Team Members

Name	Seat Number	Department	Level
ارساني عادل قسطندي تاوضروس	2022170050	CS	3rd
خالد سيد عبدالعزيز محمود	2022170137	CS	3rd
محمد عبد العليم عبد الحميد مكتى	2022170379	CS	3rd
علي أمجد	2022170000	CS	3rd
عمر أمجد	2022170000	CS	3rd
محمود باسم محمدصبري	2022170395	CS	3rd

1 Introduction

This report presents the first milestone of a project aimed at predicting the progression of Parkinson's disease using the Unified Parkinson's Disease Rating Scale (UPDRS) as the target variable. The dataset includes patient attributes such as demographic information, medical history, symptoms, and lifestyle factors. The focus of this phase is on data preprocessing, feature analysis, and regression modeling to establish a machine learning pipeline. The goal is to identify key predictors of UPDRS and evaluate regression models for their predictive performance.

2 Dataset and Preprocessing

2.1 Dataset Description

The dataset, stored in `parkinsons_disease_data_reg.csv`, contains various features related to Parkinson's disease patients, including:

- Demographic features: Age, Gender, EducationLevel
- Lifestyle factors: WeeklyPhysicalActivity, AlcoholConsumption
- Clinical measures: BMI, CholesterolTotal, CholesterolHDL, MoCA, FunctionalAssessment
- Medical history: Conditions like Depression, Stroke (stored as nested lists in MedicalHistory)
- Symptoms: Features like PosturalInstability (stored as nested lists in Symptoms)
- Target: UPDRS (a continuous score indicating disease severity)

The dataset was split into training and test sets with an 80:20 ratio.

- Training set: 1620 rows
- Test set: 406 rows

No separate validation set was used in this phase, as the focus was on initial model evaluation.

2.2 Preprocessing Techniques

Several preprocessing steps were applied to prepare the data for modeling:

2.2.1 Handling Missing Values

Missing values in the `EducationLevel` column were filled with `No Education` to ensure completeness:

```
data.fillna({'EducationLevel': 'No Education'}, inplace=True)
```

2.2.2 Feature Extraction

The `MedicalHistory` and `Symptoms` columns contained nested data structures (lists), which were expanded into separate binary features (e.g., `MedHist_Depression`, `Symptom_PosturalInstability`):

```
data[’MedicalHistory’] = data[’MedicalHistory’].apply(ast.literal_eval)
data[’Symptoms’] = data[’Symptoms’].apply(ast.literal_eval)
medical_history_data = data[’MedicalHistory’].apply(pd.Series)
symptoms_data = data[’Symptoms’].apply(pd.Series)
data = pd.concat([data.drop(columns=[’MedicalHistory’, ’Symptoms’]),
                  medical_history_data.add_prefix(’MedHist_’),
                  symptoms_data.add_prefix(’Symptom_’)], axis=1)
```

This transformation converted each medical condition and symptom into a separate column, with 1 indicating presence and 0 indicating absence.

2.2.3 Data Transformation

The `WeeklyPhysicalActivity (hr)` feature, originally in hours:minutes format, was converted to total minutes for consistency:

```
def hour_to_minutes(time):
    split = str(time).split(’:’)
    hour = int(split[0])
    minute = int(split[1])
    return hour * 60 + minute
data["WeeklyPhysicalActivity (hr)"] = data["WeeklyPhysicalActivity (hr)"].apply(hour_to_minu
```

2.2.4 Normalization

Numerical features were normalized to a [0, 1] range using `MinMaxScaler` to ensure all features contribute equally to the model:

```
scaler = preprocessing.MinMaxScaler()
data[Numerical_cols] = scaler.fit_transform(data[Numerical_cols])
```

2.2.5 Categorical Encoding

Categorical features (e.g., `EducationLevel`, `Gender`) were encoded into numerical values using `LabelEncoder`:

```
for col in categorical_cols:
    le = preprocessing.LabelEncoder()
    data[col] = le.fit_transform(data[col].astype(str))
```

2.2.6 Outlier Detection Using IQR

To ensure data quality, outliers in numerical features were identified using the Interquartile Range (IQR) method. The IQR method calculates the range between the first quartile (Q1) and the third quartile (Q3), and identifies outliers as values falling below $Q1 - 1.5 \times IQR$ or above $Q3 + 1.5 \times IQR$. The following function was implemented to detect outliers:

```
def detect_outliers_iqr(data):
    Q1 = data.quantile(0.25)
    Q3 = data.quantile(0.75)
    IQR = Q3 - Q1
    outliers = ((data < (Q1 - 1.5 * IQR)) | (data > (Q3 + 1.5 * IQR)))
    return outliers.sum()
```

This function was applied to numerical columns to quantify outliers, which helped in understanding the data distribution and informed subsequent preprocessing decisions.

3 Feature Analysis and Selection

3.1 Feature Correlation Analysis

Pearson correlation was used to identify numerical features with a significant linear relationship with `UPDRS` (threshold $|r| > 0.03$). The correlation matrix for a subset of selected features is shown below:

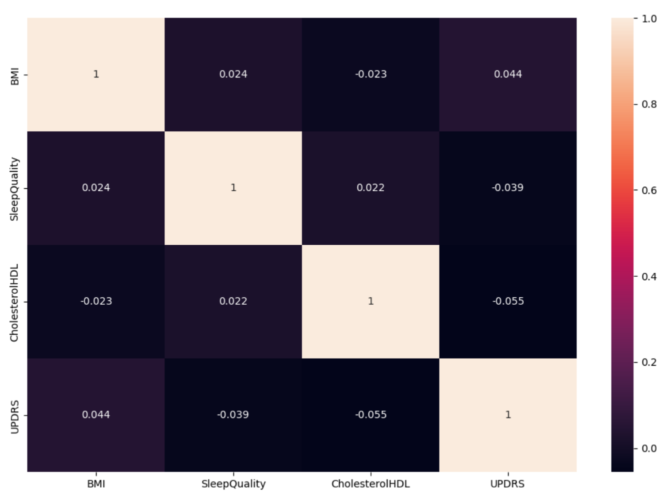

Figure 1: Correlation heatmap of selected numerical features.

3.2 Feature Selection Methods

3.2.1 Numerical Features (Pearson Correlation)

Features with $|r| > 0.03$ with `UPDRS` were selected:

- BMI
- CholesterolTotal
- SleepQuality

3.2.2 Categorical Features (ANOVA/T-Test)

ANOVA and t-tests were used to select significant categorical features (p-value ¡ 0.05), with a sample size check to avoid issues with small groups:

- ANOVA for features with 3+ categories (e.g., `EducationLevel`).
- T-test for binary features (e.g., `Gender`).

The following categorical feature was selected:

- Symptom_PosturalInstability

3.2.3 Random Forest Feature Importance

A Random Forest model was used to rank features by importance. The top 7 features were:

- BMI
- SleepQuality
- CholesterolHDL
- FunctionalAssessment
- CholesterolLDL
- AlcoholConsumption
- WeeklyPhysicalActivity (hr)

3.2.4 Recursive Feature Elimination (RFE)

RFE with a linear regression model was used to select 7 features:

- BMI
- CholesterolHDL
- MedHist_Depression

- MedHist_Stroke
- MoCA
- SleepQuality
- Symptom_PosturalInstability

3.3 Final Feature Set

The final feature set was the union of features selected by Pearson correlation, ANOVA/t-test, Random Forest, and RFE:

- AlcoholConsumption
- BMI
- CholesterolHDL
- CholesterolLDL
- FunctionalAssessment
- MedHist_Depression
- MedHist_Stroke
- MoCA
- SleepQuality
- Symptom_PosturalInstability
- WeeklyPhysicalActivity (hr)

4 Regression Models

4.1 Models Used

Three regression models were implemented with polynomial features (degree=2):

- **Polynomial Regression**: Standard linear regression with polynomial features.
- **Ridge Regression**: Linear regression with L2 regularization to reduce overfitting.
- **Lasso Regression**: Linear regression with L1 regularization for implicit feature selection.

4.2 Model Performance

The models were evaluated using Mean Squared Error (MSE) on the training and test sets:

Model	Training MSE	Testing MSE
Polynomial Regression	0.0779	0.0791
Ridge Regression	0.078	0.0784
Lasso Regression	0.0818	0.0764

Table 1: Performance of regression models with polynomial features (degree=2).

4.3 Actual vs. Predicted Plots

Scatter plots of actual vs. predicted `UPDRS` values were generated for each model:

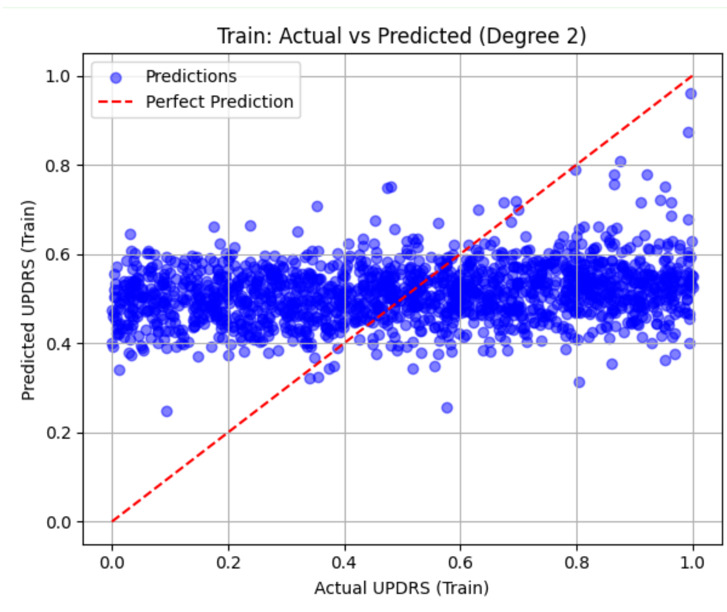

Figure 2: Polynomial Regression: Actual vs. predicted UPDRS values (Training set).

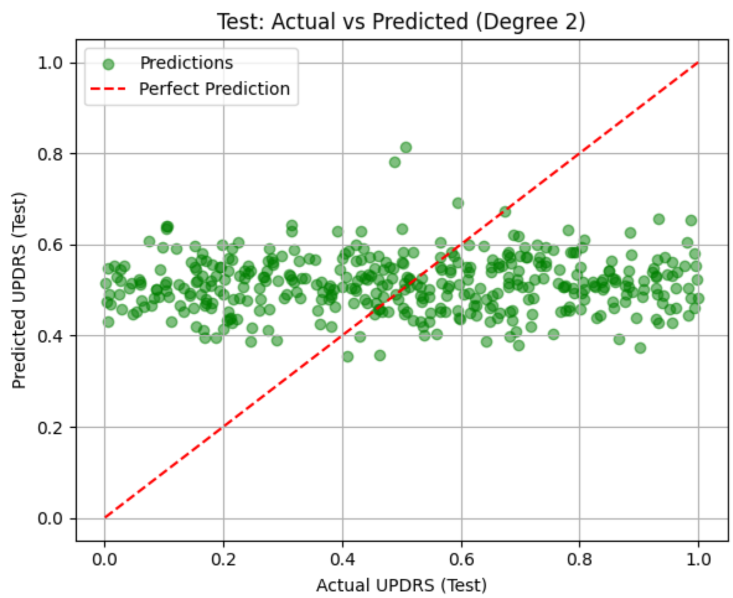

Figure 3: Polynomial Regression: Actual vs. predicted UPDRS values (Test set).

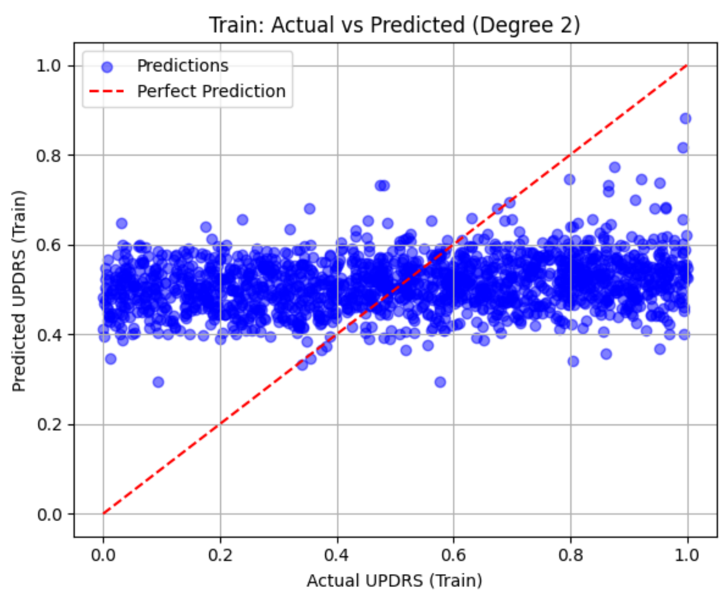

Figure 4: Ridge Regression: Actual vs. predicted UPDRS values (Training set).

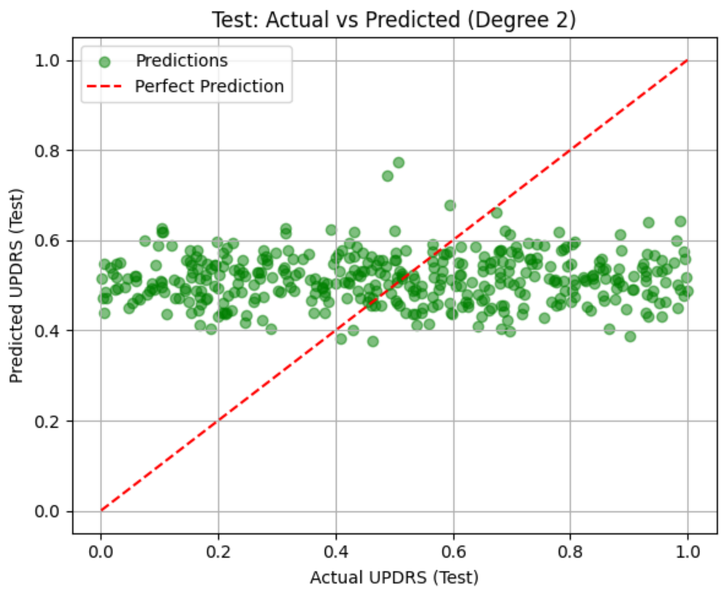

Figure 5: Ridge Regression: Actual vs. predicted UPDRS values (Test set).

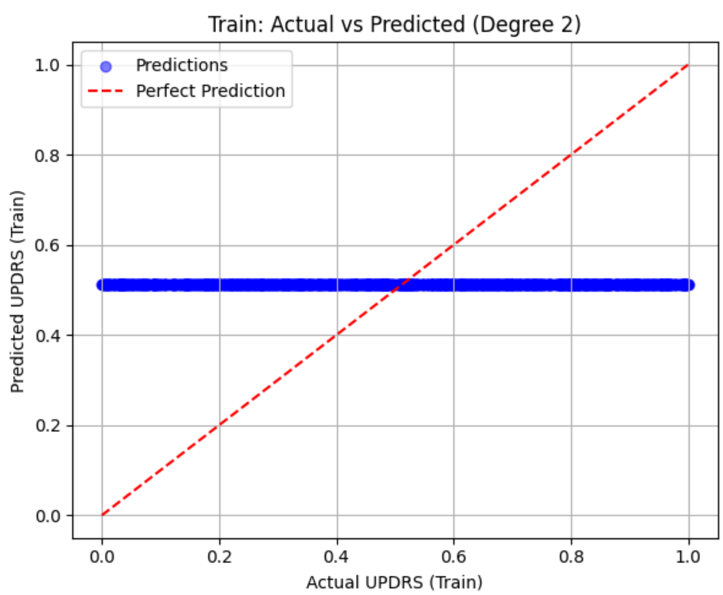

Figure 6: Lasso Regression: Actual vs. predicted UPDRS values (Training set).

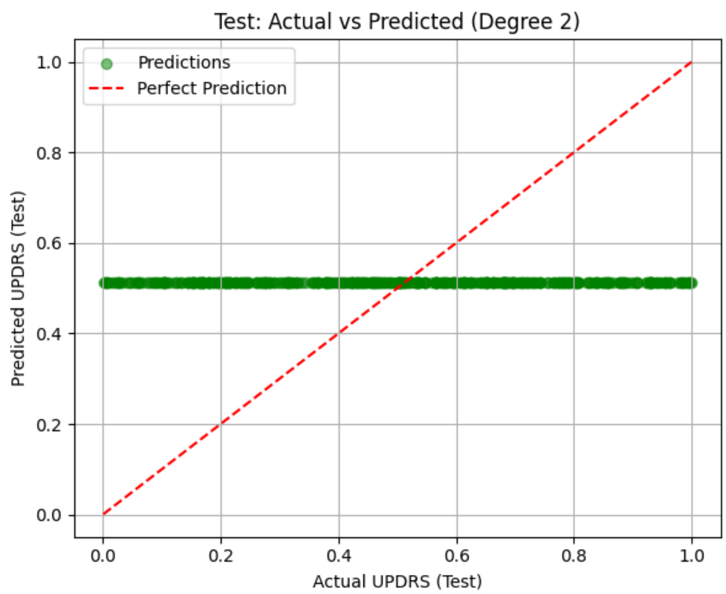

Figure 7: Lasso Regression: Actual vs. predicted UPDRS values (Test set).

5 Discussion

5.1 Model Comparison

- **Polynomial Regression**: Showed balanced performance (Training MSE:0.0779, Testing MSE: 0.0791), indicating good generalization but limited ability to capture complex patterns.
- **Ridge Regression**: Slightly better testing performance (0.0783) than Polynomial Regression, benefiting from L2 regularization to mitigate overfitting.
- **Lasso Regression**: Best testing performance (0.0764) despite a higher training error (0.0818), leveraging L1 regularization for implicit feature selection.

All models showed similar performance, with differences in MSE being small. The slightly better test MSE of Lasso suggests that feature selection via L1 regularization was beneficial.

6 Conclusion

This phase of the project focused on building a foundation for predicting Parkinson's disease progression through data preprocessing, feature analysis, and regression modeling. Our intuition was that clinical and lifestyle factors, such as

`FunctionalAssessment` and `WeeklyPhysicalActivity`, would be strong predictors of UPDRS scores. This intuition was proven, as the selected features demonstrated predictive power, and the regression models achieved test MSE values ranging from 0.0764 to 0.0791, indicating a successful initial pipeline.