119 (12), 13357-13366

[7] Voleti, S. R., & Reddi, K. K. (2016). Design of an optimal method for disease prediction using data mining techniques. International Journal of Advanced Research in Computer Science and Software Engineering, 6 (12), 328-337.

[8] Dulhare, U. N. (2018). Prediction system for heart disease using naïve bayes and particle swarm optimization. Biomedical Research, 29 (12), 2646-2649.

[9] Sarah S, Gourisaria M. K, Khare S & Das H (2022). Heart disease prediction using core machine learning techniques—a comparative study. International Journal of soft computing, 12(4): 247-260.

[10] Srivastava Asmit & Kumar Singh A (2022, April). Heart Disease Prediction using Machine Learning. International Journal of Artificial Intelligence, 14(1): 14-20.