Laboratory 4

Variant 2

Group 12

By Andrea Amato and Dimitrios Georgouisis

Introduction

The goal of this project was to predict disease progression in patients based on the Diabetes dataset. This dataset comprises several diagnostic measurements, which consist of ten baseline variables, such as age, sex, body mass index (BMI), average blood pressure, and six blood serum measurements taken from patients diagnosed with diabetes. The target variable is a quantitative measure of disease progression one year after the baseline.

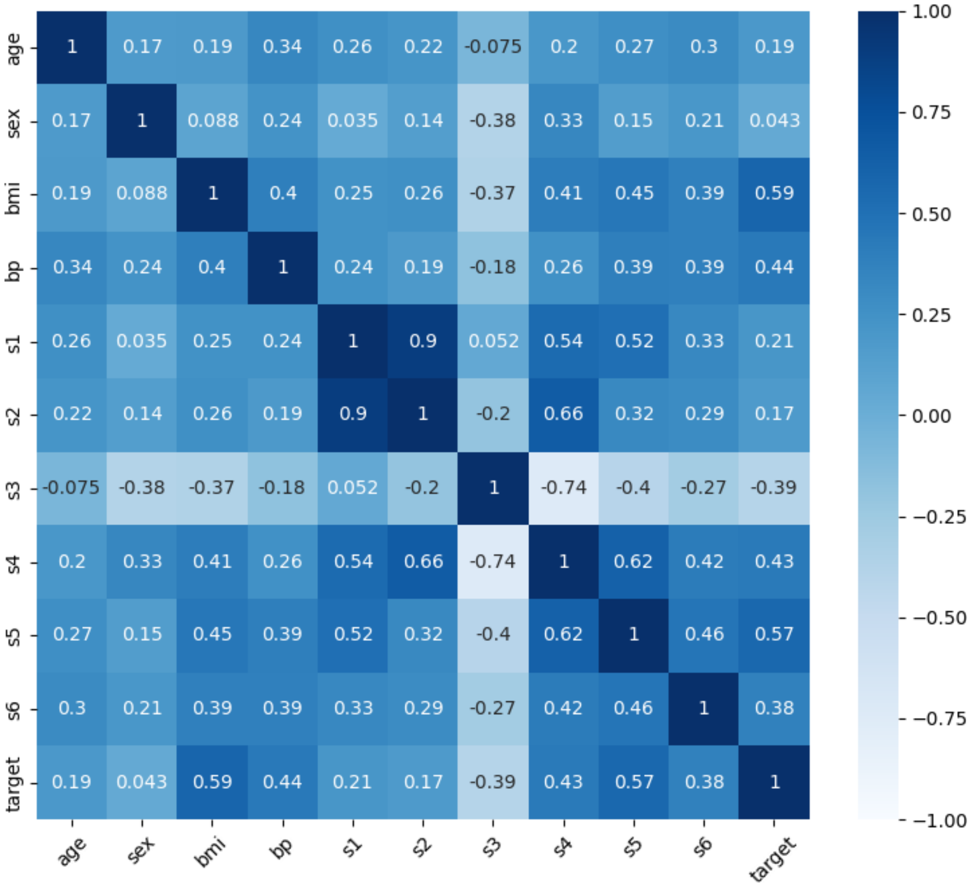
Data Preparation

Upon initially reviewing the dataset, it was noted that each of the ten feature variables had been mean centered and scaled by the standard deviation times the square root of the number of samples. This unique scaling means that while each feature's variance was normalized to ensure the sum of squares totals 1 across each column, it did not conform to the standard normalization where features are typically scaled to have a standard deviation of 1. To address this and convert the scaling to standard normalization, the data was multiplied by the square root of the number of samples to undo the initial scaling. This adjustment ensures that features contribute equally to the model training process, maintaining consistency and reliability in predictive performance.

Significant correlations were observed among certain features *(Fig.1)*:

- Strong positive correlations: BMI with target, and several serum measurements (s1 with s2, s4, and s5, etc.).
- Strong negative correlation: Between s3 and s4.

To reduce multicollinearity, which could affect the model predictions, features s2 and s4 were removed. This step simplifies the models and helps in improving the interpretability of the results.



	age	sex	bmi	bp	s1	s2	s3	s4	s5	s6	target
age	1	0.17	0.19	0.34	0.26	0.22	-0.075	0.2	0.27	0.3	0.19
sex	0.17	1	0.088	0.24	0.035	0.14	-0.38	0.33	0.15	0.21	0.043
bmi	0.19	0.088	1	0.4	0.25	0.26	-0.37	0.41	0.45	0.39	0.59
bp	0.34	0.24	0.4	1	0.24	0.19	-0.18	0.26	0.39	0.39	0.44
s1	0.26	0.035	0.25	0.24	1	0.9	0.052	0.54	0.52	0.33	0.21
s2	0.22	0.14	0.26	0.19	0.9	1	-0.2	0.66	0.32	0.29	0.17
s3	-0.075	-0.38	-0.37	-0.18	0.052	-0.2	1	-0.74	-0.4	-0.27	-0.39
s4	0.2	0.33	0.41	0.26	0.54	0.66	-0.74	1	0.62	0.42	0.43
s5	0.27	0.15	0.45	0.39	0.52	0.32	-0.4	0.62	1	0.46	0.57
s6	0.3	0.21	0.39	0.39	0.33	0.29	-0.27	0.42	0.46	1	0.38
target	0.19	0.043	0.59	0.44	0.21	0.17	-0.39	0.43	0.57	0.38	1

Figure 1 – Correlation matrix of the dataset features.

Data Split

For this project, the dataset was split into training and test partitions with 80% of the data allocated for training and 20% reserved for testing. This division allows the models to learn from a substantial portion of the data (training set) and then be evaluated on a separate set of data that the models have not seen during training (test set).

This approach helps in assessing the generalizability of the models — how well they can perform on new, unseen data, which is indicative of their practical applicability.

Model Selection

Two models were chosen for this regression task:
1. **Linear Regression**: A fundamental model for regression tasks due to its simplicity and interpretability.
2. **Random Forest Regressor**: Selected for its robustness to outliers and ability to model non-linear relationships without extensive hyperparameter tuning.

These models were chosen to provide a comparison between a simple linear approach and a more complex ensemble method.

Model Training and Evaluation

The models were trained using a 4-fold cross-validation approach to ensure that the evaluation is robust and not biased towards any particular split of the data. The primary metric used for evaluation was the Root Mean Squared Error (RMSE), which provides a clear indication of the average error magnitude in the predictions.

The cross-validation results were as follows:

- **Linear Regression**: An RMSE of 53.566.
- **Random Forest Regressor**: An RMSE of 57.009.

Surprisingly, the simpler Linear Regression model outperformed the more complex Random Forest Regressor in terms of RMSE. This could suggest that the relationships in the dataset are more linear, and additional complexity introduced by the Random Forest does not capture the underlying patterns any better.

To assess the possibility of overfitting, both models were subsequently trained on the entire training set and evaluated on both the training and test sets. The RMSE for these evaluations yielded:

- **Linear Regression**: A training RMSE of 52.407 and a test RMSE of 58.874.
- **Random Forest Regressor**: A training RMSE of 21.836 and a test RMSE of 62.161.

The results highlighted a significant discrepancy in the Random Forest Regressor's performance on the training versus the test set, indicating overfitting. The model's complexity allowed it to perform exceptionally well on the data it was trained on, but it failed to generalize to the test data. On the other hand, the Linear Regression model showed a smaller gap between training and test RMSE, pointing to a better balance between bias and variance.

Hyperparameter Tuning

In the process of parameter tuning, it was essential to choose models that allowed for the exploration and manipulation of hyperparameters to optimize performance. While Linear Regression is a fundamental model for regression tasks, it lacks hyperparameters that could be varied for tuning purposes. Consequently, Ridge Regression was used in place of Linear Regression. Ridge Regression is an extension of Linear Regression where a regularization term (L2 penalty) is added to the cost function *(see Eq.1)*. This regularization term is controlled by the hyperparameter α (alpha). When $\alpha = 0$, Ridge Regression reduces to Linear Regression.

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)^2 + \frac{\alpha}{2}\sum_{j=1}^{n}\theta_j^2$$

Equation 1 – Ridge Regression cost function. When $\alpha = 0$, it corresponds to Linear Regression cost function.

The first term in this cost function is the mean squared error (MSE) between the predicted values and the actual values, which is typical for regression models. The second term is the regularization term that penalizes the magnitude of the coefficients, effectively controlling their size to prevent overfitting. This helps in making the model less sensitive to the training data and improves its generalizability.

To optimize model performance and select the best hyperparameters, we employed the `GridSearchCV` method along with 4-fold cross-validation. This allowed us to systematically explore a range of hyperparameter values and

assess model performance using Root Mean Squared Error (RMSE) as the evaluation metric.
The evaluation involved:

- **Ridge Regression**: Tuning the alpha parameter, exploring values from 0.001 to 500.
- **Random Forest Regressor**: Adjusting parameters such as as `n_estimators`, `max_depth`, `min_samples_split`, `min_samples_leaf`, and `max_features` to optimize the model.

The results of these evaluations were used to identify the most effective model configurations.

Results

Upon fine-tuning our models, we identified the best hyperparameters for both Ridge Regression and the Random Forest Regressor. The best RMSE score for Ridge Regression was observed at `alpha = 10` *(Tab.1)*, showing a slight improvement over the base Linear Regression model. For the Random Forest Regressor, the best configuration used 100 estimators, no limit on `max_depth`, and a higher `min_samples_split` and `min_samples_leaf`, indicating a preference for a more complex model to capture intricate patterns in the data *(Tab.2)*.

Alpha	RMSE
0.001	53.566
0.01	53.566
0.1	53.565
1	53.558
10	53.513
50	53.735
100	54.404
500	60.105

Table 1 – Ridge Regression hyperparameter tuning results.

Max Depth	Max Features	Min Samples Leaf	Min Samples Split	Estimators	RMSE
10	sqrt	3	6	50	56.177
10	sqrt	3	6	100	55.622
10	sqrt	3	10	50	56.073
10	sqrt	3	10	100	55.556
10	sqrt	4	6	50	55.944
10	sqrt	4	6	100	55.673
10	sqrt	4	10	50	56.072
10	sqrt	4	10	100	55.491
None	sqrt	3	6	50	56.256
None	sqrt	3	6	100	55.493
None	sqrt	3	10	50	56.017
None	sqrt	3	10	100	55.432
None	sqrt	4	6	50	55.980
None	sqrt	4	6	100	55.656
None	sqrt	4	10	50	55.929
None	sqrt	4	10	100	55.380

Table 2 – Random Forest Regressor hyperparameter tuning results.

The above tables reveal the impact of different parameter values on the model's ability to predict disease progression accurately. The Random Forest Regressor showed variability in performance based on the depth and complexity of the trees, with no single configuration dominating across all metrics. This suggests that while hyperparameter tuning can yield incremental improvements, there is a limit to the benefits that can be extracted purely from tuning parameters, and further feature engineering or model selection strategies might be necessary to achieve significant performance gains.

To rigorously evaluate their performance and assess their generalization capability, we trained these optimal models on the entire training dataset. We then conducted a final evaluation on both the training and test sets to check for possible overfitting.
The final RMSE evaluation on the Training and Test sets with Best Models is as follows:

- Ridge Regression Best Model:

- Train RMSE: 52.423
- Test RMSE: 58.596

- Random Forest Best Model:

- Train RMSE: 40.567
- Test RMSE: 59.362

The Ridge Regression model demonstrated a modest gap between the training and test RMSE, indicating that the model had not severely overfitted the training data. Notably, the optimal Ridge Regression did not significantly outperform the initial Linear Regression model on the test set. This suggests that the regularization introduced by Ridge Regression's L2 penalty had a limited effect on enhancing the model's generalization to unseen data in this particular case.

For the Random Forest Regressor, the training RMSE was substantially lower than the test RMSE. Such a large discrepancy indicates that the Random Forest model, despite being the best model from hyperparameter tuning, did overfit the training data. Nonetheless, post-hyperparameter tuning, the gap between training and test RMSE significantly narrowed, indicating reduced overfitting. This improvement can be attributed to the hyperparameter tuning process which optimized the model to balance fitting the training data against the complexity of the model. By imposing more constraints, it was unable to fit the training data as closely as before, hence the higher RMSE on the training set.

Conclusion

This study on predicting diabetes progression provided valuable insights into effective machine learning practices. Through rigorous data preparation, model selection, and evaluation, we learned that simpler models like Linear Regression can perform comparably to more complex ones like Random Forest, especially when the latter is prone to overfitting despite hyperparameter tuning.

This exercise underscored the importance of understanding the characteristics of the dataset and highlighted the need for careful balance between model complexity and its ability to generalize. Our findings emphasize that while hyperparameter tuning is crucial for optimizing model performance, it must be complemented with a deep understanding of the underlying data to achieve meaningful improvements.