Machine Learning Course NOVA FCT

Survival Time in Multiple Myeloma Patients

Group: Big Data RNV

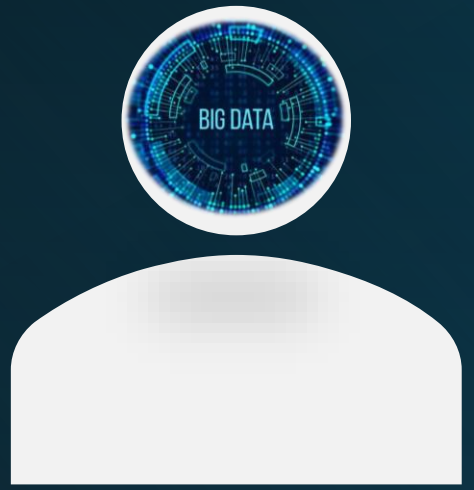
2024/2025

- **Name 1: Nuno Nogueira**
- **Number 1: 70169**

- **Name 2: Rodrigo Pedro**
- **Number 2: 70058**

- **Name 3: Victtor Moraes**
- **Number 3: 69964**

- **Final score: 3.56622**
- **Leaderboard ranking: 49th**



Team Identification

Final Solution

- 1.1
  - 5 cells hidden
- 1.2 Baseline model
  - 4 cells hidden
- 1.3 Baseline with gradient descent
  - 6 cells hidden

2.1 & 2.2

- Polynomial regression
  - 1 cell hidden
- KNN
  - 1 cell hidden

3.1

- Imputation of missing data with SimpleImputer
  - 2 cells hidden
- Imputation of missing data with IterativeImputer
  - 4 cells hidden
- imputation of missing data with knn imputer (better)
  - 4 cells hidden
- KNN model performance with imputed data from KNNImputer
  - 1 cell hidden
- Polynomial regression with imputed data from KNNImputer
  - 1 cell hidden

3.2

- HistGradientBoostingRegressor
  - 1 cell hidden
- CatBoostRegressor
  - 1 cell hidden
- CatBoostRegressor with SurvivalAft
  - 1 cell hidden
- Greedy search to find the best parameters for CatBoost with Survival Aft
  - 2 cells hidden
- Predictions
  - 2 cells hidden

3.3

- Gradient Boosting on imputed data from KNNImputer
  - 2 cells hidden

4.1

- KNNImputer trained on the data with and without labels
  - 1 cell hidden
- Linear regression with imputed data from KNNImputer trained on the data with and without labels
  - 1 cell hidden
- Isomap
  - 4 cells hidden

Outline

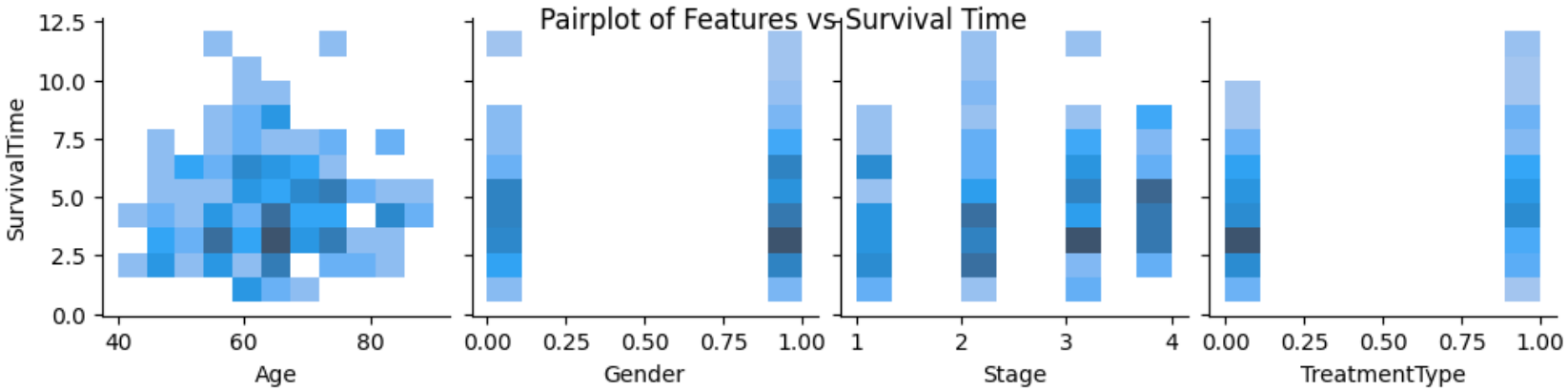
Task [1] - **Data Preparation**

- We began by analyzing the distribution of missing values in the dataset. Among the 400 data points in the training set, three features were found to have missing data: the Comorbidity Index (17.50% missing), Genetic Risk (20.00% missing), and Treatment Response (5.25% missing).

- Our inspection also revealed that 90 entries (22.50%) were unlabeled. Notably, all of these unlabeled rows were marked with a "Censored" value of 0, indicating that they were not censored.

- The differences in the proportions and characteristics of missing values across these features played a key role in shaping our approach to data imputation, as detailed in the subsequent discussion.

Task [1] - **Data analysis**



- The pairplot illustrates the relationship between survival time and key patient features, including age, gender, disease stage, and treatment type. Regarding the relationship with age, there is a broad dispersion in survival times across all age groups, with no clear linear trend.

- Younger patients (ages 40–60) appear to have slightly longer survival times compared to older groups, though this pattern is not strongly defined. This suggests that age alone may not be a strong predictor of survival time and should be analyzed in combination with other variables.

Task [1] - **Data analysis**

- When examining gender, the data indicates no significant differences in survival times between male and female patients. Conversely, the disease stage shows a clear relationship, with earlier stages (1 and 2) being associated with longer survival times compared to advanced stages (3 and 4).

- Lastly, the relationship between treatment type and survival time reveals limited variability across different treatments. This could indicate that treatment type, as presented in the data, may have less individual impact on survival outcomes. However, its effect might be more pronounced when considered alongside other factors, such as disease stage.

Task [1.2] – **Baseline Model**

- We developed a baseline model using linear regression to perform the initial prediction. The model pipeline included a normalization process (StandardScaler) to scale the data before applying the regressor, ensuring that all variables were on the same scale.

- **Objectives of the Baseline Model:**

  - Predict survival time for complete and uncensored data.

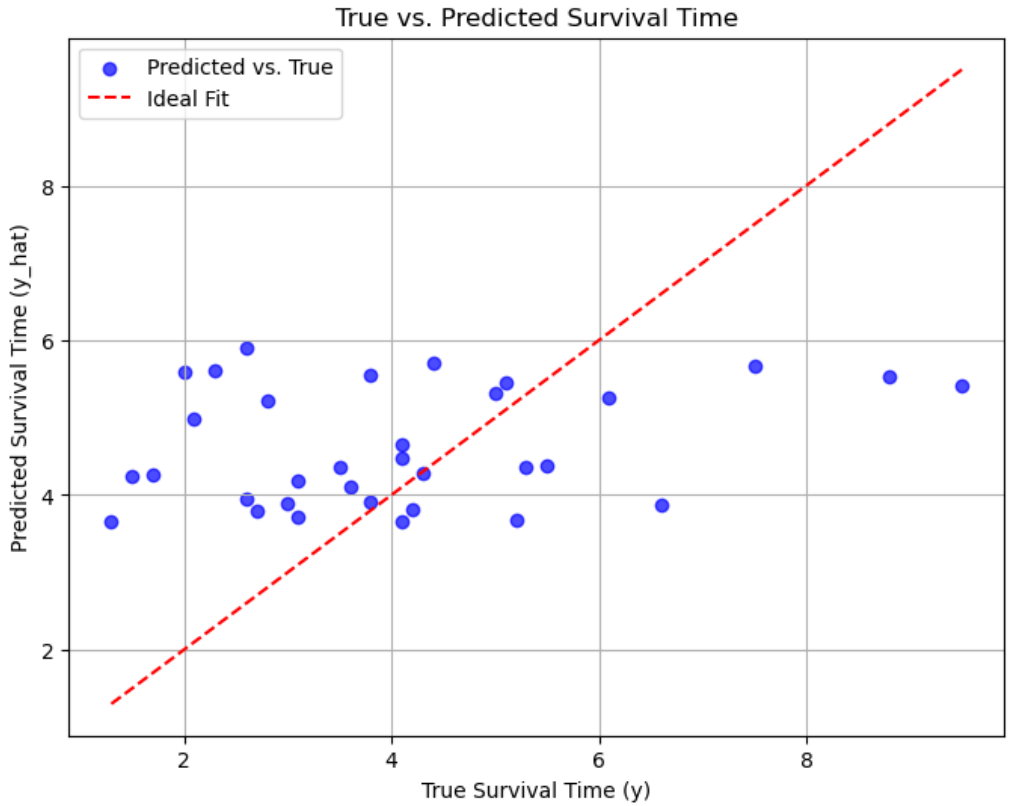
  - Establish a performance benchmark (cMSE) for comparison with more complex models.

  - Implement cross-validation to evaluate the robustness of the model.

Task [1.2] – **Baseline Model**

- Most of the predictions align along the ideal fit line, suggesting that the model captures general trends.

- Some deviations, especially for longer survival times, indicate the model's limitations in capturing more complex patterns. The simplicity of linear regression may explain these discrepancies.



- Cross-Validation cMSE (mean): 4.4908435769424075
- Cross-Validation cMSE (std): 1.9970152130446666
- Test Set cMSE: 3.757100793186035

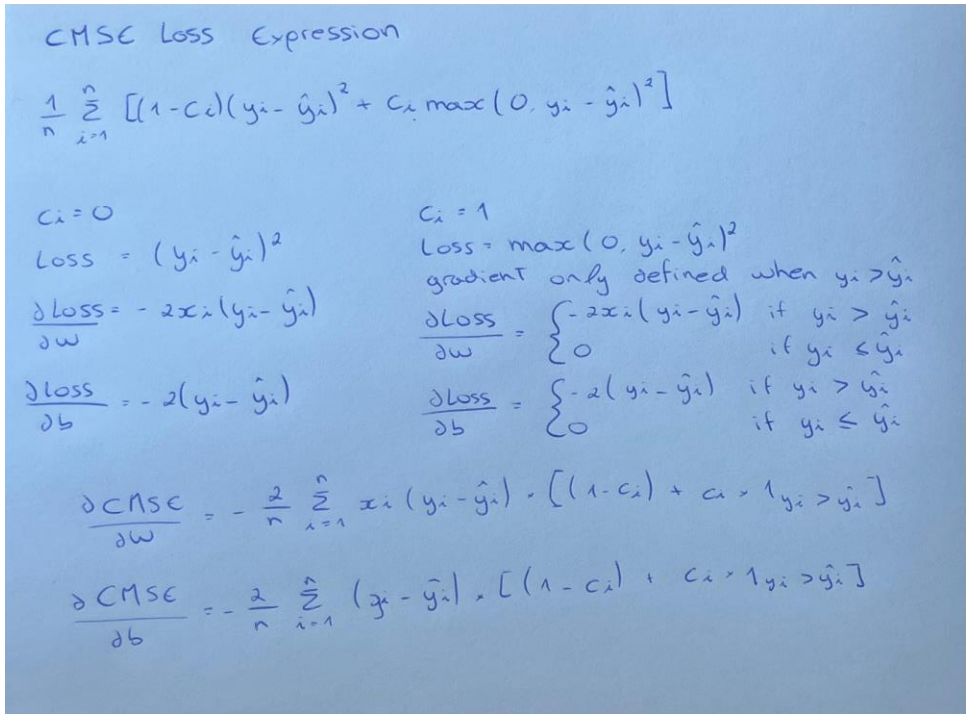
Task [1.3] – **Gradient Descent Implementation**

Gradient Descent Function:

- Scales the data using StandardScaler for better convergence.
- Updates weights and bias iteratively using the derived gradients.
- Incorporates regularization (Lasso or Ridge) to prevent overfitting.

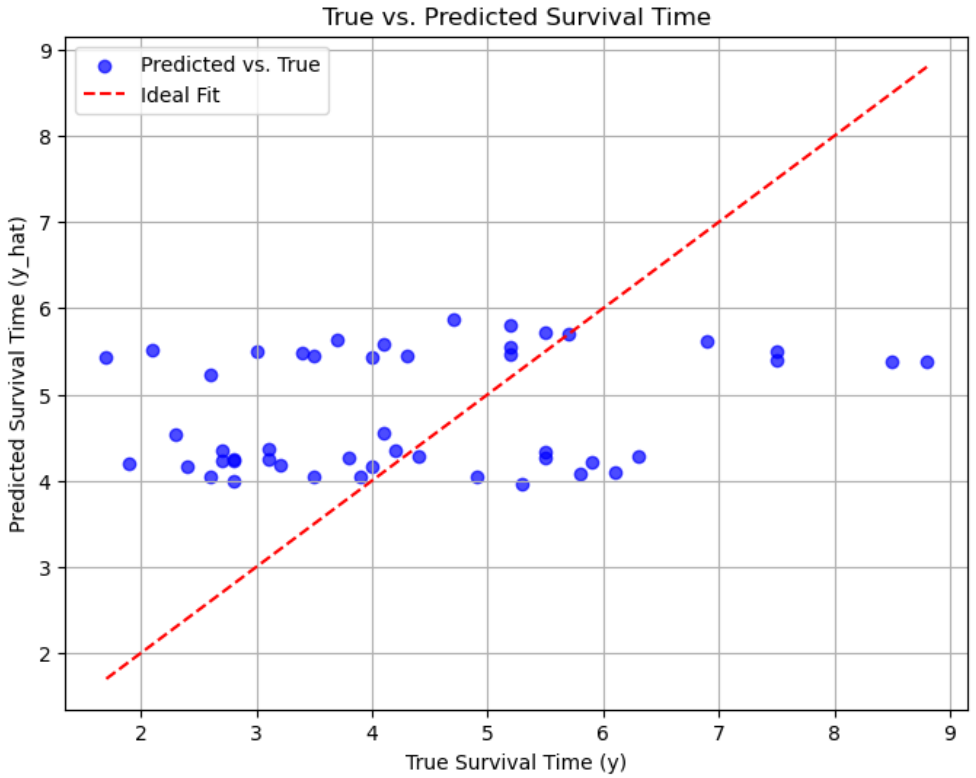
Experimentation with Regularization:

- Ridge (L2) was used in this experiment, adding stability to the weights.
- Regularization parameter (λ=0.01) controlled the trade-off between bias and variance.



Task [1.3] – **Baseline with Gradient Descent**

- The baseline model only utilized uncensored data, leading to potential information loss.
- Gradient Descent incorporated censored data effectively, yielding lower cMSE on the test set.
- The baseline model struggled with capturing complexities in the data, while the gradient descent model adapted better by leveraging the additional censored data.



- Training cMSE: 3.5520881015241668
- Test Set cMSE: 1.8460717908309618

Task [2] – **Nonlinear models**

- Develop and evaluate nonlinear models for predicting survival time using Polynomial Regression and k-Nearest Neighbors (k-NN). These methods allow us to capture more complex relationships in the data compared to linear models.

Polynomial Regression:

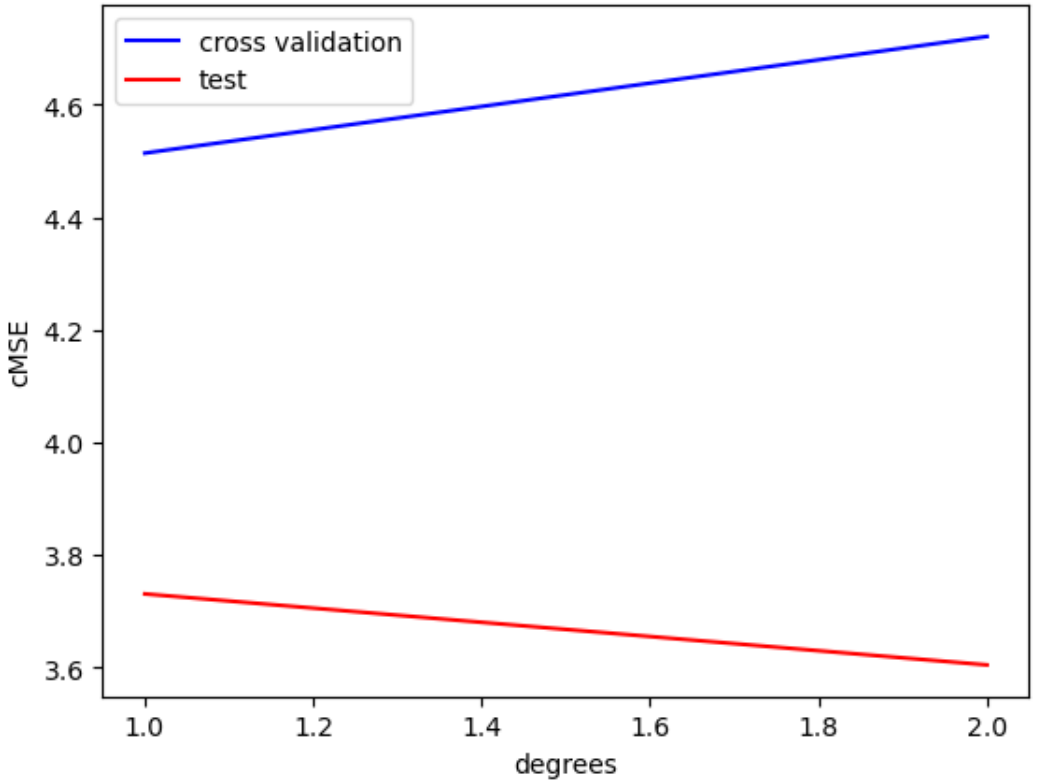
- Expanded the feature set using polynomial degrees.
- Implemented RidgeCV for regularization to avoid overfitting.
- Used cross-validation to evaluate performance for each degree.

K-Nearest Neighbors (k-NN):

- Trained models with different values of kkk to find the optimal neighborhood size.
- Employed cross-validation to evaluate the model and select the best kkk.

Task [2.1] – **Polynomial Regression**

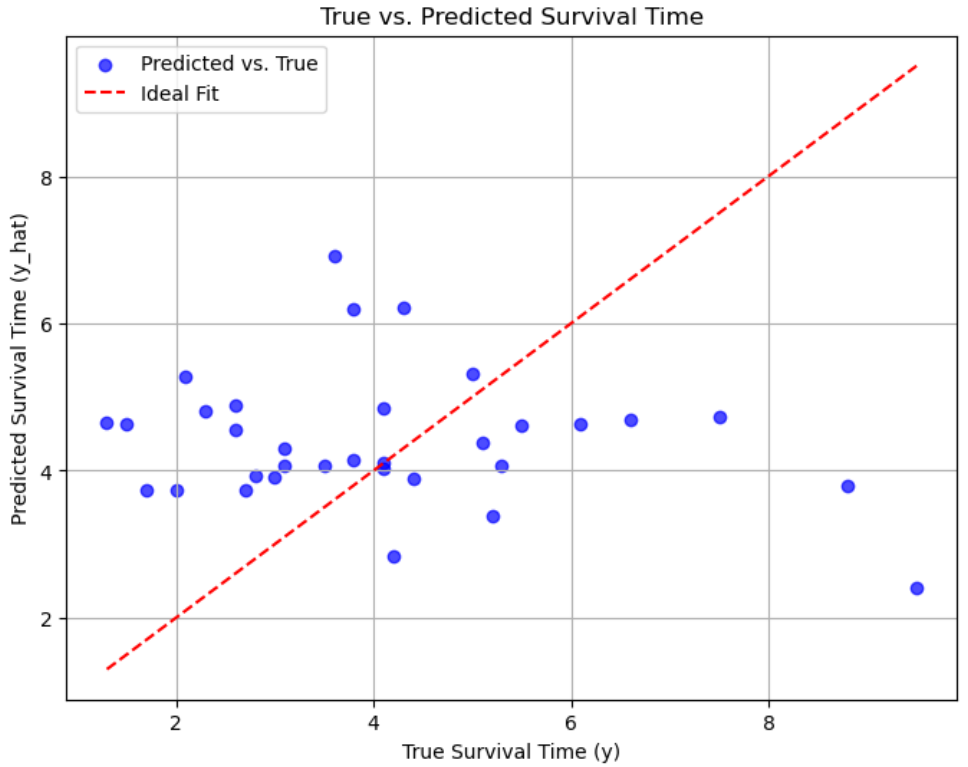
- The cross-validation error (blue line) increases with higher polynomial degrees, indicating overfitting to the training data.
- The test error (red line) decreases slightly but stabilizes for lower degrees, suggesting a trade-off between complexity and generalization.
- Best Degree: The model performs best at lower degrees (d=2), balancing bias and variance.



Fitting degree 1 with 5 features...
C-Val cMSE: 4.5142
C-Val cMSE (std):0.7026
Test cMSE: 3.7303
Fitting degree 2 with 15 features...
C-Val cMSE: 4.7216
C-Val cMSE (std):0.5933
Test cMSE: 3.6041

Task [2.2] – **Knn**

- Predictions are spread closer to the ideal fit line, but some deviations exist, especially for extreme values.
- The k=5k = 5k=5 neighborhood size achieves the best balance between stability and responsiveness to the data.
- Nonlinear models outperform the baseline linear model by capturing complex relationships in the data.
- Cross-validation is essential for hyperparameter tuning to ensure optimal performance.
- Regularization (RidgeCV) and scaling improve stability and generalization.



C-Val cMSE: 5.6763
C-Val cMSE (std): 2.0538
Test Set cMSE: 5.390(78)

Task [3] – **Handling missing data**

- We Handle the missing values in the dataset using different imputation methods:
  - **Simple Imputer**
  - **Iterative Imputter**
  - **Knn Imputer (better)**
- Missing data was handled separately for training and test sets to avoid data leakage.
- Dropped rows with missing target values (SurvivalTime) to ensure data consistency.

Task [3.1] – **Handling missing data with Knn Imputer**

- Handle missing data in the dataset using the **KNNImputer**, an imputation technique that predicts missing values based on the nearest neighbors' values. This complements previous methods (SimpleImputer and IterativeImputer).

- **Imputation Method:**
KNNImputer with k=3 neighbors.

- **Data Handling:**
Applied only to rows with missing features, after dropping rows with missing SurvivalTime

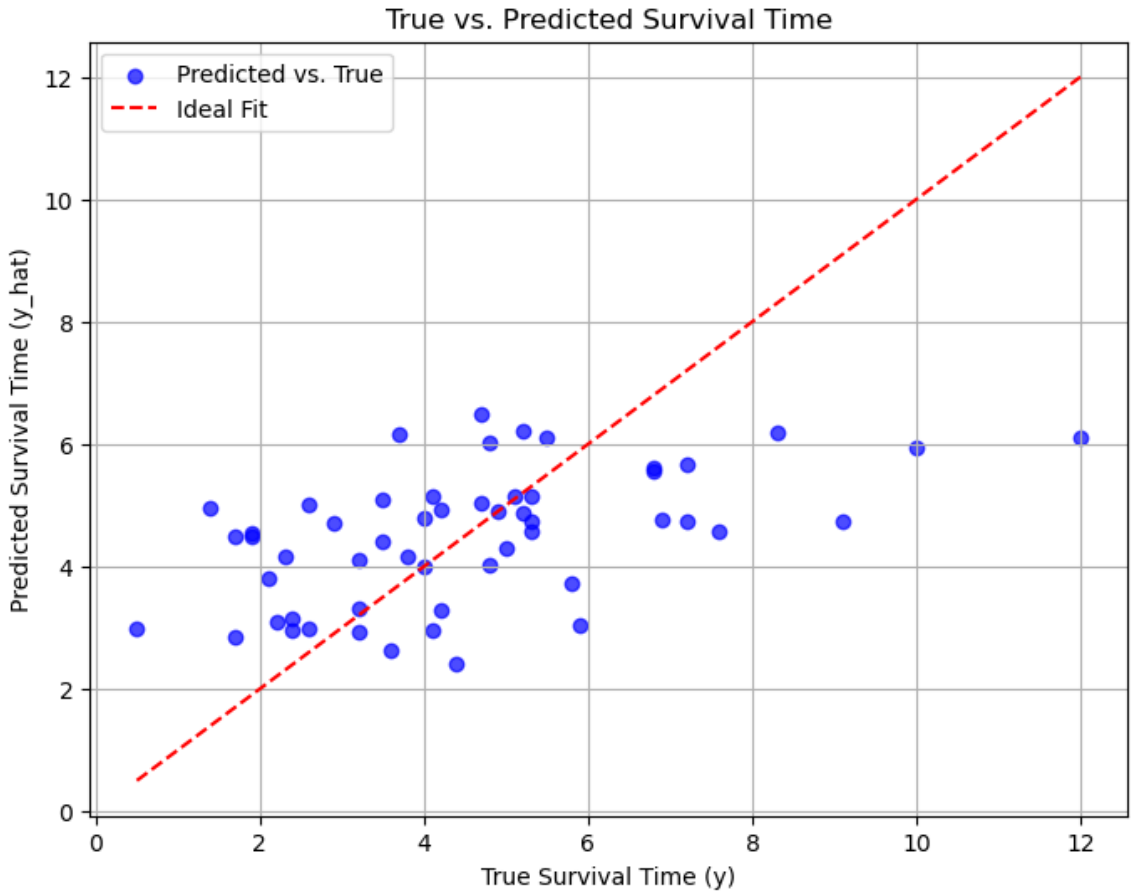
Advantages:

- Captures local patterns in the data.
- Effective for datasets with non-linear relationships.

Limitations:

- Sensitive to outliers and data scaling.
- Computationally expensive for large datasets.

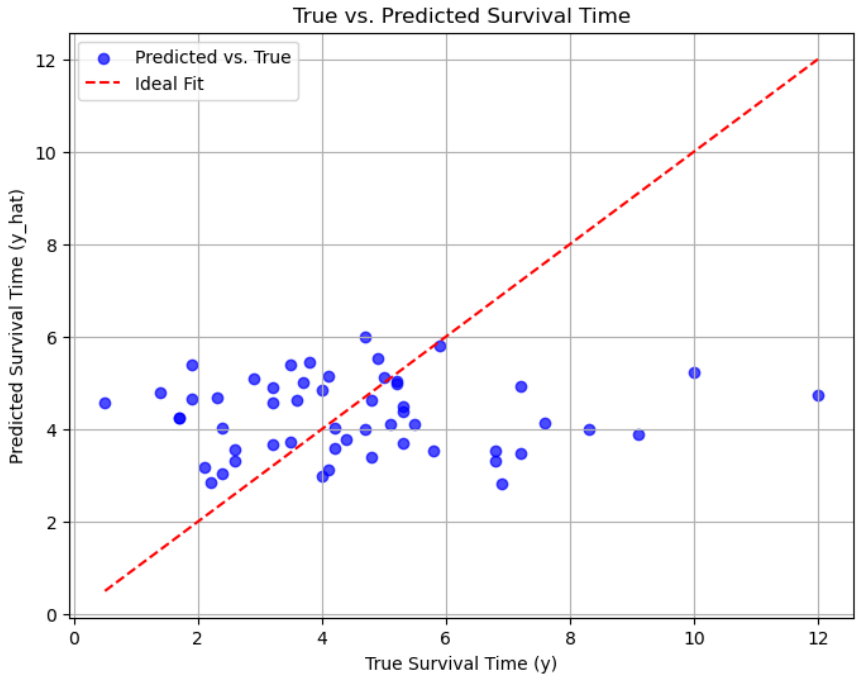
Task [3.1] – **Handling missing data with Knn Imputer**



Cross-Validation cMSE (mean): 2.2704
Cross-Validation cMSE (std): 0.4495
Test Set cMSE: 3.3241

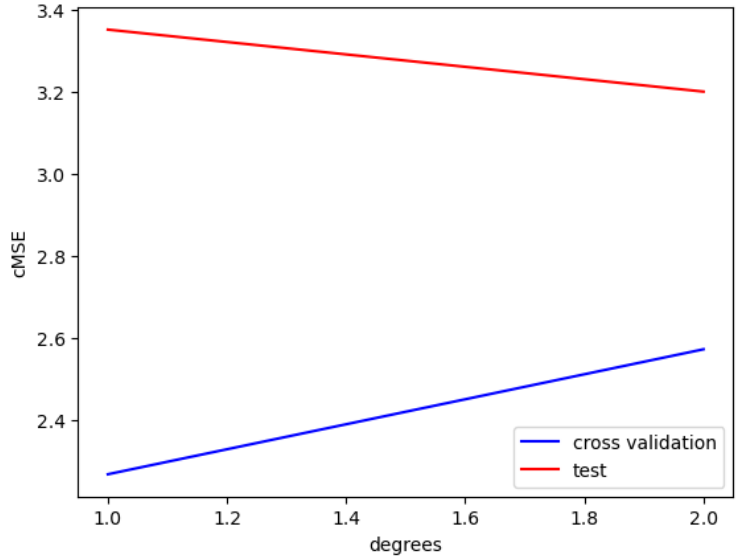
Task [3.1] – Knn and Polynomial with Knn Imputer

Knn



- C-Val cMSE: 3.3401
- C-Val cMSE (std): 0.5472
- Test Set cMSE: 5.403(1)

Polynomial



fitting degree 1 with 8 features...

- C-Val cMSE: 2.2669
- C-Val cMSE (std):0.4205
- Test cMSE: 3.3515

fitting degree 2 with 36 features...

- C-Val cMSE: 2.5718
- C-Val cMSE (std):0.3549
- Test cMSE: 3.2003

Task [3.2] – **Training Models Without Imputation**

Models that can directly handle missing data without the need for imputation.

- **HistGradientBoostingRegressor** (tree-based Scikit-Learn model) : A gradient-boosting method optimized for handling large datasets with missing values. Hyperparameters used : max_iter=1000, max_depth=15, learning_rate=0.001.

    - C-Val cMSE: 3.0078 ; C-Val cMSE (std): 1.0312; Test Set cMSE: 1.5313

CatBoostRegressor with support for censored data using the Survival AFT loss function: A powerful boosting method that supports categorical features natively and handles missing data seamlessly. The standard regression uses regular regression for continuos target variables.

- C-Val cMSE: 3.3871 ; C-Val cMSE (std): 0.9525 ; Test Set cMSE: 1.6611

CatBoost with Survival AFT: Applied the Accelerated Failure Time (AFT) loss function to leverage censored data effectively. It utilizes stratified splits based on the Censored column to ensure balanced training and testing.

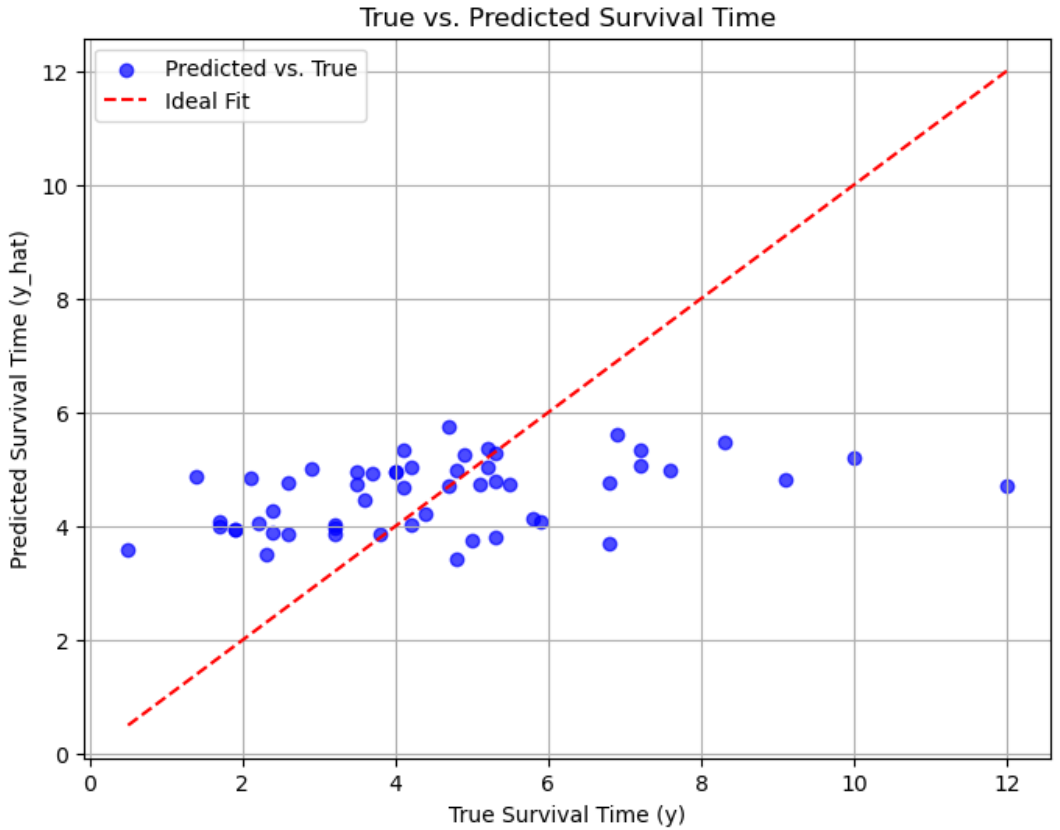
Task [3.3] – **Evaluation**

CatBoost with Survival AFT

Predictons :

- Validation cMSE: 1.53
- Test cMSE: 3.89

Models capable of directly handling missing data, such as HistGradientBoostingRegressor and CatBoost (Survival AFT), proved to be more robust. These methods bypassed the need for explicit imputation, handling incomplete data internally. CatBoost (Survival AFT) stood out by incorporating censored data directly into the training process, offering superior alignment between predicted and actual survival times.



Task [4] – **Semi-supervised learning for unlabeled data**

In this task we utilize both labeled and unlabeled data to enhance the prediction of survival time. This task integrates unsupervised learning techniques, such as dimensionality reduction with Isomap, with supervised learning to handle missing and censored data more effectively.

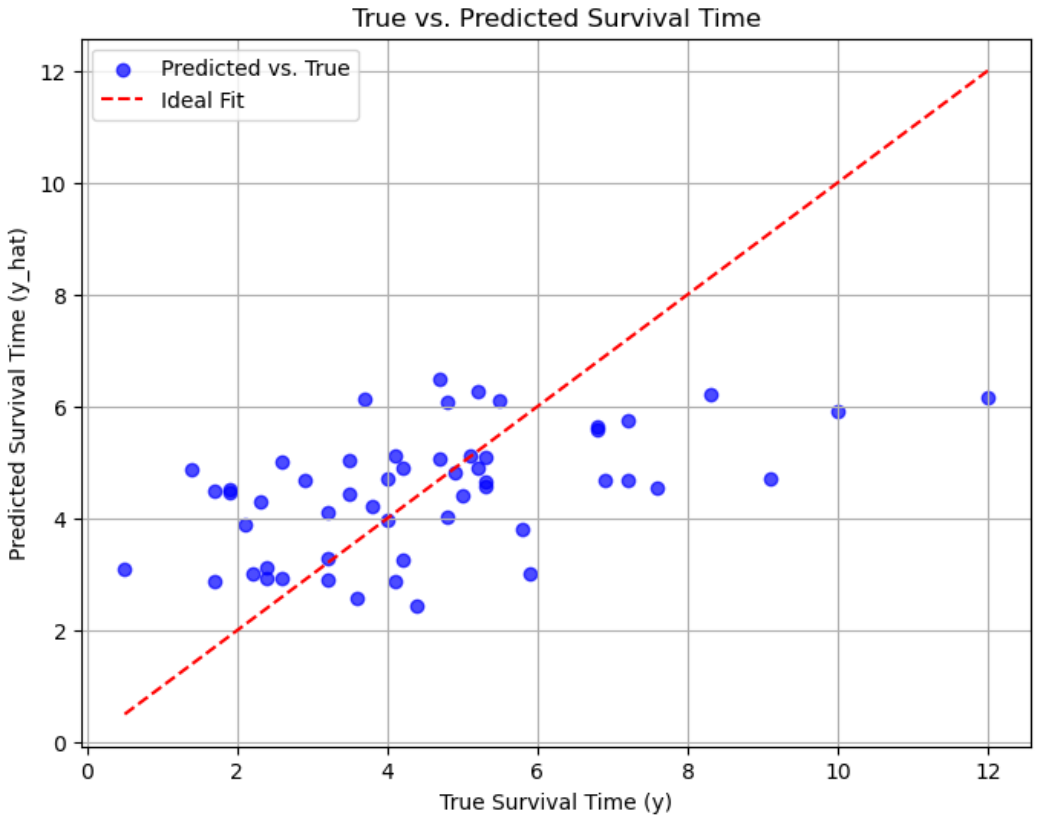
1. Imputation of missing values using the best imputation strategy from Task 3.1.

2. Incorporation of unlabeled data for dimensionality reduction using Isomap.

Task [4] – **KNNImputer trained on the data with and without labels**

Adding unlabeled data for imputation slightly improved model performance. However, the model still struggled with complex patterns due to the linear assumption.

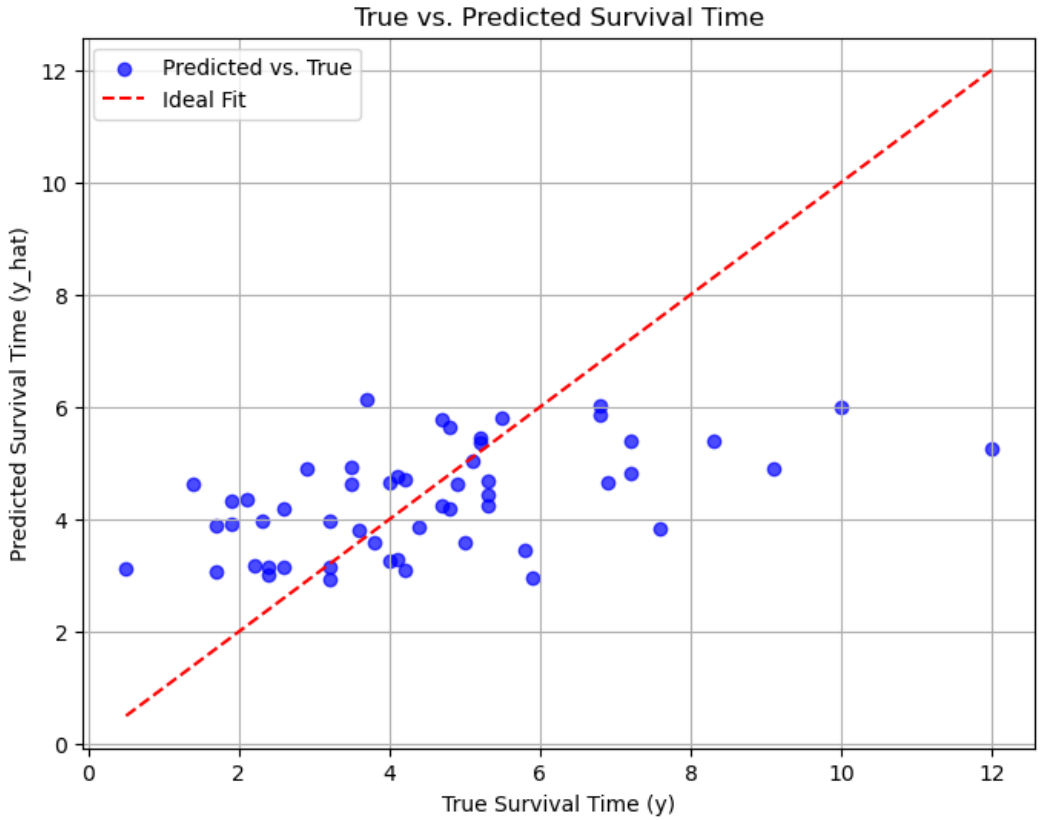
Cross-Validation cMSE (mean): 2.2871
Cross-Validation cMSE (std): 0.5165
Test Set cMSE: 3.3303



Task [4] – **Isomap for Dimensionality Reduction**

- Combined labeled and unlabeled data for training an Isomap model.
- Dimensionality was reduced using Isomap (3 components) after scaling the data.
- The resulting lower-dimensional representation was fed into a Linear Regression pipeline

C-Val cMSE: 2.3056
C-Val cMSE (std): 0.4237
Test cMSE: 3.4656



What went wrong

- Attempting to use Scikit-Learn's KFold for cross-validation initially yielded inconsistent and strange results. This led to the development of a custom cross-validation function, which resolved the issue but added extra time and complexity.

- The performance of SimpleImputer and IterativeImputer was unexpectedly similar, which made it unclear why no noticeable improvement occurred with the more sophisticated IterativeImputer. This raised questions about potential data limitations or improper imputation parameterization.

- Some imputation methods struggled to address more complex patterns in missing data, especially when interacting with censored variables, limiting their ability to enhance predictive performance.

What went ok

- The workflow for the project progressed steadily, with no major blockers that required abandoning approaches or starting over from scratch.

- The baseline Linear Regression and Gradient Descent models performed as expected, providing reliable starting points for comparison with more advanced techniques.

- While not the most accurate, the KNNImputer provided reasonable results and demonstrated its capability to handle missing data effectively when local patterns were present.

What went great

- Developing a custom cross-validation function resolved the issues faced with KFold, ensuring consistent and interpretable performance evaluation across models.

- The application of CatBoost's Survival AFT loss function was a highlight of the project. It effectively handled censored data and achieved superior performance, particularly in capturing complex survival patterns.

- Leveraging Isomap for dimensionality reduction on labeled and unlabeled data significantly improved the model's ability to capture non-linear relationships, showcasing the potential of semi-supervised techniques in survival analysis.