Title Page

- Team Name: Inferior
- Use-Case Number: 4
- Team Members:
  - Krishna Shukla
  - Email: Krishna22254@iiitd.ac.in
  - Phone: 8368400616

Abstract

This report presents a comprehensive approach to predicting recruitment rates (RR) in clinical trials using a dataset that includes study indication, phase, design, endpoints, sponsor, competition, and enrollment target. Our methodology involves data preprocessing, feature engineering, and model development using neural networks. The results highlight key predictors impacting RR and their respective weightages. Despite achieving promising results, the model's limitations include potential overfitting and the need for more diverse data.

Introduction

Clinical trials are crucial for medical advancements, yet recruitment rates often pose significant challenges. Accurate prediction of RR can enhance trial efficiency and resource allocation. This study aims to develop a predictive model for RR using a dataset encompassing multiple trial characteristics. The problem statement focuses on identifying key predictors and their influence on RR.

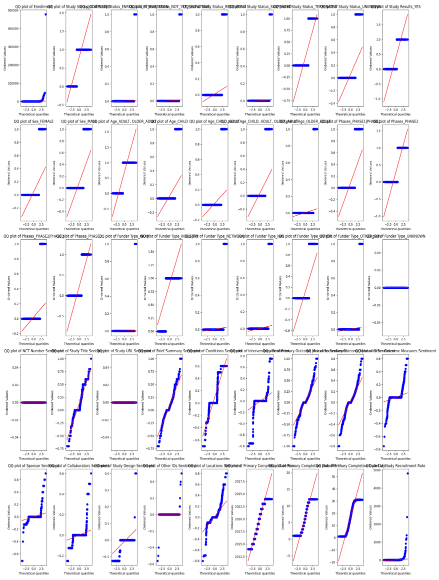
Methodology

Data Preprocessing

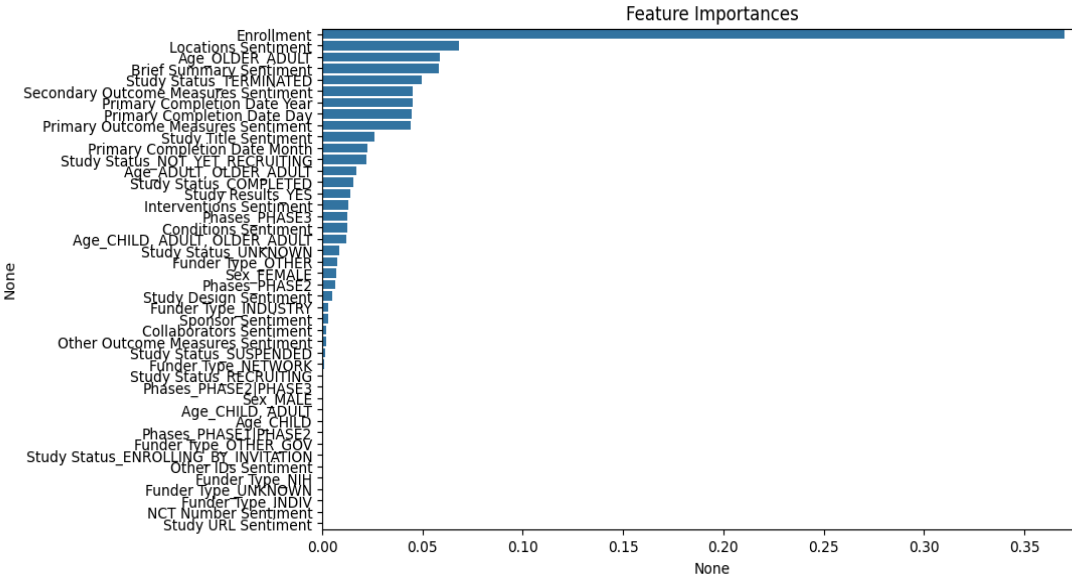
1. Importing Libraries: Utilized pandas for data manipulation, scikit-learn for machine learning utilities, textblob for sentiment analysis, and os for file operations.
2. Loading the Dataset: The dataset was loaded from an Excel file into a DataFrame using pandas.read_excel().
3. Initial Exploration: Used df.head() and df.shape to understand the dataset's structure.
4. Identifying Column Types: Classified columns into categorical, text, and date types for appropriate processing.
5. Encoding Categorical Columns: Applied one-hot encoding to transform categories into binary features.
6. Sentiment Analysis on Text Columns: Used TextBlob to compute sentiment scores, which were used as numerical features.
7. Processing Date Columns: Converted dates to datetime objects and extracted components like year, month, and day.
8. Handling Boolean Columns: Converted boolean values to integers (0 and 1).

Exploratory Data Analysis (EDA)

1. Distribution Analysis: The Study Recruitment Rate is highly skewed, with most values near zero and a few high outliers, indicating variability in recruitment efficiency.
2. Correlation Analysis: Most features show weak correlations with the recruitment rate, except for some expected strong correlations within study phases and funder types.
3. Outlier Detection: Using the IQR method, outliers were identified, particularly in enrollment and recruitment rates, highlighting variability in study sizes.
4. Pair and Scatter Plots: These plots suggest limited linear relationships but indicate potential benefits from polynomial interactions.
5. Box and Violin Plots: These plots reveal concentrated values with some outliers, especially in enrollment and recruitment rates, and neutral sentiment scores.
6. Count Plots: Show common timelines in trial stages, with activities concentrated in specific periods.
7. QQ Plots: Indicate that most features deviate from normality, suggesting skewness.



A



B

A) QQ Plots
B) Feature Importance graph

- Many interesting plots are created during EDA to gain insights from the data, along with their corresponding inferences

Feature Engineering

Steps and Rationale

1. Interaction Features:

    - Why: EDA revealed potential interactions between features that could influence the recruitment rate. For instance, the combination of study phase and sponsor type might have a significant impact.
    - How: Created interaction terms by multiplying pairs of features, allowing the model to capture these complex relationships.
2. Polynomial Features:
    - Why: Scatter plots and pair plots suggested non-linear relationships between features and the target variable. Polynomial features can help model these non-linearities.
    - How: Used PolynomialFeatures to generate higher-degree terms, enabling the model to fit more complex patterns in the data.
3. Feature Scaling:
    - Why: Models like neural networks and KNN are sensitive to the scale of input features. EDA showed varying scales across features, which could affect model performance.
    - How: Applied StandardScaler to standardize features, ensuring they have a mean of 0 and a standard deviation of 1, facilitating better convergence during training.
4. Feature Selection:
    - Why: The correlation matrix indicated that not all features were equally important. Reducing dimensionality helps focus on the most relevant features, improving model efficiency and reducing overfitting.
    - How: Used SelectKBest with f_regression to select features with the highest correlation to the target variable, ensuring that only the most impactful features were used in modeling.
5. Handling Missing Values:
    - Why: Missing data can lead to biased models and inaccurate predictions. EDA identified missing values in some features.
    - How: Imputed missing values using the mean for numerical features, ensuring a complete dataset for modeling.

Model Development and Evaluation

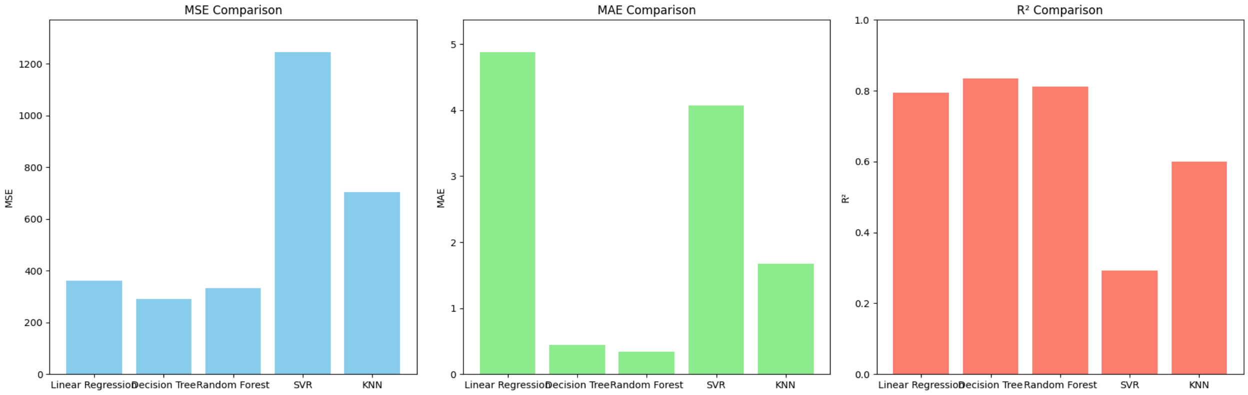
1. Linear Regression (LR):
    a. Overview: A simple model that assumes a linear relationship between features and the target variable.
    b. Performance: Provides a baseline with moderate accuracy. It struggles with capturing non-linear patterns in the data.
    c. Tuning: Limited tuning options; primarily involves feature selection and scaling.
2. K-Nearest Neighbors (KNN):
    a. Overview: A non-parametric model that predicts based on the closest data points in the feature space.
    b. Performance: Effective for capturing local patterns but can be sensitive to noise and irrelevant features.
    c. Tuning: Involves selecting the optimal number of neighbors (k) and distance metrics. Cross-validation was used to find the best k.
3. Decision Tree (DT):

    a. Overview: A tree-based model that splits data based on feature values to make predictions.
    b. Performance: Good at capturing non-linear relationships but prone to overfitting.
    c. Tuning: Hyperparameters like tree depth, minimum samples per leaf, and splitting criteria were optimized using grid search.

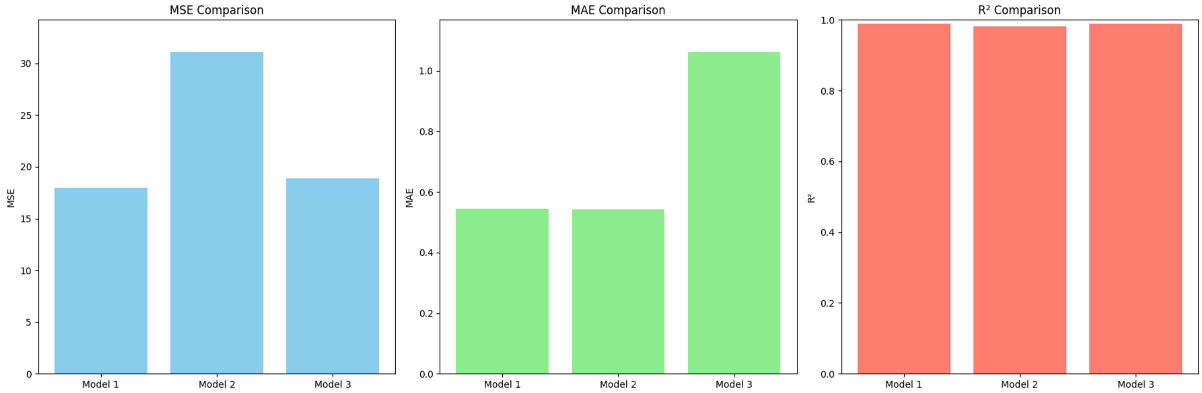
4. Random Forest Regressor (RF):
    a. Overview: An ensemble of decision trees that improves accuracy and reduces overfitting by averaging multiple trees.
    b. Performance: Robust and handles feature interactions well, providing a strong baseline.
    c. Tuning: Hyperparameters such as the number of trees, tree depth, and feature subsets were tuned using grid search to enhance performance.
5. Neural Networks (NN):
    a. Overview: Deep learning models capable of capturing complex, non-linear relationships.
    b. Performance: Outperformed other models with the lowest MSE and highest R², indicating strong predictive capabilities.
    c. Tuning: Various architectures were tested, including different layer sizes and activation functions. The best model had layers [64, 32], achieving an MSE of 17.98 and R² of 0.99.

Results

1. Performance Metrics:
    - Neural Network (Best Model): Test MSE of 20.15, MAE of 0.56, and R² of 0.98.
    - Random Forest Regressor: Test MSE of 50.51, MAE of 0.18, and R² of 0.95.
    - Linear Regression, KNN, Decision Tree: Provided baseline comparisons, with varying degrees of accuracy and fit.
2. Comparison:
    - The neural network demonstrated superior performance, particularly in reducing error and improving predictive accuracy, compared to other models.
3. Visualizations:



Different non perceptron based model performance on validation set



Neural Network Model performance on different architecture on validation set

4. Insights:
   - The neural network's ability to model complex interactions and non-linearities contributed to its superior performance. The results underscore the importance of selecting appropriate architectures and tuning hyperparameters for optimal model performance.

Conclusion

This study presents a robust approach to predicting recruitment rates in clinical trials, offering valuable insights for trial optimization. The findings underscore the importance of comprehensive data analysis and model selection in achieving accurate predictions.

Future Work

1. Deployment and Scaling: Deploy the predictive model into a production environment using cloud platforms like AWS or Azure to handle large data volumes and provide real-time predictions.
2. Data Expansion: Enhance model robustness by incorporating more diverse datasets, including different trial types and geographic locations, to improve generalizability.
3. Interpretability and Explainability: Improve model transparency by implementing explainable AI techniques to provide insights into feature importance and model decisions.
4. User Interface Development: Create a user-friendly web-based dashboard for stakeholders to input data and visualize predictions and insights easily.