ML@NOVA:
[João Cristóvão]

The three-body problem

Team identification

Name 1: João Cristóvão

Number 1: 70569

Name 2: Ricardo Rodrigues

Number 2: 72054

Final score: 1.48701

Leaderboard private ranking: 46

Task [1.1]

What was done in task [1.1]

- In this task we started by analysing the given dataset (X_train) and we plotted some of the trajectories in there. We observed that there were some trajectories where the bodies colided with each other causing the data to contain anomalies.
- Then we proceed to remove all the anomalies in the dataset. By anomalies we mean all the trajectories that have collisions. We did this by iterating through the data. If we found a row, that did not represent the beggining of a trajectory, where t=0 that meant that there has been a collision and therefore that was an anomaly.

What was done in task [1.1]

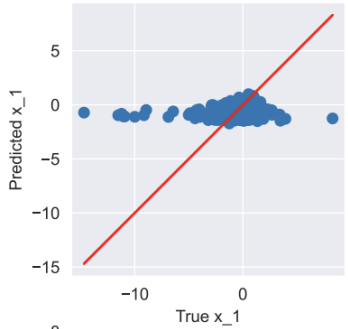
- After having a clean and processed dataset with no anomalies we chose the features and targets of the model. As features we chose the initial positions of each body (x0_1, y0_1, x0_2, y0_2, x0_3, y0_3) and the time (t). As targets, since we want to predict the trajectories of the bodies, we chose the variables of the bodies positions (x_1, y_1, x_2, y_2, x_3, y_3).
- To add the initial positions of each body to the dataframe we iterated through each trajectory in the processed dataset and stored, in a list, the values of the positions in the initial row of each trajectory. Then, we added the columns of the features to the dataframe.
- Having the dataframe ready, we then splitted the data into training, validation and testing.

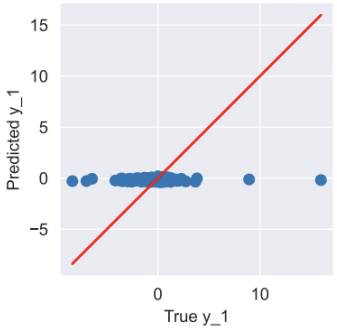
What was done in task [1.2]

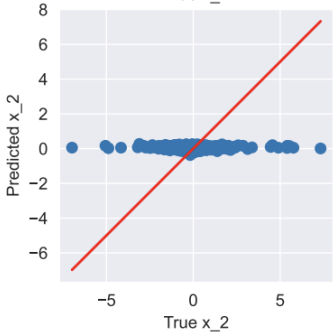
- In this task we create a baseline model using Linear Regression. We created a pipeline with a StandardScaler and with a Linear Regression.
- We then trained the model by fitting the training data to it.
- After training the model we then used the model to predict the trajectories of the bodies.
- We analysed the performance of our model by calculating the Root Mean Squared Error (RMSE) and visualizing the y_yhat plot.

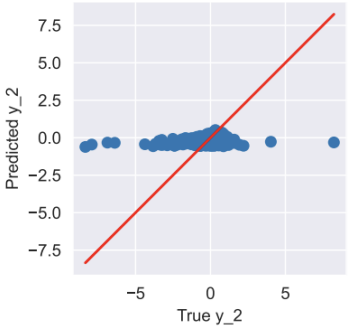
What was done in task [1.2]

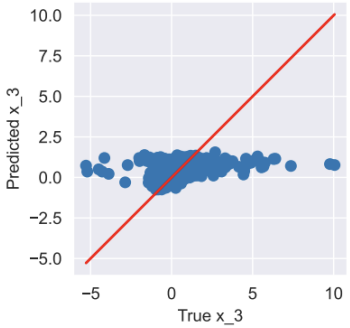
- Here are the plots obtained for the baseline model and the respective RMSE.
- **RMSE: 1.4403**
- This RMSE value tells us that there was a big deviation between the predicted trajectories and the observed ones.
- We can easily conclude that the model did not perform well on predicting the trajectories. This is proven, not only by the RMSE value but also by the plots. As we can see the predicted values did not correspond to the real/observed ones, being that the reason why the majority of the points were not fitted in the red line, that represents the ideal prediction.

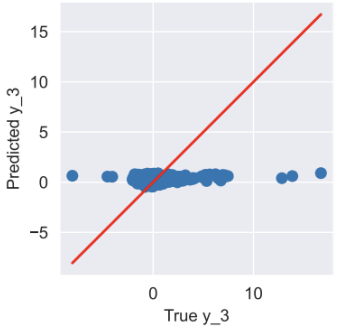












Task [2.1]

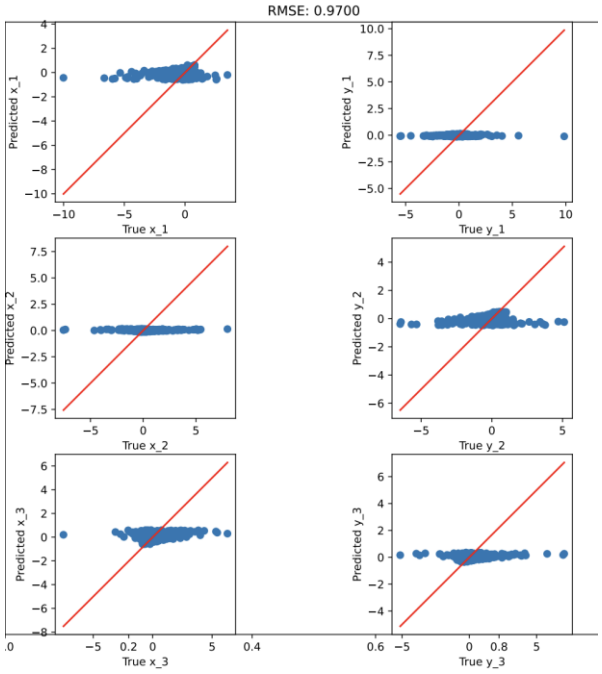
What was done in task [2.1]

- In this task we created a nonlinear model using Polynomial Regression
- We trained the model with diferente degrees and we concluded that the best performing degree for our model was the degree 6.
- During the development of this model we noticed that the number of parameters grows with the polynomial degree chose for the Polynomial Regression Model.

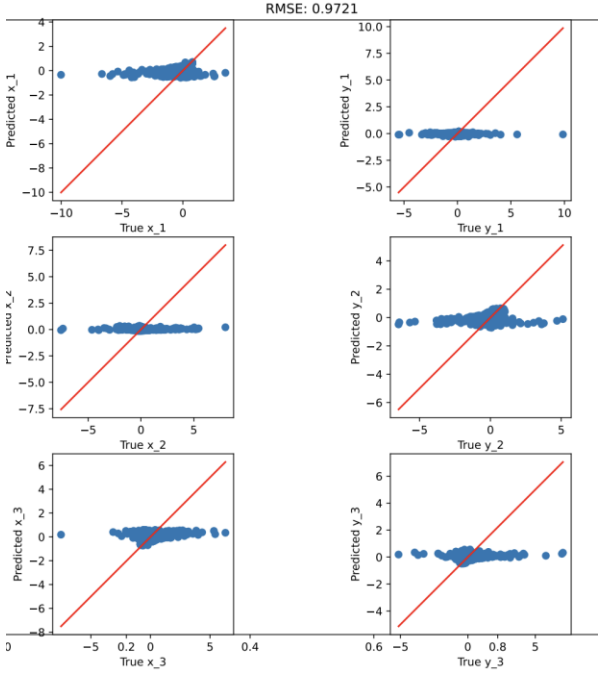
What was done in task [2.1]

- In this task we ran our model 10 times and the polynomial degree, between 1 and 9, with the best RMSE on average was the degree 6.
- With this we concluded that the best model for this task was a Polynomial Regression Model with degree 6.
- In the following slides we have the plots and RMSE values obtained for one of the runs of the model.

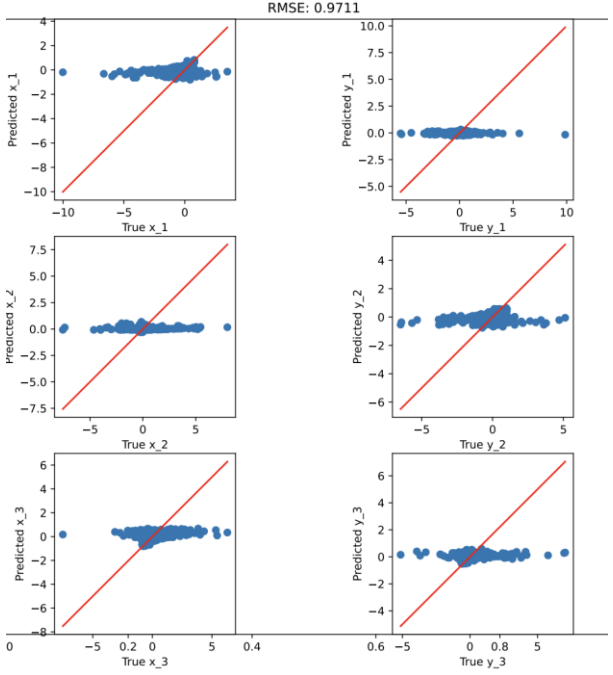
DEGREE: 1
RMSE: 0.9700



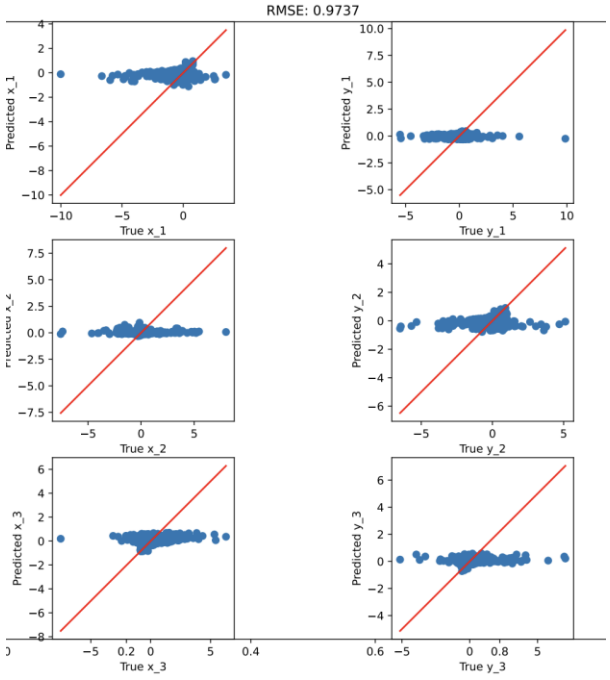
DEGREE: 2
RMSE: 0.9721



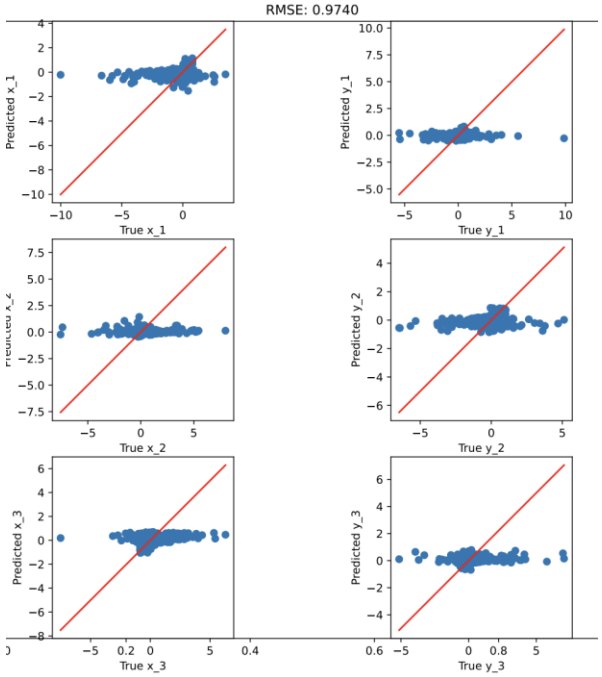
DEGREE: 3
RMSE: 0.9711



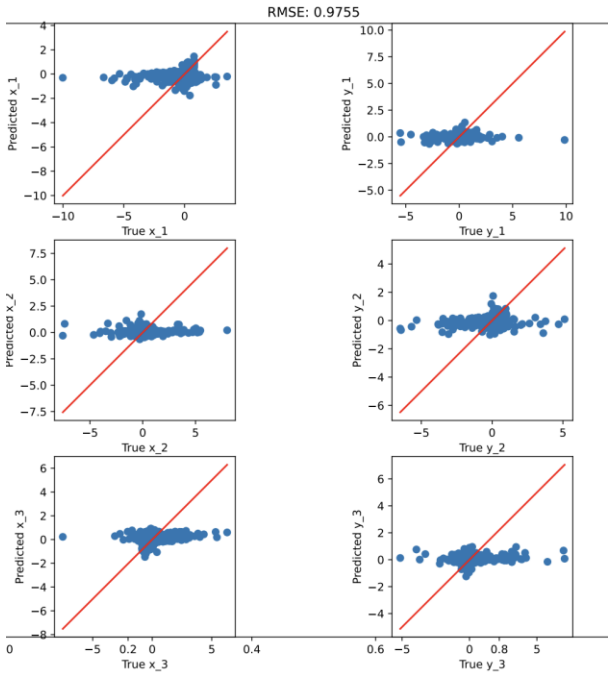
DEGREE: 4
RMSE: 0.9737



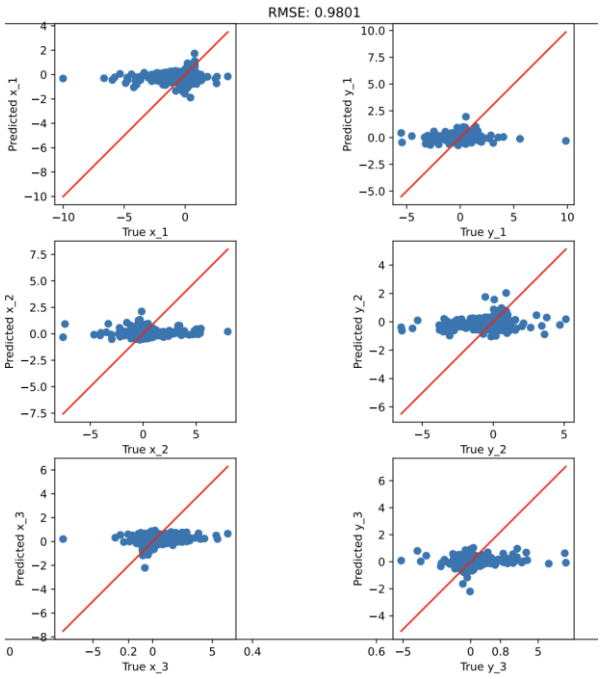
DEGREE: 5
RMSE: 0.9740



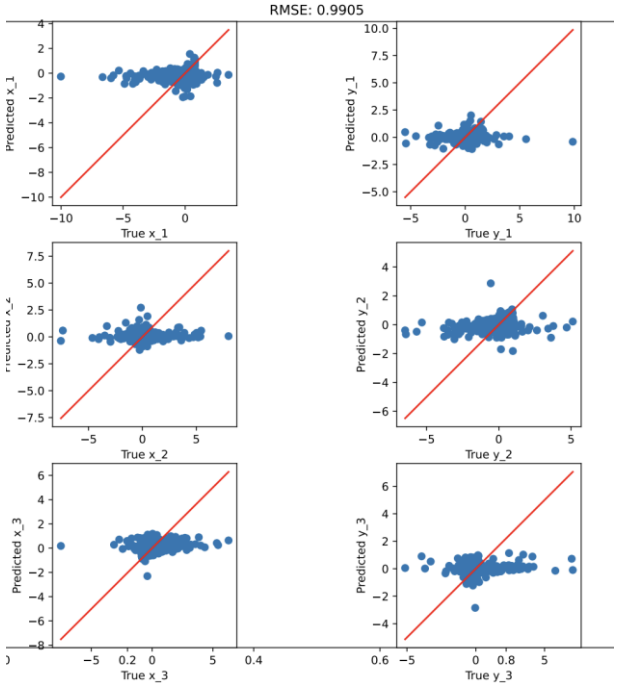
DEGREE: 6
RMSE: 0.9755



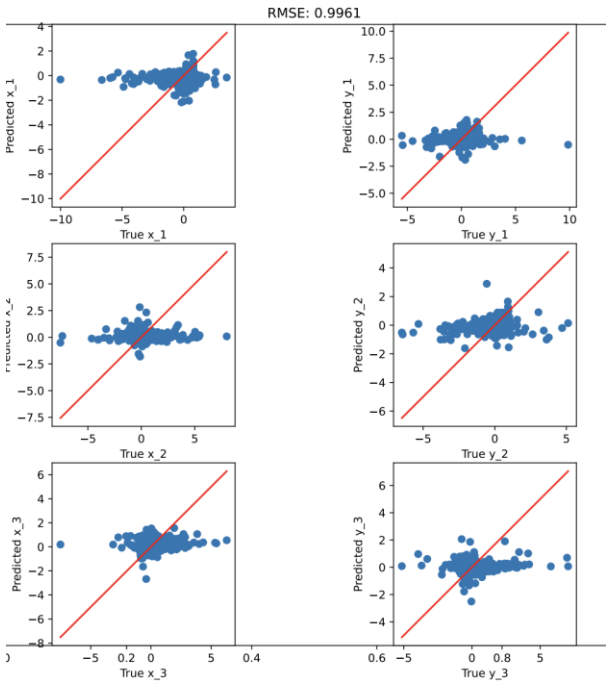
DEGREE: 7
RMSE: 0.9801



DEGREE: 8
RMSE: 0.9905



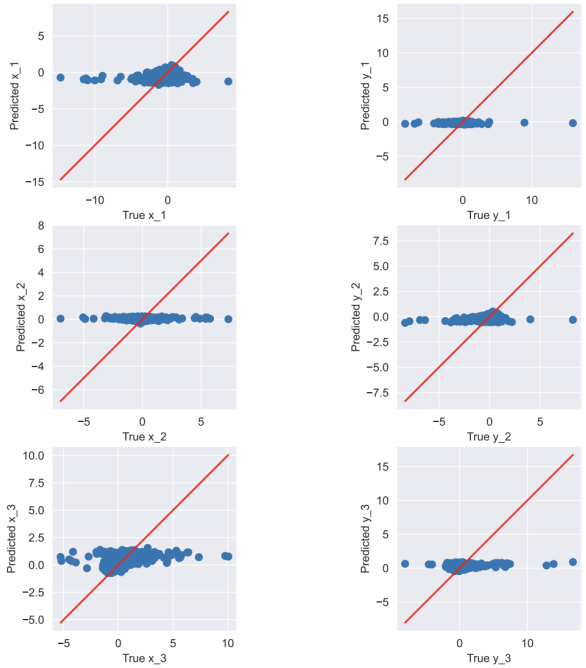
DEGREE: 9
RMSE: 0.9961



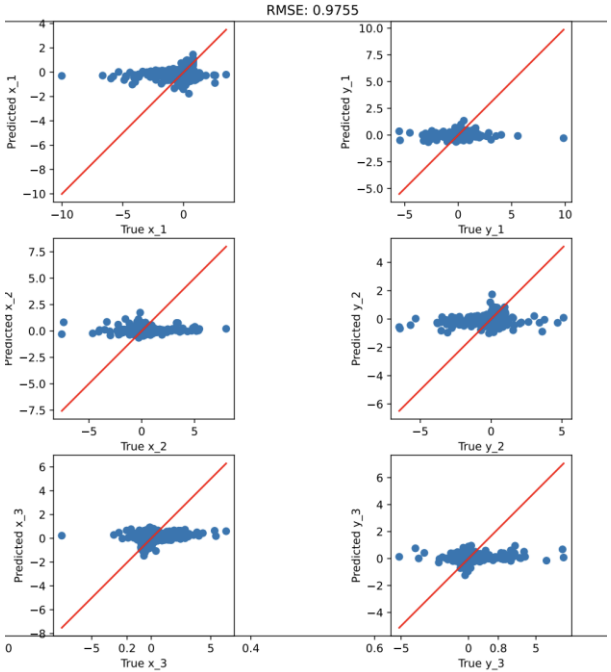
Task [2.2]

What was done in task [2.2]

- Comparing our nonlinear model (Polynomial Regression) against the baseline (Linear Regression) we clearly see an improvement in the RMSE, comproved by the plots drawn. In the nonlinear model y_yhat plot we can see that the points got closer to the line, meaning that the predicted values got closer to the real/observed values.
- The baseline model got a RMSE value of 1.4403, while the Polynomial Regression Model obtained a RMSE value of 0.9755.



Linear Regression Model

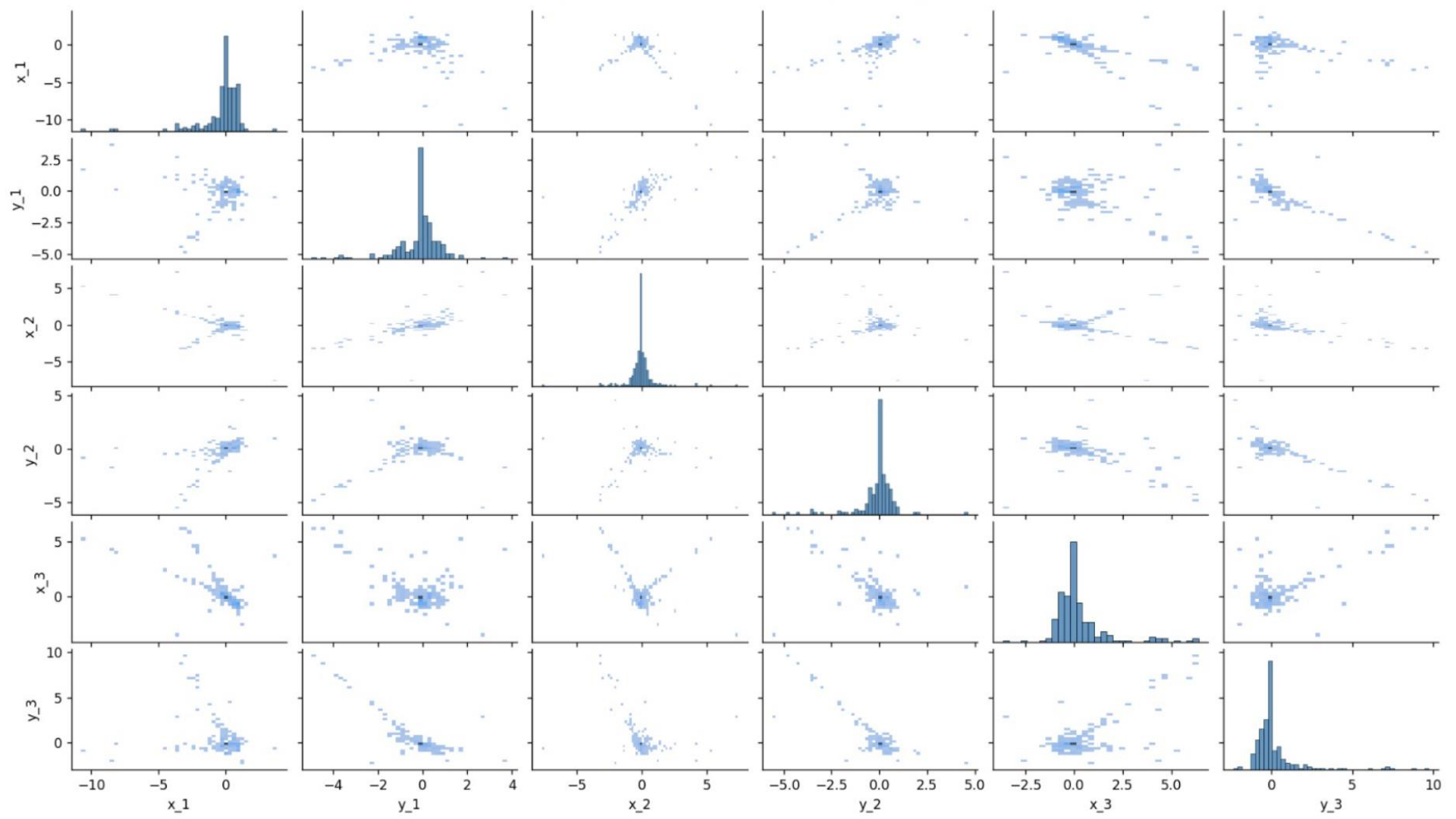


Polynomial Regression (degree 6)

Task [3.1]

What was done in task [3.1]

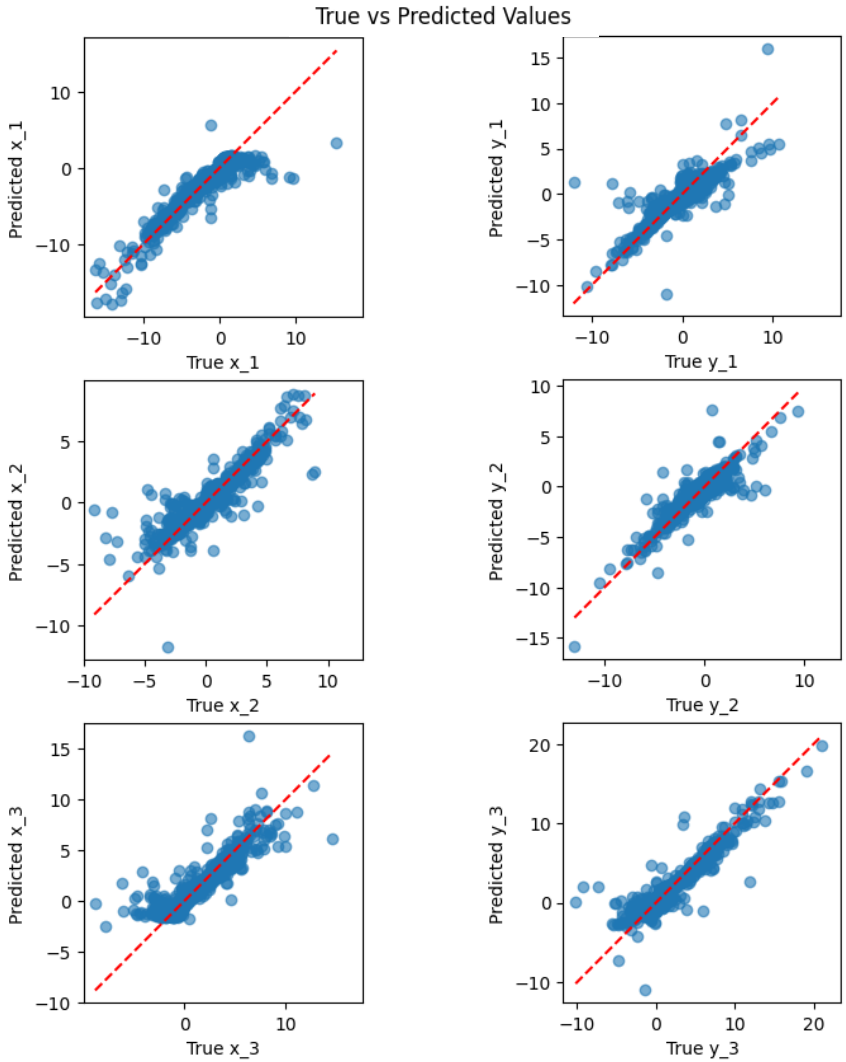
- As mentioned before, we chose the features: x0_1, y0_1, x0_2, y0_2, x0_3, y0_3, t. In this task we analyzed the correlations between the features and the targets variables. To do this we used a pairplot (shown in the next slide) and we took the following conclusions:
  - The feature y_3 was correlated with many other features
  - The feature x_3 was also correlated with other features
- Since these two features were very correlated with others, we decided to remove them from our model since they were redundant and were not affecting the model.


x_1
y_1
x_2
y_2
x_3
y_3

Task [3.3]

What was done in task [3.3]

- In this task, we focused on **feature engineering** using insights from the physics of the **three-body problem**. The goal was to improve model performance by adding meaningful features, such as distances between the bodies and their accelerations, based on Newtonian mechanics.
  - the distances between the bodies (d_12, d_13, d_23)
  - and the acceleration of each body (ax_1, ax_2, ax_3, ay_1, ay_2, ay_3)
- By analyzing pairplots, we noticed relationships between body positions, distances, and velocities, which led us to engineer new features that capture the system's dynamics more effectively.



Task [3.4]

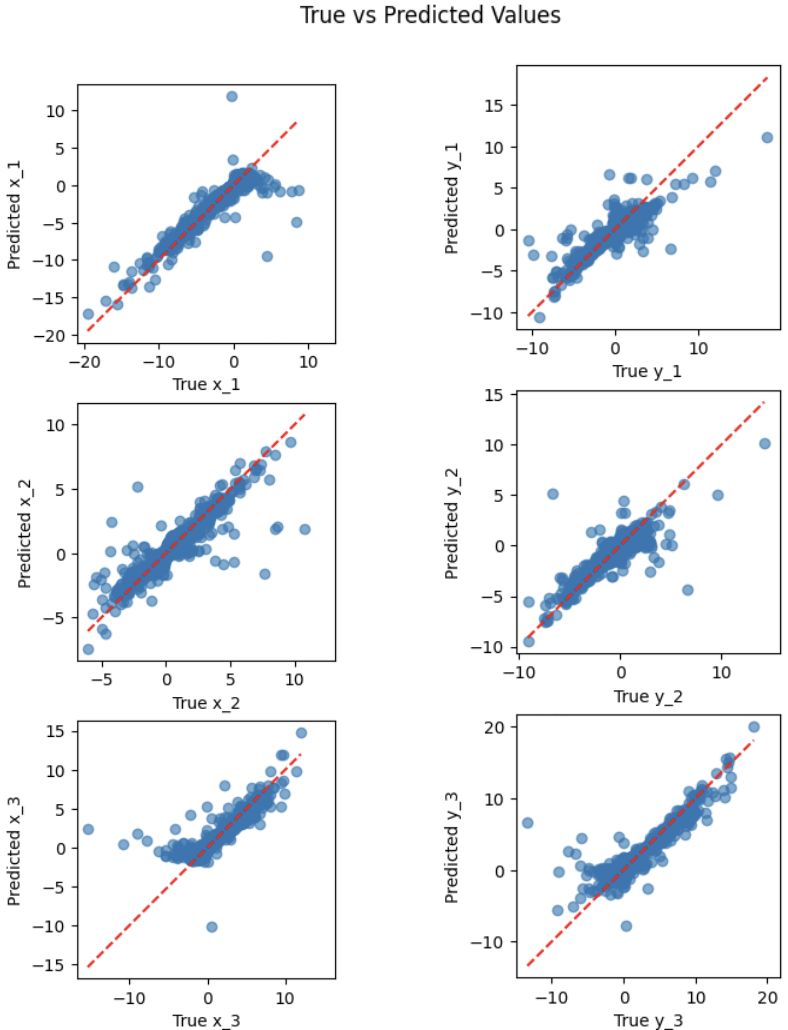
What was done in task [3.4]

Model Training:

We tested various models, but **K-Nearest Neighbors (KNN, n = 6 (Task 4))** performed the best, with an **RMSE of 0.75-0.80**, significantly better than the polynomial regression model (RMSE 0.96-0.98). And both were also considerably better than the models from task2. (As you can see by the plots on the right)

Conclusion:

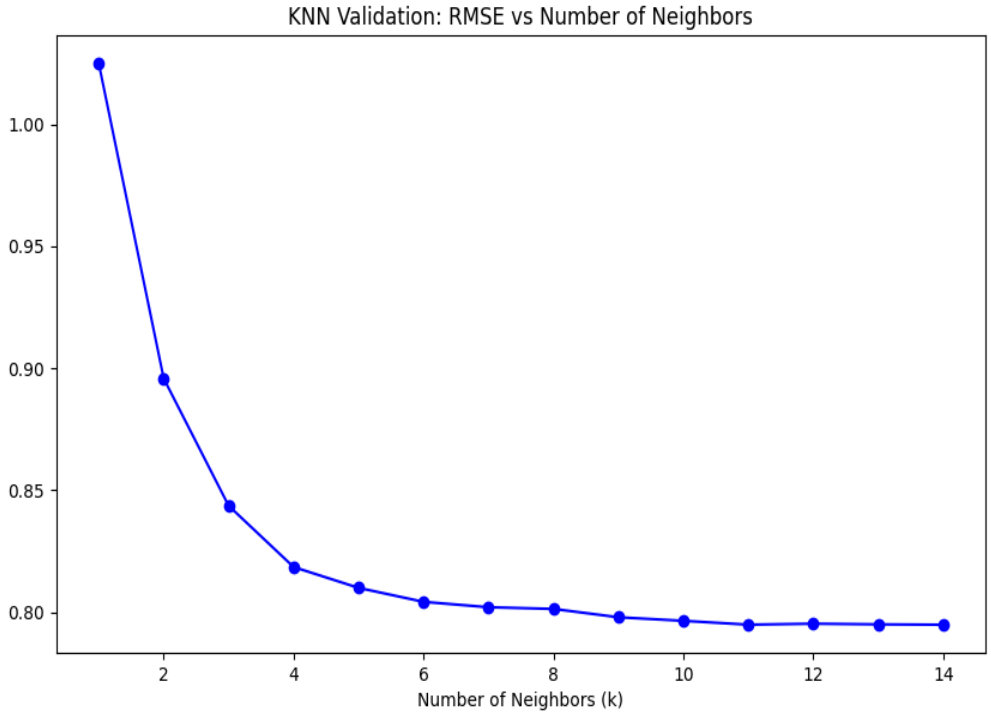
- Incorporating physics-based features (distances, accelerations) and polynomial transformations improved our model's ability to predict body trajectories. However, **KNN with n = 6** was the clear standout, yielding significantly lower RMSEs compared to other models. This indicates that **local relationships in the data** are key to effective prediction in this problem, which KNN captures well.

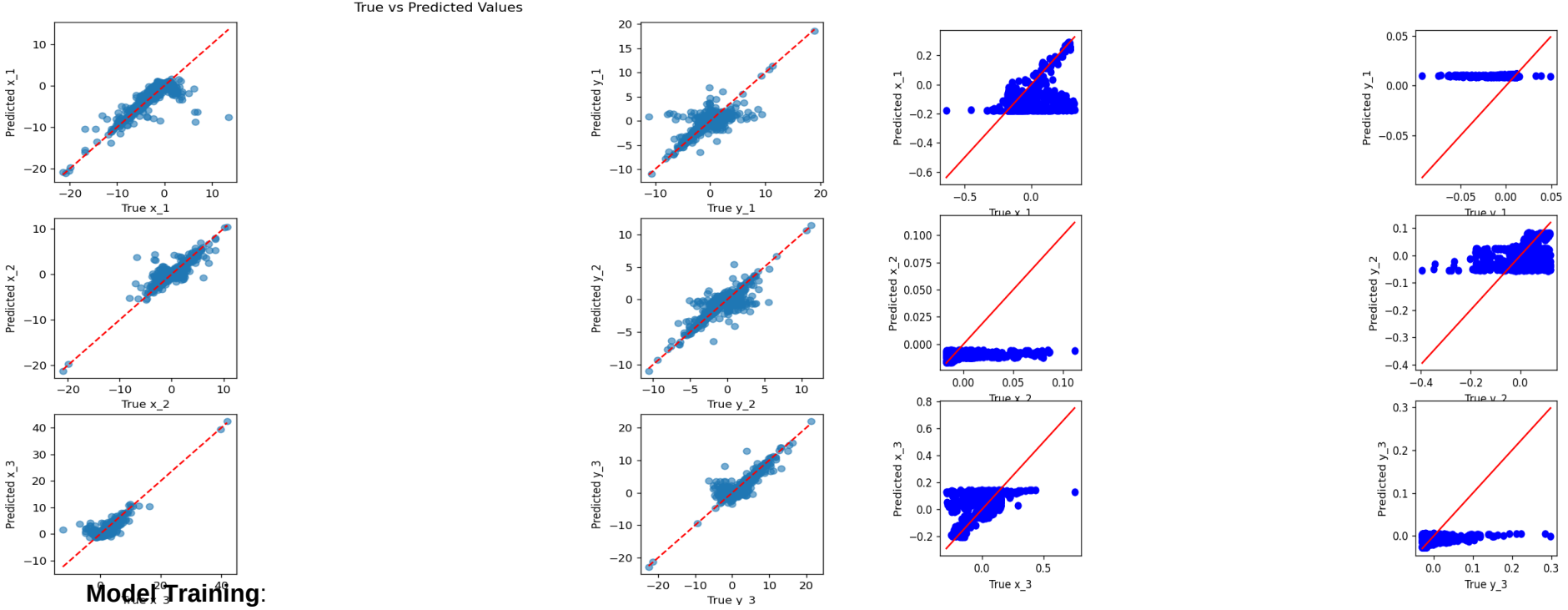


Task [4.1]

What was done in task [4.1]

- In this task we developed a new model, this time a nonparametric one, using the K-Nearest-Neighbors Regressor (KNN Regressor).
  - k = 1: RMSE = 1.0251
  - k = 2: RMSE = 0.8958
  - k = 3: RMSE = 0.8436
  - k = 4: RMSE = 0.8185
  - k = 5: RMSE = 0.8099
  - k = 6: RMSE = 0.8042
  - k = 7: RMSE = 0.8020
  - k = 8: RMSE = 0.8013
  - k = 9: RMSE = 0.7979
  - k = 10: RMSE = 0.7964
  - k = 11: RMSE = 0.7948
  - k = 12: RMSE = 0.7952
  - k = 13: RMSE = 0.7950
  - k = 14: RMSE = 0.7948





Model Training:

(**KNN**) **Augmented** (**polynomial model**)

We tested various models, but **K-Nearest Neighbors (KNN, n = 9)** performed the best, with an **RMSE of 0.75-0.80**, significantly better than the polynomial regression model (RMSE 0.96-0.98). And both were also considerably better than the models from task2.

Conclusion:

Incorporating physics-based features (distances, accelerations) and polynomial transformations improved our model's ability to predict body trajectories. However, **KNN** was the clear standout, yielding significantly lower RMSEs compared to other models. This indicates that **local relationships in the data** are key to effective prediction in this problem, which KNN captures well.

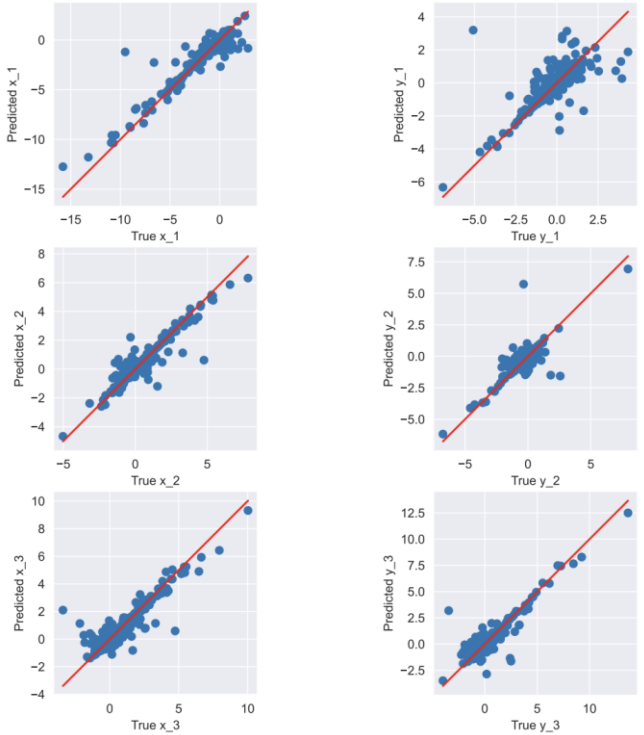
Task [4.2]

What was done in task [4.2]

- We did the following comparisons:
  - KNN Regression Model vs Baseline Model (Linear Regression)
  - KNN Regression Model vs Polynomial Regression Model (Linear Regression)
  - KNN Regression Model vs Best Model of Task 3(Augmented Polynomial)
- The analysis and results are in the following slides

KNN Regression Model

K: 9
RMSE: 0.8028

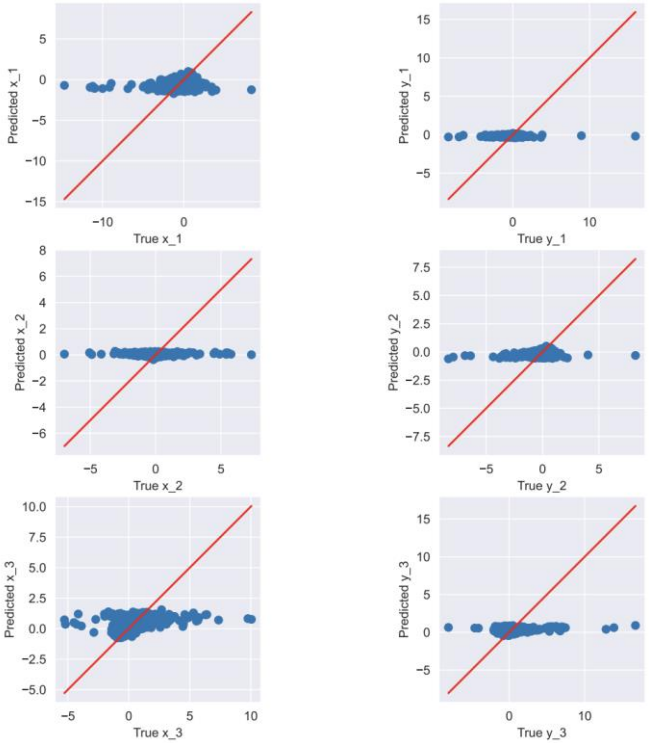


VS

As shown in the plots, there is a very significant difference between the results obtained by the KNN Regression Model and the ones obtained by the Linear Regression Model. We can easily see that the KNN model performed much better than the baseline model. The nonparametric model, for k=9, obtained a RMSE value of 0.8028. On the other side, the RMSE value for the baseline model was 1.4403, which shows a significant difference. The Linear Regression Model is the less accurate of the ones analyzed in this assignment.
The much lower RMSE value, tell us that the KNN, as a nonparametric model, provides much more accurate predictions.
This difference is proven in the plots drawn, where we can easily observe that for the baseline model the majority of the points were not fitted to the line, which proves that the predicted trajectories did not correspond to the observed ones. On the other side, the KNN model plots indicate that it has a better and closer fit to the data, which proves that is capturing the underlying patterns in the data more effectively.

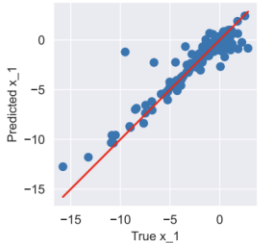
Linear Regression Model

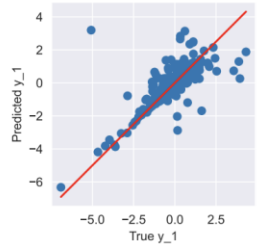
RMSE: 1.4403

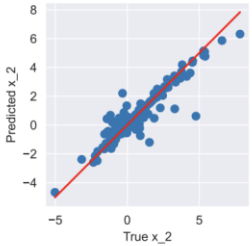


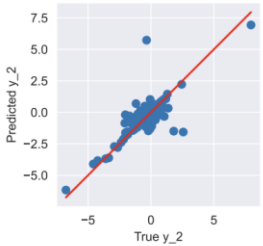
KNN Regression Model

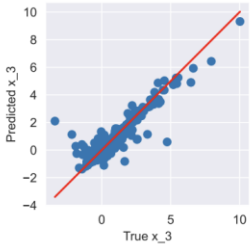
K: 9
RMSE: 0.8028

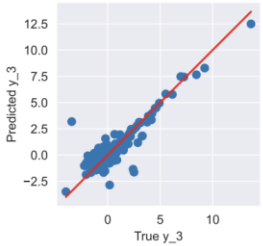












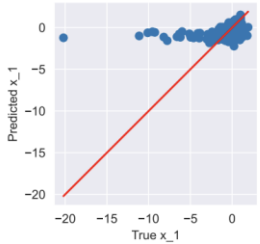
VS

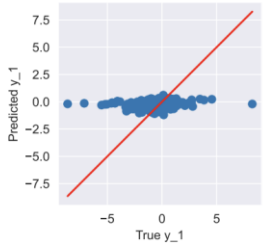
Although the Polynomial Regression Model had better results than the baseline model, it is still far away from the results obtained by the KNN model.
For degree 6 the nonlinear model computed the RMSE value of 1.4164, which is better than the values obtained in the Linear Regression Model.

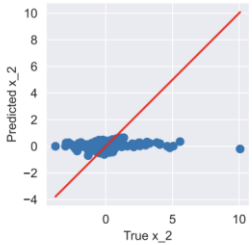
Once again, as shown in the plots, there is still a very significant difference between the results obtained by the KNN Regression Model and the ones obtained by the Polynomial Regression Model. We can easily see that the KNN model performed much better than the baseline model. The nonparametric model, for k=9, obtained a RMSE value of 0.8028. On the other side, the RMSE value for the polynomial model was 1.4164, which shows a significant difference. The much lower RMSE value, tell us that the KNN, as a nonparametric model, provides much more accurate predictions.
This difference is proven in the plots drawn, where we can easily observe that for the baseline model the majority of the points were not fitted to the line, which proves that the predicted trajectories did not correspond to the observed ones. On the other side, the KNN model plots indicate that it has a better and closer fit to the data, which proves that is capturing the underlying patterns in the data more effectively.

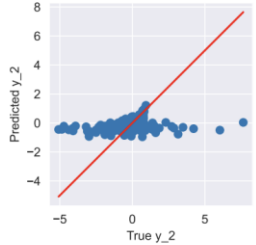
Polynomial Regression Model

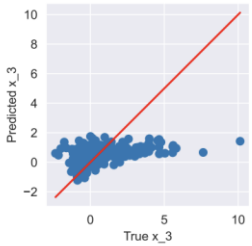
Degree: 6
RMSE: 1.4164

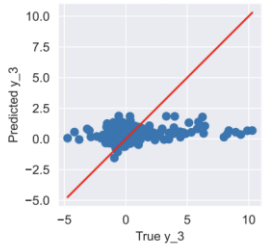






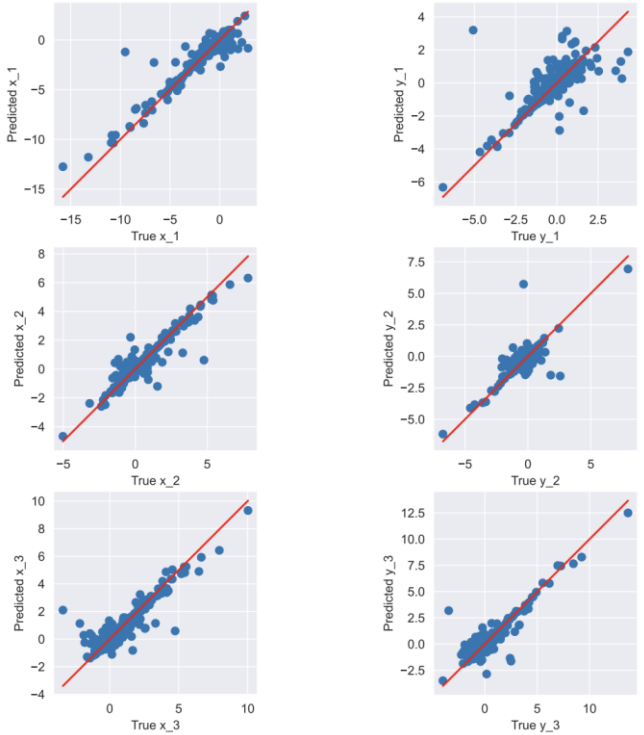






KNN Regression Model VS Augmented Linear Regression

K: 9
RMSE: 0.8028

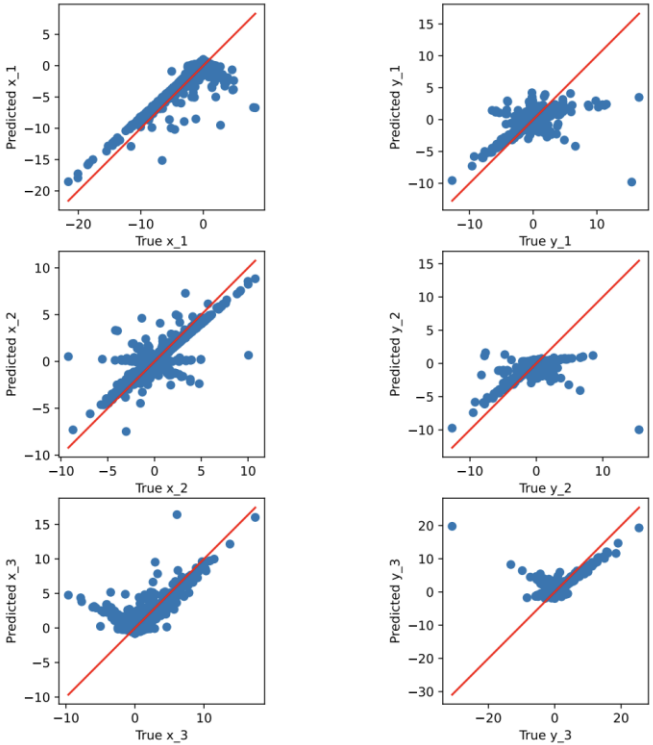


Now comparing the **KNN Regression Model (K = 9)** with the **Augmented Linear Regression Model**, the results show that the KNN model still performs slightly better. The KNN model achieves an **RMSE of 0.8028**, whereas the Augmented Linear Regression has a higher **RMSE of 1.0429**.

Although the Augmented Linear Regression model represents a significant improvement over standard linear regression, it still falls short in predictive accuracy compared to KNN. The plots provide further insight into this performance gap. For the Augmented Linear Regression, the points exhibit a larger spread around the diagonal line indicating less accurate predictions. In contrast, the KNN model's predictions are much more tightly clustered around the diagonal line, which shows that it captures the underlying data patterns more effectively.

Therefore, although the Augmented Linear Regression model shows improvement, the **KNN Regression Model** continues to provide superior predictive accuracy, particularly for non-linear relationships in the data.

RMSE: 1.0429



Overall assessment

What went wrong

- We faced a lot of difficulties in the split of the data into training, validation, and test sets. We believe we did not use the test data effectively, which caused the poor results obtained in our submissions. The training, validation, and testing sets should be completely independent of each other, and we must guarantee that we do not have the same initial positions in any two sets. We did not do this, which led to issues with data leakage and invalid performance metrics.
- We inadvertently included overlapping samples between the training, validation, and test sets. This overlap can significantly skew the evaluation of our models, as they may perform well on data they have effectively "seen" before, failing to generalize to truly unseen data. This was particularly evident when we used early iterations of our models to select features and preprocess data, inadvertently introducing bias into the validation and test sets.
- Many models exhibited signs of overfitting, indicated by low RMSE values on the training data but significantly higher RMSE values on the test data. This discrepancy highlighted that our models were learning noise and specifics of the training data rather than underlying patterns. We relied too heavily on complex models without adequate regularization or validation strategies to curb this overfitting.
- In our experimentation, we also failed to implement cross-validation properly, which would have provided a more robust mechanism to assess model performance and mitigate overfitting.
- For time-dependent datasets, ensuring that we did not include the same initial positions in multiple sets was crucial. We neglected to preserve the temporal order of our data, which is essential for time series analysis. This oversight not only affected the integrity of our training and validation sets but also compromised our test set.
- All these conclusions were reached close to the project deadline. Since we are limited to a certain number of submissions per day (five), we could not adequately improve our overall score. In future projects, we must ensure to start our evaluations and optimizations earlier to maximize our potential for improvement and success.

What went great

- During this assignment we explored multiple machine learning models, such as K-Nearest Neighbors Regression, Linear regression, and Polynomial Regression. This broad experimentation allowed us to test both linear and non-linear approaches, providing valuable insights into which methods worked best for the Three Body Problem.
- Despite facing a lot of challenges related to data leakage and overfitting, we were able to identify these problems and tried to solve them. Instead of being discouraged, we took these issues as valuable learning experiences, which will greatly benefit us in the next assignment. Even though the results were not the desired we believe we learned a lot from this assignment and feel much more prepared for the second assignment.