ML PROJECT REPORT: PATIENT RECOVERY INDEX DATASET

Team Members:
Sasank L (IMT2023120)
Akul Anhith (IMT2023558)

Team SKArmy
Github Link: https://github.com/aam2k6/ML-Course-Project

Contents

1 Introduction

The Patient Recovery Index Prediction project aims to estimate the recovery progress of patients based on a concise dataset of medical and lifestyle-related predictors. The dataset contains 10,000 patient records, which were split into a training set (8,000 records) and a test set (2,000 records) for this project.

The features capture key aspects of a patient's treatment and lifestyle, including `Therapy Hours`, `Initial Health Score`, `Lifestyle Activities`, `Average Sleep Hours`, and `Follow-Up Sessions`. By analyzing these variables, the project seeks to build a model that can accurately predict the `Recovery Index` (the target variable).

The dataset is framed as a supervised regression problem. Various machine learning algorithms, including linear models such as `LinearRegression`, `SGDRegressor`, `PolynomialRegression`, `Lasso`, and `Ridge`, as well as ensemble-based models like `RandomForestRegressor`,`GradientBoostingRegressor`, and `XGBRegressor`, were employed to build predictive models.

This report documents the full methodology, including exploratory data analysis, data preprocessing, model training, evaluation, and hyperparameter tuning. The ultimate objective is to deliver a robust predictive model that can accurately estimate a patient's recovery progress.

2 Exploratory Data Analysis

The data was explored to:

1. Gain a preliminary understanding of available data.
2. Check for missing or null values and duplicates.
3. Find potential outliers.
4. Assess correlations amongst features.
5. Check for data skew.

2.1 Preliminary Observations

After importing the `train.csv` dataset into a Pandas Data Frame, preliminary inspection revealed:

- **Shape:** (8000, 7)
- **Features:** 5 predictor variables (`Therapy Hours`, `Initial Health Score`, `Lifestyle Activities`, `Average Sleep Hours`, `Follow-Up Sessions`) and 1 identifier (`Id`).
- **Target Variable:** `Recovery Index` (an integer from 10 to 100).
- **Missing Values:** An inspection with `train.info()` confirmed **no missing or null values** in the 8,000 training records.

- **Duplicates:** An inspection with `train.duplicated().any()` confirmed **no duplicate rows**.
- **Target Variable Skew:** A boxplot of the target variable, `Recovery Index`, showed a symmetrical distribution, indicating that a log transformation to correct for skew (a common practice) was not necessary for this dataset.

2.2 Exploring Numerical Attributes

The numerical attributes (`Therapy Hours`, `Initial Health Score`, `Average Sleep Hours`, `Follow-Up Sessions`) were analyzed for their relationship with the `Recovery Index`.

- **Correlation Analysis:** A correlation heatmap revealed a **very strong positive correlation (0.91)** between `Initial Health Score` and `Recovery Index`. This indicates that the patient's initial health is the dominant predictor of their recovery. Other features showed weak to moderate positive correlations:
  - `Therapy Hours` (0.38)
  - `Follow-Up Sessions` (0.04)
  - `Average Sleep Hours` (0.04)
- **Visualization:** Scatter plots of each attribute against `Recovery Index` confirmed the strong, clear linear relationship between `Initial Health Score` and the target.
- **Distribution Analysis:** Histograms for all numerical *predictor* variables (e.g., `Therapy Hours`, `Initial Health Score`) show that they are all relatively **uniformly distributed**. This contrasts with the target variable, which is normally distributed.
- **Outlier Analysis:** Box plots were generated for all numerical features. No significant outliers were identified that required removal or special handling.

2.3 Exploring Categorical Attributes

The dataset contains one categorical variable, `Lifestyle Activities`, with two possible values: 'Yes' or 'No'.

- **Frequency Distribution:** A `value_counts()` analysis showed that the feature is almost perfectly balanced (4,043 'No' entries and 3,957 'Yes' entries).
- **Bivariate Analysis:** Boxplots of `Lifestyle Activities` vs. `Recovery Index` revealed a similar distribution in medians and quartile, suggesting this feature is not a strong predictor on its own. This balance was also observed when splitting the data by recovery score ($<=$ 55 and $>$ 55), indicating the feature is not heavily skewed by the outcome.

3 Data Preprocessing

Based on the EDA, the data was found to be very clean, requiring minimal preprocessing before modeling.

3.1 Feature Removal

The `Id` column was dropped from the training and test datasets as it is an identifier and not a predictive feature.

3.2 Train-Test Split

The 8,000-row training data was split into a training set (6,400 rows) and a validation set (1,600 rows) using an 80/20 split (`test_size=0.2`). This validation set (`x_test`, `y_test`) was used to evaluate the models after hyperparameter tuning.

3.3 Label Encoding

The single categorical feature, `Lifestyle Activities`, was converted into numerical values, using `sklearn.preprocessing.LabelEncoder` to make it compatible with the machine learning algorithms.

3.4 Feature Scaling

All features in both the training and validation sets were standardized using `StandardScaler`. This process (centering and scaling) ensures that all features contribute equally to the model's calculations, which is particularly important for regularized linear models (Lasso, Ridge) and gradient-based methods (SGD).

4 Feature Selection & Engineering

4.1 Feature Selection

All 5 preprocessed features (`Therapy Hours`, `Initial Health Score`, `Lifestyle Activities`, `Average Sleep Hours`, `Follow-Up Sessions`) were retained for modeling. .

The features with close to 0 correlation with Recovery Index could have been dropped, however, through trial and error, it was observed that models generally performed better with all features included.

4.2 Feature Engineering

For the `PolynomialRegression` model, new features were engineered by creating 2nd-degree polynomial and interaction terms from all 5 input features. This was done to test if a non-linear combination of features could capture relationships missed by the standard linear models.

5 Machine Learning Algorithm Assessment

5.1 Approach

A wide range of regression algorithms were trained and tuned to find the best-performing model. The primary method for hyperparameter tuning was `GridSearchCV` (for most models) and `RandomizedSearchCV` (for RandomForest and XGBoost).

All models were trained on the 6,400-row training set and evaluated using 5-fold cross-validation (`cv=5`) with `neg_mean_squared_error` as the scoring metric. The final R^2 score was then calculated on the unseen 1,600-row validation (test) set.

5.2 Validation R^2 Results

The performance of the best-tuned version of each model on the validation set is shown below.

Table 1: Model Performance on Validation Set

Model	**Test Set R^2**	**Best CV Score (Neg MSE)**
Linear Regression	**0.98790179**	-4.119
Lasso Regression	**0.98790353**	-4.120
Ridge Regression	**0.98790180**	-4.120
SGD Regressor	**0.98790025**	-4.120
Polynomial Reg. (Deg 2)	**0.98788455**	-4.127
XGBoost Regressor	0.98747545	-4.319
Gradient Boosting	0.98721481	-4.423
Random Forest	0.98580606	-4.978

6 Selection of Best Algorithm & Fine-Tuning

6.1 Approach to Selecting the Best Model

The results from Chapter 5 show that all models performed exceptionally well on our validation set, with R^2 scores above 0.985. The linear-based models (Linear, Lasso, Ridge, SGD, and Polynomial) all achieved nearly identical, top-tier cross-validation scores of $\approx$**0.9879**.

Interestingly, the 2nd-degree Polynomial Regression model achieved a slightly better score on the public leaderboard. However, this can be a sign of overfitting, as the public leaderboard only represents a subset of the full test data. Given that its cross-validation score was not superior to the simpler linear models, and its complexity is higher, a simple linear model is likely a more robust and generalizable choice.

This conclusion is strongly supported by the tuning results for Lasso and Ridge, which both converged on a tiny `alpha` of 0.01. An alpha this close to zero indicates that very little regularization is needed, and these models are effectively behaving just like a standard Linear Regression model. This confirms the data's relationship is inherently linear.

Therefore, **Lasso Regression** (which had the marginal-best R^2 of 0.98790353 in CV) or the standard **Linear Regression** model are the preferred choices, as they are the simplest, most interpretable, and most robust models.

6.2 Hyperparameter Tuning and Results

The following parameters were tuned, with the best-performing parameters identified by `GridSearchCV` or `RandomizedSearchCV`:

- **Linear Regression:**
  - *GridSearch Params:* `{'fit_intercept': [True, False], 'positive': [True, False]}`
  - *Best Params:* `{'fit_intercept': True, 'positive': True}`
  - *Public Leaderboard Score:* `1.982`
- **Lasso Regression:**
  - *GridSearch Params:* `{'alpha': [0.01, 0.1, 0.5, 1.0, 2.0]}`
  - *Best Params:* `{'alpha': 0.01}`
  - *Public Leaderboard Score:* `1.984`
- **Ridge Regression:**
  - *GridSearch Params:* `{'alpha': [0.01, 0.05, 0.09, 0.1, 0.5, 1.0, 2.0, 5.0]}`
  - *Best Params:* `{'alpha': 0.01}`
  - *Public Leaderboard Score:* `1.982`
- **SGD Regressor:**
  - *GridSearch Params:* `{'penalty': ['l2', 'l1', 'elasticnet', None], ...}`
  - *Best Params:* `{'alpha': 0, 'eta0': 0.001, 'penalty': 'l2'}`
  - *Public Leaderboard Score:* `1.982`
- **Polynomial Regression:**
  - *GridSearch Params:* `{'preprocess__poly__degree': [2, 3]}`
  - *Best Params:* `{'preprocess__poly__degree': 2}`

  - *Public Leaderboard Score:* `1.981`

- **Gradient Boosting:**
  - *GridSearch Params:* `{'n_estimators': [100, 200, 300], ...}`
  - *Best Params:* `{'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 200}`
  - *Public Leaderboard Score:* `2.029`

- **XGBoost Regressor:**
  - *RandomSearch Params:* (Multiple, 100 iterations)
  - *Best Params:* `{'subsample': 0.9, 'reg_lambda': 0.1, ...}`
  - *Public Leaderboard Score:* `1.999`

- **Random Forest:**
  - *RandomSearch Params:* (Multiple, 10 iterations)
  - *Best Params:* `{'n_estimators': 100, 'min_samples_split': 10, ...}`
  - *Public Leaderboard Score:* `2.177`

6.3 Interpretation and Summary

The linear models (Linear, Lasso, Ridge, SGD) performed best in cross-validation, indicating a predominantly linear data relationship. The strong **0.91 correlation** between `Initial Health Score` and `Recovery Index` (found in EDA) likely explains why simple linear models were so effective.

While 2nd-degree Polynomial Regression achieved a slightly better score on the public leaderboard, its performance in our cross-validation was marginally worse than the pure linear models. This, combined with its added complexity, suggests a high risk of overfitting to the public test set.

The most robust, generalizable, and interpretable model is a simple linear one. The fact that tuned Lasso and Ridge models defaulted to near-zero alphas (0.01) strongly supports this, as they are effectively performing like a standard Linear Regression model. This suggests many features are likely unnecessary, and a simple linear model or a Lasso model (which can perform feature selection) is the most appropriate.

Ensemble methods (Random Forest, GB, XGB) and Polynomial Regression, which are designed to capture more complex, non-linear patterns, did not provide a significant performance boost. This suggests the additional features and lifestyle factors have a weak, linear, or negligible impact on recovery compared to the patient's initial health.

7 Discussion of Performance

7.1 Key Observations

1. **Linear Models Performed Best in Cross-Validation:** All linear-based models (Lasso, Ridge, Linear, SGD) achieved the highest and virtually indistinguishable CV R^2 scores ($\approx$0.9879).

2. **Public Leaderboard vs. Validation:** While Polynomial Regression (Deg 2) scored highest on the public leaderboard, it did not outperform the linear models in our 5-fold cross-validation. This discrepancy suggests it may be overfitting to the specific data in the public set and may not generalize as well to the private leaderboard.

3. **Regularization Tends to Zero:** The best-tuned Lasso and Ridge models selected an alpha of 0.01. An alpha this close to zero implies minimal regularization, meaning these models are behaving almost identically to a standard Linear Regression model. This strongly indicates the data's underlying relationship is linear and many features may be unnecessary.

4. **Dominant Feature:** The `Initial Health Score` is the single most dominant predictor, as identified in the EDA heatmap (correlation of 0.91). This strong linear relationship is the primary driver of the high R^2 scores.

5. **Ensemble Models Underperformed:** Ensemble methods (RF, GB, XGB) also performed very well ($R^2 > 0.985$) but were slightly outperformed by the simpler linear models, likely due to the data's strong linearity.

7.2 Conclusion

Based on robust 5-fold cross-validation, the Lasso Regression model (with `alpha=0.01`) and the standard Linear Regression model are the best-performing and most reliable choices.While Polynomial Regression achieved a higher public leaderboard score, its slightly lower cross-validation score and increased complexity suggest a risk of overfitting.Therefore, the simple, fast, and interpretable Linear Regression model or Lasso model represents the most robust and generalizable solution.