**Heart Attack Analysis**

Lalitha Priya Bijja1  
Computer Science Department  
University of South Dakota Vermillion, South Dakota  
[lalithapriya.bijja@coyotes.usd.edu](mailto:lalithapriya.bijja@coyotes.usd.edu)

Venkata Surya Deepak Lakshimpalli2  
Computer Science Department  
University of South Dakota Vermillion, South Dakota  
venkatasuryadeepak.l@coyotes.usd.edu

***Abstract*** - This report presents an analysis of a heart health dataset aimed at predicting the risk of heart disease. The dataset, containing 303 entries and 14 columns, underwent exploratory data analysis (EDA) to understand its structure and relationships. Preprocessing steps included handling missing values, one-hot encoding categorical variables, and standardizing numerical features. Two predictive models were built and evaluated: a Random Forest Classifier and a Convolutional Neural Network (CNN). The Random Forest model achieved a training accuracy of 100% and a testing accuracy of approximately 81%. Cross-validation yielded a mean accuracy of about 82%. The CNN model attained a testing accuracy of around 61%. Evaluation metrics such as precision, recall, F1-score, and confusion matrices were computed for both models. Recommendations for future work include exploring additional feature engineering techniques, experimenting with different algorithms, and collaborating with domain experts for model refinement and interpretation.

**Key words** – Random Forest classifier, Convolutional Neural Network (CNN), K-Fold, Cross – Validation.

**Introduction**

Heart disease is one of the leading causes of mortality worldwide. Early detection and accurate prediction of heart attacks can significantly improve patient outcomes. In this report, we explore a dataset containing various clinical features and use machine learning techniques, including Random Forest and Convolutional Neural Networks (CNNs), to analyze and predict the risk of heart attacks. By harnessing the power of these advanced algorithms, we seek to uncover insights that could potentially revolutionize the way heart disease is diagnosed and managed, ultimately contributing to better patient care and outcomes.

**Dataset Exploration and Preprocessing**

**Dataset Description**

The dataset used in this analysis contains a comprehensive set of clinical features related to heart health, including demographic information, medical history, and various physiological measurements. Each instance in the dataset is labeled with the presence or absence of a heart attack ('output').

The analysis begins with the loading of essential libraries, including pandas (pd), NumPy (np), matplotlib (plt), seaborn (sns), scikit-learn (sklearn), and TensorFlow (tf), facilitating data manipulation, visualization, modeling, and deep learning tasks. The heart disease dataset, crucial for our analysis, is then imported from a CSV file named "heart.csv" using pandas' read\_csv function. Moving on to exploratory data analysis (EDA), we delve into understanding the dataset's characteristics comprehensively. This involves examining basic statistics such as central tendency, spread, and distribution of features through the describe function.

Additionally, we utilize the info method to gain insights into the data structure and the types of features present. To identify any potential data quality issues, we employ the isnull().sum method to check for missing values. Visual exploration of the dataset includes creating countplots from seaborn to showcase the distribution of categorical features like sex and chest pain type.

Furthermore, we generate pair plots using seaborn to explore relationships between numerical features such as age and blood pressure. This thorough exploration sets the foundation for subsequent data preprocessing and model building stages.

**Feature Engineering and Preprocessing**

To prepare the data for model training, we performed feature engineering and preprocessing steps. Categorical variables were one-hot encoded to convert them into numerical representations, while numeric features were standardized to ensure uniform scale and distribution. Visualization techniques such as box plots and swarm plots were employed to assess the impact of feature scaling on the data distribution.

**Machine Learning Model: Random Forest**

Following the data exploration phase, the dataset undergoes a crucial step known as the train-test split, facilitated by the train\_test\_split function from scikit-learn. This partitioning separates the data into two subsets: a training set utilized to train the model and a testing set employed to evaluate the model's performance on unseen data, thus ensuring its generalizability. Moving forward, model building commences with the initialization of a Random Forest Classifier model from scikit-learn. Random Forest, an ensemble learning technique, amalgamates multiple decision trees to enhance predictive accuracy and robustness. Hyperparameters such as the number of trees (n\_estimators) and random state (random\_state) are specified to fine-tune the model's behavior. Subsequently, the model undergoes training on the training data via the fit method. Throughout this process, the model assimilates patterns from the data to make predictions regarding the risk of heart attacks, the target variable under scrutiny. Model evaluation ensues, encompassing an array of assessment metrics to gauge the model's efficacy comprehensively. These metrics include the accuracy score, which quantifies the proportion of correct predictions made by the model on both the training and testing sets. Furthermore, an ROC curve is generated using roc\_curve to illustrate the model's discriminatory prowess between individuals with and without heart disease across various classification thresholds. A detailed classification report employing classification\_report furnishes insights into metrics such as precision, recall, and F1-score, offering a holistic understanding of the model's performance. Additionally, a confusion matrix visualizes the distribution of correct and erroneous predictions for each class, aiding in the identification of potential biases or class imbalances. Lastly, the optional application of KFold cross-validation via cross\_val\_score provides a robust estimate of the model's generalizability by iteratively training and evaluating the model on multiple folds of the data, further bolstering confidence in its performance metrics.

**Deep Learning Model: Convolutional Neural Network (CNN)**

For the Convolutional Neural Network (CNN) model, data preprocessing constitutes a crucial initial step, ensuring compatibility with the CNN architecture's input requirements. This involves reshaping the data to prepare it for CNN input. Given CNNs' proficiency in handling sequential data, features are typically reshaped into a 3D tensor, with dimensions representing samples, features, and steps if applicable. Moving on to model architecture, a CNN architecture is meticulously crafted utilizing the sequential model framework from TensorFlow. CNNs are revered for their adeptness in capturing spatial patterns within data, making them an ideal choice for tasks like image classification. The architecture is anticipated to encompass convolutional layers for feature extraction, pooling layers for dimensionality reduction, flattening layers to transform the data into a 1D vector, and dense layers for classification. Activation functions such as Rectified Linear Unit (ReLU) and dropout regularization are likely employed throughout the network to enhance model performance and mitigate overfitting. Upon defining the model architecture, the next step involves model compilation. Here, the model is compiled with the Adam optimizer, a popular choice known for its efficiency in training deep neural networks. The binary cross-entropy loss function is selected, suitable for binary classification problems like the prediction of heart disease risk. Additionally, the accuracy metric is designated to monitor the model's performance during training and evaluation. This meticulous configuration of the model's components lays the foundation for subsequent training and evaluation stages, ensuring optimal performance and robustness in predicting heart disease risk.

**Conclusion**

In conclusion, both the Random Forest and Convolutional Neural Network (CNN) models were trained and evaluated for the task of predicting heart disease risk. While both models showed promise, the Random Forest model exhibited superior accuracy compared to the CNN model. Despite this discrepancy, it's important to acknowledge that further experimentation and optimization of the models could potentially enhance their predictive performance. By fine-tuning hyperparameters, exploring alternative architectures, and incorporating additional features or data augmentation techniques, we may uncover opportunities to bolster the models' efficacy in predicting heart disease risk with greater precision and reliability. As such, ongoing refinement and exploration remain imperative for maximizing the utility of machine learning techniques in advancing cardiovascular health assessment and management.

**Future work**

In future endeavors, several avenues for advancing the predictive capabilities of heart disease risk assessment models can be explored. Incorporating more advanced techniques such as hyperparameter tuning and ensemble learning holds promise for enhancing model performance. By systematically optimizing model parameters and leveraging the collective wisdom of multiple models, we can potentially achieve heightened predictive accuracy and robustness. Furthermore, investigating additional datasets or features that offer valuable insights into heart health prediction could enrich the model's understanding and predictive capacity. Collaboration with domain experts remains paramount, as their insights can inform model refinement and facilitate the effective interpretation of results. Additionally, the exploration of additional feature engineering techniques presents an opportunity to extract more relevant information from the data, further augmenting model performance. Experimentation with a diverse array of machine learning and deep learning algorithms is also warranted to identify the most suitable and effective modeling approach. Moreover, efforts to gather more data or enrich the dataset through feature engineering endeavors can provide a broader and more representative basis for prediction, ultimately yielding more accurate and reliable results. Through these concerted efforts, we can continually advance the field of heart disease risk assessment and contribute to improved patient outcomes and healthcare practices.

**References**

1. American Heart Association. (2022). Heart Disease and Stroke Statistics—2022 Update: A Report from the American Heart Association. [Link](https://www.heart.org/-/media/data-import/downloadables/heart-disease-and-stroke-statistics-2022---at-a-glance---english/pdf)

2. Kaggle: Heart Attack Analysis & Prediction Dataset. [Link](https://www.kaggle.com/code/kanncaa1/heart-attack-analysis-prediction)

3. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830. [Link](https://jmlr.org/papers/v12/pedregosa11a.html)

4. Chollet, F. (2015). Keras. [Link](https://keras.io/)

5. TensorFlow Documentation. [Link](https://www.tensorflow.org/guide)

6. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Science & Business Media.

7. Brownlee, J. (2020). Introduction to Time Series Forecasting with Python: How to Prepare Data and Develop Models to Predict the Future. Machine Learning Mastery.

8. Understanding Convolutional Neural Networks for NLP. [Link](https://towardsdatascience.com/understanding-convolutional-neural-networks-for-nlp-5d2bf6e56db8)

9. TensorFlow Tutorials. [Link](https://www.tensorflow.org/tutorials)

10. Karpathy, A. (2016). Convolutional Neural Networks for Visual Recognition. [Link](http://cs231n.stanford.edu/)