ML Project Report

Patient Recovery Prediction Challenge

(Checkpoint 1)

Team Members

1. KNV Aditya – IMT2023033
2. Abhijit Dibbidi – IMT2023054
3. Adwait Priyadarshan – IMT2023502

Github Link

https://github.com/erakin027/Patient-Recovery-Prediction-Challenge/

Task

The primary objective of this project is to develop and compare multiple regression models to accurately predict a patient's Recovery Index, an integer value ranging from 10 to 100. This represents a classic supervised regression problem where the goal is to build predictive models using a training dataset containing patient health metrics and their corresponding recovery outcomes. The trained models are then used to predict recovery indices for a testing dataset that contains all input features except the target variable. The ultimate aim is to identify, tune, and deliver the best- performing regression model for predicting patient recovery outcomes.

Dataset Description

Description:

The Patient Recovery Dataset is designed to examine the factors influencing patient recovery outcomes. It contains 10,000 patient records, with each record including medical and lifestyle-related predictors, along with an overall Recovery Index.
The objective of this challenge is to build predictive models that can estimate a patient's recovery progress based on their treatment and lifestyle factors.

The training dataset train.csv consists of 8000 rows and 7 columns, with the testing dataset test.csv on which recovery index needs to be predicted is of size 2000 x 6.

Variables:

- Id(int): a unique identification number for a patient

- Therapy Hours(int): The total number of hours a patient spent in therapy sessions.
- Initial Health Score(int): The patient's health assessment score recorded during their first check-up.
- Lifestyle Activities: Whether the patient engaged in additional healthy lifestyle activities (Yes or No) – binary encoded to 1 and 0 in data preprocessing step
- Average Sleep Hours(int): The average number of hours the patient slept per day.
- Follow-Up Sessions(int): The number of follow-up sessions the patient attended.

Target Variable:

- Recovery Index: A measure of the overall recovery progress of each patient. The index ranges from 10 to 100, with higher values indicating better recovery outcomes. It has been rounded to the nearest integer.

Pre-processing & EDA

The following section outlines the necessary steps for conducting essential Exploratory Data Analysis (EDA) to gain insights, detect potential issues, and prepare data for model training.

1. Importing Libraries and Loading the Dataset

The first step is to import the necessary libraries, including pandas, numpy, matplotlib, seaborn, and relevant modules from scikit-learn for preprocessing tasks. The dataset is then loaded into a pandas DataFrame for further inspection and manipulation.

2. Initial Data Exploration

The dataset is analyzed by displaying the first few rows, checking the data types of each column, and identifying any missing values. This helps in gaining an initial understanding of the structure and quality of the data. A statistical summary of the numerical columns is also generated to assess the central tendency, variability, and potential outliers.

3. Handling Missing Values and Duplicates

To address missing values, we thoroughly examined the dataset for any null values. It was clear that no NaN values were present, so, no imputation or removal of data was necessary. This ensures the dataset is complete and does not require any modifications in terms of missing data, which could otherwise impact model performance.

Similar thorough checks were done for duplicate records, and none for found, so no removal of data was necessary. (if found we would've deleted the duplicates using drop_duplicates() method in Pandas library)

4. Data Preprocessing

At this stage, the dataset is prepared for modeling by completing the following tasks:

- **Encoding Categorical Variables:** Categorical variables are encoded using label encoding to convert them into a numerical format suitable for machine learning algorithms. This step ensures that the data can be interpreted by models.

  Lifestyle Activities is such categorical variable in which “Yes” is encoded to 1 and “No” to 0 in this step.

- **Dropping Useless Features:** In this step, useless features which have little to no relation with Recovery Index, and Recovery Index itself (as it itself needs to be predicted) are removed to train the model.

  Here the feature “Id” is dropped as it has no impact on the Recovery Index of a patient, for it is just an identifier. Having it would just introduce noise and mess up the model. This could be supported by basic common sense, and also by a correlation heat map, which comes later in EDA.

- **Standardizing Numerical Features**: Numerical features were standardized using StandardScaler from scikit-learn to ensure all features have similar scales with mean of 0 and standard deviation of 1. This standardization is crucial for distance-based algorithms and regularized linear models, as it prevents features with larger scales from dominating the model.

  Two versions of the data were maintained

  Unstandardized data (X_train, X_test): Used for tree-based models (Decision Tree, Random Forest, XGBoost) which are scale-invariant.

  Standardized data (X_train_scaled, X_test_scaled): Used for linear models (Linear Regression, Ridge, Lasso, Elastic Net) to ensure fair coefficient estimation.

5. Exploratory Data Analysis (EDA)

Various visualizations were created to analyze the data distribution, detect outliers, and explore relationships between features. These visualizations provide important insights that guide further preprocessing and modeling decisions. Below are key visualizations and their interpretations:

- **Correlation Matrix (Heatmap):** A correlation matrix heatmap was generated to explore relationships between variables. Notable observations include:
  - Id: Id shows a very weak correlation with Recovery Index (0.0088), suggesting that it has little to no relation to Recovery Index, as we have discussed earlier.
  - Initial Health Score: Initial Health Score has a high positive correlation with Recovery Index (0.91), revealing a strong relation between the two.
  - Therapy Hours shows a moderate correlation of 0.38 with Recovery Index.

    - Other features like Lifestyle Activities (0.019), Average Sleep Hours (0.044) and Follow-up sessions (0.044) have a pretty weak correlation with Recovery Index.

- **Scatter Plots:** For all features, scatter plots were plotted vs Recovery Index to observe the data's relation to it. Some key observations:
    - All the features have linear relation with Recovery Index, as in a straight line can be visualised through the plot.
    - This is particularly visible in the plot between Recovery Index vs Initial Health score, which has the highest correlation with Recovery Index as found out in the correlation map.

  This linear relation might suggest that a linear regression model to be the best fit for the data.

- **Boxplots for Outlier Detection:** Boxplots were used to examine each numerical feature for outliers, represented as points outside the plot's whiskers. No significant outliers were detected, meaning the data is clean without any extreme values that could potentially skew model results.

6. Splitting the Dataset

The dataset is divided into training and testing sets, using train_test_split typically using an 80/20 split (test_size = 0.2). This resulted in 6,400 samples for training and 1,600 samples for validation, enabling proper model evaluation before final testing. This ensures that the model can be trained on one portion of the data and evaluated on an unseen portion, helping to assess its generalization ability.

This is done using the scaled data we got from standardisation for linear models, and unscaled data for tree models.

Key Observations from EDA

The exploratory analysis revealed several critical insights that guided model selection and development:

Linear Relationships: All features demonstrated approximately linear relationships with the Recovery Index, suggesting that linear regression and regularized linear models would perform well on this dataset.

Feature Importance Hierarchy: Initial Health Score emerged as the dominant predictor, followed by Therapy Hours, with other features showing weaker but non-negligible contributions.

Data Quality: The absence of missing values, duplicates, and outliers indicated high data quality, reducing the need for complex preprocessing strategies.

Scale Considerations: The varying scales of features necessitated standardization for linear models while allowing tree-based models to work with raw features.

These observations informed the decision to test both linear models (which could capture the observed linear relationships efficiently) and tree-based ensemble methods (which could capture potential non-linear interactions and provide robustness).

Models Used

Multiple regression models were trained and evaluated to identify the optimal approach for predicting the Recovery Index. The models were categorized into two main groups: linear models and tree-based models, each with specific preprocessing requirements and evaluation strategies.

Linear Models

Linear regression models assume a linear relationship between input features and the target variable. These models were trained on standardized data to ensure fair coefficient estimation and proper regularization performance.

1. Linear Regression

Best Achieved Leaderboard Score in Test Data = 1.982

Linear regression serves as the baseline model, establishing a benchmark for comparison with more complex approaches. The model learns a linear combination of input features to predict the Recovery Index using ordinary least squares optimization.

The model achieved strong performance with an RMSE of 2.100407 and an R^2 score of 0.988 on the validation set, indicating that approximately 98.8% of the variance in Recovery Index could be explained by the linear combination of features.

2. Ridge Regression (L2 Regularization)

Best Achieved Leaderboard Score in Test Data = 1.982

Ridge regression extends linear regression by adding an L2 regularization penalty term that constrains the magnitude of coefficients. This technique helps prevent overfitting by shrinking coefficient values without setting any to exactly zero, making it suitable when all features are potentially relevant.

The Ridge model was initially trained with a default alpha value of 1, achieving an RMSE of 2.100279 and R^2 of 0.988. To optimize performance, GridSearchCV was employed to test alpha values of [0.01, 0.1, 1, 10, 100] using 5-fold cross-validation. The best alpha value was determined to be 0.1, resulting in a final RMSE of 2.100394 and R^2 of 0.988.

The minimal difference between Ridge and standard linear regression suggests that overfitting was not a significant concern for this dataset.

3. Lasso Regression (L1 Regularization)

Best Achieved Leaderboard Score in Test Data = 1.983

Lasso regression employs L1 regularization, which can shrink coefficients to exactly zero, effectively performing automatic feature selection.

Initial training with alpha = 0.1 yielded an RMSE of 2.110 and R^2 of 0.988. GridSearchCV was applied with alpha values of [0.0001, 0.001, 0.01, 0.1, 1], with 5-fold cross-validation identifying the optimal alpha as 0.001. The optimized model achieved an RMSE of 2.100380 and R^2 of 0.988, representing the best performance among all models tested.

The superior performance of Lasso with a small alpha value indicates that all features contribute meaningfully to predictions, with minimal benefit from aggressive feature elimination. The fact that no coefficients were reduced to zero suggests that the dataset's features are all relevant to recovery prediction.

4. Elastic Net Regression (L1 + L2)

Best Achieved Leaderboard Score in Test Data = 1.982

Elastic Net combines both L1 and L2 regularization penalties, attempting to balance the benefits of Ridge and Lasso regression. This hybrid approach can be advantageous when dealing with correlated features.

The model was trained with alpha = 0.1 and l1_ratio = 0.5, achieving an RMSE of 2.284 and R^2 of 0.986. The relatively lower performance compared to pure Ridge or Lasso suggests that the dataset does not significantly benefit from the combined regularization approach.

Then we used GridSearchCV for this too for values of alpha in logspace(-3,1,10) and li1_ration in linspace(0,1,10), and subsequently got the best score of 1.982 on test data. In this particular example of validation data, elastic net completely became lasso, and gave same RMSE and R^2 of it.

5. Polynomial Regression

Best Achieved Leaderboard Score in Test Data = 1.982

To capture potential non-linear relationships, polynomial regression was implemented by transforming features to include polynomial terms up to degree 2. This approach creates additional features representing interactions and quadratic terms while maintaining the linear regression framework.

The polynomial features were generated using PolynomialFeatures with degree=2 and include_bias=False, applied to the standardized training and validation data. The resulting linear regression model on these transformed features achieved an RMSE of 2.102845 and R^2 of 0.988.

The marginal performance difference compared to simple linear regression confirms the EDA finding that relationships in this dataset are predominantly linear. Higher polynomial degrees were deliberately avoided to prevent overfitting, as they would add unnecessary complexity without capturing meaningful patterns. This was verified when we used a grid with different values of degree from 1 to 5, and we found degree 1 to be the best fitting.

Tree-Based Models

Tree-based ensemble methods were evaluated as alternatives to linear models. These models were trained on unstandardized data, as decision trees are invariant to feature scaling and make decisions based on threshold splits rather than distance calculations.

6. Decision Tree Regressor

Best Achieved Leaderboard Score in Test Data = 2.426

An initial model with default hyperparameters achieved an RMSE of 3.043 and R^2 of 0.975 on the validation set. To improve performance, GridSearchCV was applied with the following parameter grid:

max_depth: [None, 5, 10, 20]

min_samples_split: [2, 5, 10]

min_samples_leaf: [1, 2, 4]

The 5-fold cross-validation identified optimal parameters as max_depth=10, min_samples_split=10, and min_samples_leaf=4, improving performance massively to RMSE of 2.532 and R^2 of 0.982. While this represents substantial improvement over the default configuration, it remained inferior to linear models.

The relatively lower performance of decision trees compared to linear models aligns with the predominantly linear relationships observed in the data, suggesting that the hierarchical splitting approach does not provide advantages for this particular dataset.

7. Random Forest Regressor

Best Achieved Leaderboard Score in Test Data = 2.276 → 2.195

The initial model with 100 estimators and default parameters achieved an RMSE of 2.416 and R^2 of 0.984. GridSearchCV was employed and cross-validation identified optimal parameters as

n_estimators=200, max_depth=None, min_samples_split=2, min_samples_leaf=2, and max_features='sqrt',

achieving an RMSE of 2.415 and R^2 of 0.984.

Feature importance analysis using the trained Random Forest revealed that Initial Health Score dominated with the highest importance score, followed by Therapy Hours, confirming

the patterns observed in correlation analysis. This analysis provided additional validation of the feature hierarchy established during EDA.

8.XGBoost (Extreme Gradient Boosting)

Best Achieved Leaderboard Score in Test Data = 2.001

Initial manual hyperparameter tuning explored various configurations of n_estimators, learning_rate, max_depth, subsample, and colsample_bytree. A configuration with n_estimators=200, learning_rate=0.05, max_depth=5, subsample=0.8, and colsample_bytree=0.8 achieved an RMSE of 2.175 and R^2 of 0.987.

Systematic GridSearchCV optimization was then performed and the optimal parameters were determined as

n_estimators=200, max_depth=3, learning_rate=0.1, subsample=1, and colsample_bytree=0.8,

resulting in an RMSE of 2.160 and R^2 of 0.987. While XGBoost outperformed other tree-based models, it still fell short of the best linear models.

This outcome is notable because XGBoost typically excels in machine learning competitions and complex datasets. The relatively modest performance suggests that the dataset's predominantly linear relationships are better captured by simpler linear models, and the additional complexity of gradient boosting does not provide substantial benefits.

Note: Best Achieved Leaderboard Score in Test Data might not be based on the best parameters of GridSearchCV given above either. This is because we also did some trail and error on some parameters apart from this, which gave the best results for some models like XGBoost.

Model Comparison

A comprehensive comparison of all trained models reveals important insights about the dataset characteristics and optimal modeling approaches.

Validation Set Performance Rankings

When ranked by validation set performance, the models achieved the following results:

1. Ridge Regression: RMSE = 2.100279, R^2 = 0.988 (best)
2. Lasso Regression: RMSE = 2.100380, R^2 = 0.987902
3. Elastic Net: RMSE = 2.100380, R^2 = 0. 987902
4. Linear Regression: RMSE = 2.100407, R^2 = 0.987902
5. Polynomial Regression: RMSE = 2.102845, R^2 = 0.987874
6. XGBoost: RMSE = 2.159942, R^2 = 0.987206

7. Random Forest: RMSE = 2.415086, R^2 = 0.984005

8. Decision Tree: RMSE = 2.531674, R^2 = 0.982424 (Worst)

Test Data Leaderboard Performance Rankings

When ranked by leaderboard performance, the models achieved the following results:

1. Ridge Regression: 1.982 (Best)

2. Linear Regression: 1.982

3. Elastic Net: 1.982

4. Polynomial Regression: 1.982

5. Lasso Regression: 1.983

6. XGBoost: 2.001

7. Random Forest: 2.195

8. Decision Tree: 2.426 (Worst)

The top positions are occupied by linear models, with differences of less than 0.003 in RMSE between them. This minimal variation suggests that all linear approaches effectively captured the underlying relationships in the data. With just minor non deterministic changes like changing randomstate, it could be observed that the top 2(linear, ridge) are almost undistinguishably good, as any of them could top the list based on the seed.

Key Observations and Interpretations

1. Dominance of Linear Models: The superior performance of linear regression and its regularized variants confirm the EDA finding that features exhibit predominantly linear relationships with the Recovery Index. The simplicity of linear models proves advantageous when the true underlying relationship is indeed linear, as complex models risk overfitting noise without capturing additional info.

2. Regularization Impact: The marginal differences between Lasso, Ridge, and standard linear regression indicate that overfitting is not a significant concern for this dataset. The dataset's relatively large size (8,000 training samples) compared to the small number of features (5) provides sufficient data for stable coefficient estimation without strong regularization requirements.

3. Lasso vs Ridge Performance: Overall, Ridge regression gave better results in the test data, showing its superiority. Lasso achieved its best performance with a very small alpha value (0.001), suggesting minimal benefit from aggressive regularization. The fact that Lasso did not eliminate any features (no coefficients set to zero) indicates that all predictors contribute meaningfully to recovery prediction.

4. Tree-Based Model Limitations: The relatively poorer performance of tree-based models (Decision Tree, Random Forest, XGBoost) compared to linear models is instructive. While these algorithms excel at capturing complex non-linear relationships and feature interactions, they may be unnecessarily complex for datasets with predominantly linear patterns. The hierarchical splitting strategy of trees does not provide advantages when simpler linear combinations suffice.

5. XGBoost Expectations: XGBoost typically achieves state-of-the-art performance in machine learning competitions due to its sophisticated boosting mechanism and regularization capabilities. Its fifth-place ranking in this comparison highlights an important principle: model complexity should match problem complexity. The advanced features of XGBoost (sequential error correction, tree pruning, regularization) address challenges that are not present in this relatively straightforward dataset.

6. Polynomial Regression Insights: The near-identical performance of degree-2 polynomial regression compared to linear regression confirms that feature relationships are predominantly linear without significant quadratic effects or interactions. Higher polynomial degrees were deliberately avoided to prevent overfitting, as they would capture noise rather than meaningful patterns in a dataset where linear models already achieve a very high score.

Conclusion

Based on the leaderboard performance, linear regression (best leaderboard score of 1.982) emerges as the optimal model for this problem. However, the practical differences between the top two linear models (Linear, Ridge) are negligible, and as already discussed can interchange rankings by changing even non deterministic parameters like randomstate.

All three linear models offer computational efficiency, fast training times, and straightforward deployment compared to complex ensemble methods.

The strong performance of simple linear models validates the importance of thorough EDA. Understanding that features exhibit linear relationships with the target variable enabled appropriate model selection from the outset, demonstrating that more complex models are not always necessary or beneficial.