**Data Selection:**

A large number of csv files are provided. The simplest thing to do was **to take a random sub-sample with uniform distribution and check if it was significant or not**. If it's reasonably significant, we'll keep it. If it's not, we'll take another sample and repeat the procedure until we get a good significance level. Initially, I considered 1- CSV files that have more than enough data to be split in the training and validation set.

**Feature Selection:**

There are three main goals to feature selection. Improve the accuracy with which the model is able to predict for new data. Reduce computational cost. Produce a more interpretable model.

Through the given data I am trying to capture as much information as necessary for the controller. The given data consists of Lidar data, robot position, local goal position, and final goal position. The features defined are as follows:

* Distance from local goal position

IMAGES if possible

* Distance from final goal position
* Angular error in shortest trajectory to local goal and vehicle heading angle

A picture containing text, whiteboard

Description automatically generated

Fig. Angular error

* Angular error in shortest trajectory to final goal and vehicle heading angle
* Lidar output in LOS (Line of Sight) of the robot
* Lidar output towards the shortest view for the local goal
* Lidar output towards the shortest view for the final goal
* Maximum distance the robot can travel towards the shortest trajectory for the local goal

A whiteboard with writing on it

Description automatically generated with medium confidence

A whiteboard with writing on it

Description automatically generated with medium confidence

* Maximum distance the robot can travel towards the shortest trajectory for the final goal
* Note: The limitation in output velocity and omega will also be considered

There can be more such features

Model selection:

Linear Ridge Regression:

**Velocity Prediction:**

Parameter Tuning:

Graphical user interface

Description automatically generated with medium confidence

Learning Curve:

Chart

Description automatically generated

Scoring and error:

R2 scores:

|  |  |
| --- | --- |
| Validation Scores | |
| Regularization parameter alpha | Scores |
| 0 | 0.6125 |
| 0.2 | 0.6126 |
| 0.4 | 0.6126 |
| 0.6 | 0.6126 |
| 0.8 | 0.6126 |
| 0.9 | 0.6126 |

**Tuned parameter value = 0.9**

|  |  |
| --- | --- |
| Training Scores | |
| Regularization parameter alpha | Scores |
| 0 | 0.6272781927749702 |
| 0.2 | 0.6274001578826105 |
| 0.4 | 0.627400480084319 |
| 0.6 | 0.6274008021823565 |
| 0.8 | 0.6274011241552864 |
| 0.9 | 0.6274012851134265 |
|  |  |

**Out-sample/Testing Score:**

R2 score = 0.6074012851134265

**Omega** **Prediction**:

Learning Curve:

Chart

Description automatically generated

**SVM:**

The implementation is based on libsvm. The fit time scales at least quadratically with the number of samples and may be impractical beyond tens of thousands of samples. For large datasets consider using [**LinearSVC**](https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html#sklearn.svm.LinearSVC) or [**SGDClassifier**](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html#sklearn.linear_model.SGDClassifier) instead, possibly after a [**Nystroem**](https://scikit-learn.org/stable/modules/generated/sklearn.kernel_approximation.Nystroem.html#sklearn.kernel_approximation.Nystroem) transformer.

Hyperparameter tuning:

* RBF Kernel:

Regularization:

Plotted against 1/alpha

Chart, line chart

Description automatically generated

R2 Score for Velocity = 0.483131221616796

Learning Curve:

Chart

Description automatically generated

Omega:

Chart, line chart

Description automatically generated

Learning Curve for Omega:

Chart

Description automatically generated

Outsample score = 0.04

* **Linear Kernel**: was not able to solve
* **Poly Kernel**: degree = 3, iterated through regularization,

Regularization: Plotted against 1/alpha

Chart, line chart

Description automatically generated

Outsample R2 score = 0.10818291218618581

Learning Curve:

Chart

Description automatically generated

* Sigmoid Kernel:

The output from these kernel was negative R2 scores which meant that the data is fitting very badly with the predicted weights.