Tracking particles coordinates in RSD sensors.

Multi-Output Regression Task.

Silva Bashllari
Politecnico di Torino
Student number: [redacted]
[redacted]@studenti.polito.it

Baharak Qaderi
Politecnico di Torino
Student number: [redacted]
[redacted]@studenti.polito.it

***Abstract*—The aim of this project is to provide an approach to building a multi-output regression pipeline to predict the coordinates (x,y) which indicate where a particle of interest passed given data collected from signals in an RSD sensor (Resistive Silicon Detector). In particular, we perform an extensive analysis of all the possible steps to follow in the design of the regression model. Our approach can be best described as a mainly brute force one due to the fact that we explore a wide variety of models and preprocessing techniques, given the fact that we have limited domain knowledge. Our best pipeline outperforms the given baseline in the leaderboard of 6.629 reaching a score of 5.417 by using the Random Forest Regressor.**

I. Problem overview

In recent years, great attention has been placed on understanding how we can detect the positions where different particles, like electrons, pass in their trajectories.The aim of this project is to provide an approach to building a multi-output regression pipeline to predict the coordinates (x,y) which indicate where a particle of interest passed given data collected from signals in an RSD sensor (Resistive Silicon Detector). This sensor is composed of 12 pads and each pad provides information regarding the signal it reads. In our case, every time a particle passes through the sensor (called an *event*) we have the following signal information being collected: the pmax (peak of the signal), negative pmax, tmax (the time at which the signal achieves its maximum value), the area (the integral) and the RMS (root mean square error). We have been provided with two data sets, comprised of 514,000 events namely:

1) the development set - composed of 385,500 events;
2) the evaluation set - composed of 128,500 events.

From the variables perspective, the data sets are composed of 90 input features, a relatively large set. However, the usual rule of thumb is that the regression task normally requires a minimum of 30 observations. In multiple regression, usually are added at least an additional 10 observations for each additional independent variable added to the equation[2]. Our dataset adheres to this rule of thumb. In an initial exploratory analysis, we discovered that there are no missing values. Furthermore, there are 18 readings for each of the 5 metrics that describe the signal, but there are only 12 pads(inferring, 60 features). Thus, detecting the 6 noise readings (consequently 30 features) becomes of imperative importance. The ultimate goal of the project is to minimize the **Euclidean Distance** between the predicted $\hat{x}$ and $\hat{y}$ and the ground truth x and y values.

II. Proposed approach

A. Preprocessing

1) Data Exploration: As mentioned, the purpose of this project is to predict the ”x” and ”y” coordinates, which, when plotted have the following distribution as in Figure 1. There are a lot of points superimposed on top of each other and there are some ”empty” areas due to the inner workings of the sensor. To detect the 6 noise readings, the histograms of all the

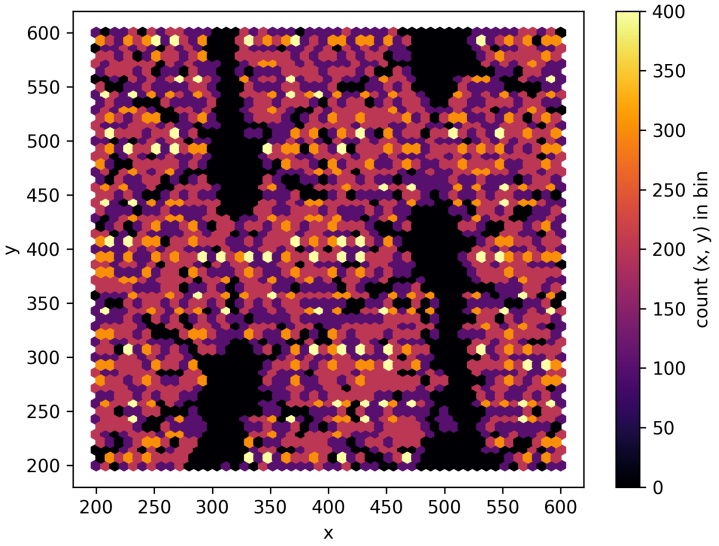

Figure 1: The distribution of ”x” and ”y” features.

features were visualized. Taking into consideration the shape of the distributions and the range of the values, it became clear that the following readings were very different from the rest of the dataset: 0,7,12,15,16 and 17. To illustrate the matter and in order to avoid cluttered images and too many superimposed distributions, a subset of features from ”area” were chosen as exemplary. As it can be observed in Figure 2, the features of area 2,3,4 and 5 have similar distributions whereas area[17] stands clearly apart. Similar observations were made for the other features.

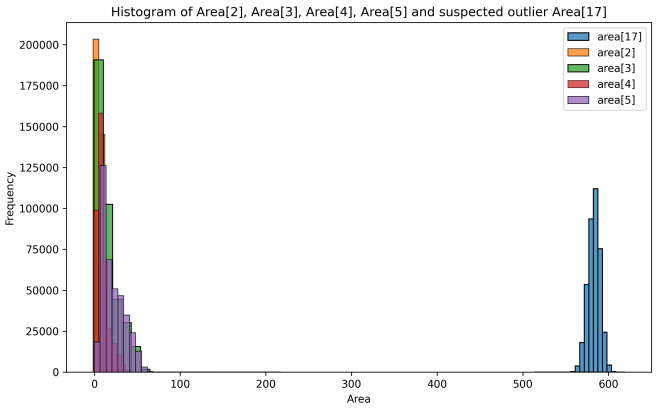

Figure 2: The histograms of a subset of ”area” features.

For further verification, the dataset was transformed using the natural logarithmic scale given that this representation facilitates outlier detection by compressing the dynamic range of data and thus reducing the visual impact of extreme values. To exemplify and illustrate it, the same readings as in Figure 1, namely 2,3,4,5 and 17 were taken but now using a subset of pmax. As it can be observed in Figure 3, the transformed pmax[17] clearly differs from the rest of the features. Similar observations were made for the other features consolidating the fact that the readings from 0,7,12,15,16 and 17 were noise.

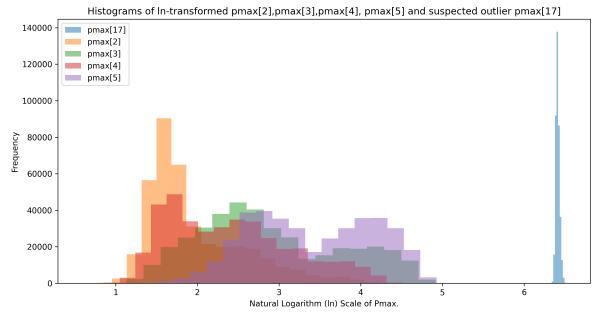

Figure 3: The histograms of a subset of ”pmax” features logarithmic scaled.

One of the usual issues in a regression task is the existence of multi-co-linearity. Removing those 6 suspected noise features and building the correlation matrix as in Figure 4, we can observe micro-clusters of correlation spread relatively uniformly and they are predominantly between ”pmax” and ”area” features. However, those are the very same features that are also mostly correlated with ”x” and ”y” coordinates. While analyzing the domains and distributions of the features, it came to our attention that some of the negpmax had positive values and some of the areas had negative values, which is unexpected. Furthermore, if the features that are considered noise are not taken into account, it can be observed that negpmax and tmax are practically static variables, with very

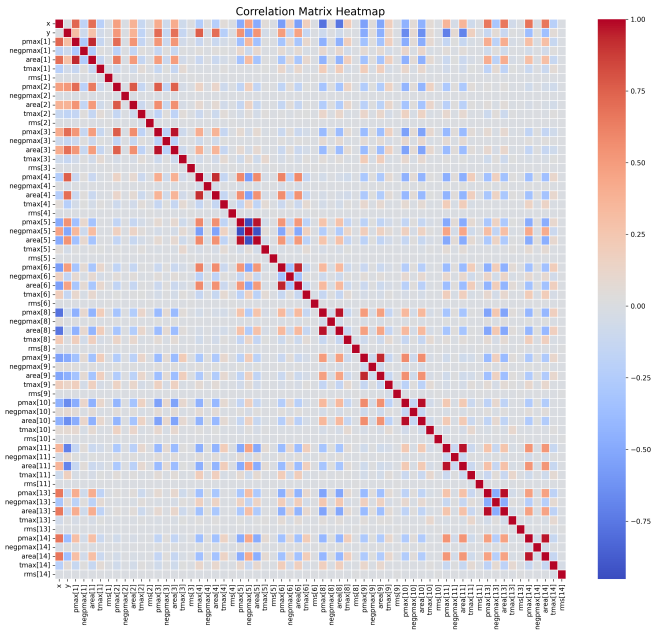

Figure 4: The correlation matrix of all 62 variables.

little variance.

2) Preprocessing: Given the fact that our input features have different domains, we employed different pre-processing techniques, mainly to transform the features to a different scale and domain in order for one feature not to surpass the others in the regression task. The following transformations were applied, at different moments:

1) The Standard Scaler - which does a z-score scaling of the features, bringing the mean to 0 and the standard deviation to 1.
2) The Log Transformation - using the natural logarithm.
3) The Robust Scaler - which uses the median and interquartile range (IQR) making the models less influenced by extreme values.

Furthermore, given the high dimensionality of the dataset and in order to avoid the curse of dimensionality, we experimented with the Principal Component Analysis (PCA) and the Fisher Score ranking in The Fisher Linear Discriminant Analysis (FLDA). In fact, when building the PCA and then plotting the components versus their explained variance, it can be easily understood that less than 30 principal components explain almost 100 percent of the variance, as it can be observed in Figure 5. Furthermore, we also tried to reduce the dimensionality of the dataset in a more manual manner by selecting different subsets of features based on the correlation matrix. For example, given the fact that ”area” and ”pmax” are very correlated, we chose to use only ”pmax” or only ”area” and so on.

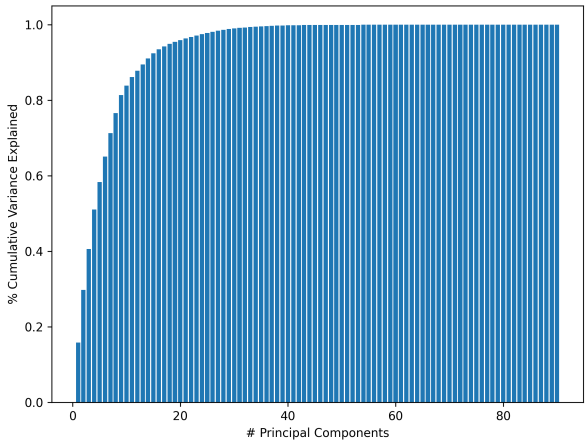

Figure 5: The 90 principal components versus explained variance.

TABLE I: The distance in the evaluation set of linear models.

	Mult. Linear	**Lasso**	**Ridge**
90 input features	17.352	17.360	17.360
60 input features	18.157	18.156	18.157

B. Model Selection

For this task, linear models were initially tested. Given that we lack specific domain expertise, we decided to start with simpler and more interpretable models and then work our way to more complex ones, if needed. The initially tested models were:

- ***Multiple Linear Regression.***
- ***Lasso Regression:*** which was initially trained with an arbitrary choice of the alpha parameter 0.01.
- ***Ridge Regression:*** which was also initially trained with an arbitrary choice of the alpha parameter 0.01.

They were trained using two different development sets: all 90 features and using the 60 features (the subset created by removing the 30 features suspected as outliers as it was described in the above section). The results in terms of the distance in the evaluation set can be observed in Table 1. The experiments show that removing the suspected outlier features does not minimize the distance in these models. This can be for a variety of reasons, for example: removing the noise and the existence of multicollinearity among the remaining features makes the error bigger, the suspected features as noise are not the actual noise ones etc. Thus, we decided to proceed not with manual feature subset selection but by using PCA. As it can be observed in Table 2, we ran multiple experiments, each having as input different versions of the dataset, namely:

- ***Version 1***: building the PCA with all the 90 features, as in Figure 5, it can be observed that 30 principal components explain almost 100 percent of the variance. Thus, version 1 is the dataset with those 30 components.

TABLE II: The distance in the evaluation set of linear models using PCA.

	Mult. Linear	**Lasso**	**Ridge**
Version 1	176.953	176.953	176.953
Version 2	699.758	694.700	699.758

TABLE III: The distance in the evaluation set of tree-based models.

	Random Forest	**Decision Tree**
90 features	***5.585***	8.370
60 features	***5.585***	8.285

- ***Version 2***: standard scaling was applied to the entire dataset, the PCA was constructed and it was observed that 70 principal components were needed in this case to explain almost 100 percent of the variance. This version contains those 70 components.

The results can be observed in Table 2. They clearly indicate that using the PCA with or without prior transformation of the dataset increases substantially the error on the evaluation set. Furthermore, the transformation of features using the natural logarithm was also experimented with but only for a subset of features, given the existence of negative values, for example in negpmax. These variations of the dataset were tested with different models, and they did not yield better results. Given that the existence of outliers highly impacts these models, the robust scaler was also used to transform the dataset and then fitted to the above-mentioned models but this did not yield much better results either. In addition, other dimensionality reduction techniques were tested, like the Fisher Score ranking in The Fisher Linear Discriminant Analysis (FLDA) but it did not yield better results. In the same fashion, the manually selected subsets of the data sets did not provide major improvements on the error. Therefore, we decided to test other methods for feature selection, namely embedded approaches in tree-based models, in which the feature selection occurs naturally as part of the data mining algorithm. Studies have shown that tree-based models can be beneficial when analysing particles in physics [1], therefore, the following models were tested:

- ***Decision Tree Regressor.***
- ***Random Forest Regressor:*** using initially the default values in Sci-Kit Learn, namely: 100 estimators, the criterion was ”the squared error” and the maximum depth was none.

The entirety of the development set with 90 features as well as the version of the development set with 60 features were used to train these models. The results, as they can be observed in Table 3, outperform not only the previous linear models but also, in the case of the Random Forest, the baseline in the leaderboard.

C. Hyperparameters Tuning

1) Decision Tree Regressor Tuning: : For the decision tree a gird search algorithm was used to test the following values for the following hyperparameters:

TABLE IV: The distance in the evaluation set of random forest tunnings.

ESTIMATORS	MAX_DEPTH	TEST DISTANCE
100	NONE	5.585
100	15	5.841
100	40	5.585
100	60	5.585
70	NONE	5.606
130	NONE	5.582
150	NONE	**5.575**

- ***Estimator Max Depth:***20, 30, None.
- ***Estimator Min Samples Split:*** 2, 5.
- ***Estimator Min Samples Leaf:*** 1,2,4.

The best hyper-parameters for the decision tree were: estimator Max Depth =20, Estimator Min Samples Split=2, Estimator Min Samples Leaf=1, by achieving an error= 8.180. However, even after tuning the Decision Tree Regressor, it still did not achieve a better score than the vanilla version of the Random Forest. Consequently, we decided to proceed with tuning the Random Forest Regressor.

2) Random Forest Regressor Tuning: Unfortunately, due to computational limitations, running a grid search with multiple parameters of the Random Forest Regressor proved impossible to carry out. Thus, we trained sequentially different versions of the Random Forest with different hyperparameters as displayed in Table 4. The input included all the 90 features and the hyper-parameter of the error as the default one (namely, squared error). Noting that the estimators are not real hyper-parameters but they are however parameters to the Random Forest Regressor of sklearn library. Last but not least, printing the features ranked by the Random Forest in terms of importance and experimenting with different subsets of them allowed us to reach the best score of **5.417** by using the top 38 features and a Random Forest with 150 estimators. In Figure 6, we can observe the x and the y outputs of this model illustrated using a hexbin plot. The shape of the distribution looks very similar to that of the development set in Figure 1 but less clearly ”cut” due to the presence of errors.

III. Results

One of the main results of these experiments is the fact that tree-based models that carry out embedded feature selection approaches provided better results in terms of minimizing the error. Regarding the Random Forest, it is clear that for this dataset, a higher number of estimators yields a lower error. The linear models did not perform as well and this could be for a variety of reasons, such as but not limited to: the input and the output do not have a linear relationship, the patterns of the data are too complex to be captured by such simplistic models, too many outliers, not only across the features but also within the features might have distorted the results given that these models and the error are sensitive to outliers. The outlier detection was a substantial part of this task. Even though removing the 30 features did not improve the performance of the linear models, our analysis of the distributions in the data exploration section was supported by the Random Forest Regressor which pushed almost entirely those features near the bottom of its ranking. Furthermore, the ranking of the Random Forest of the features is congruent with what can be observed in the Correlation Matrix, specifically that pmax and area seem to be more relevant in general as features. Last but not least, dimensionality reduction techniques such as PCA or Fisher scoring are not so well fitted for this dataset.

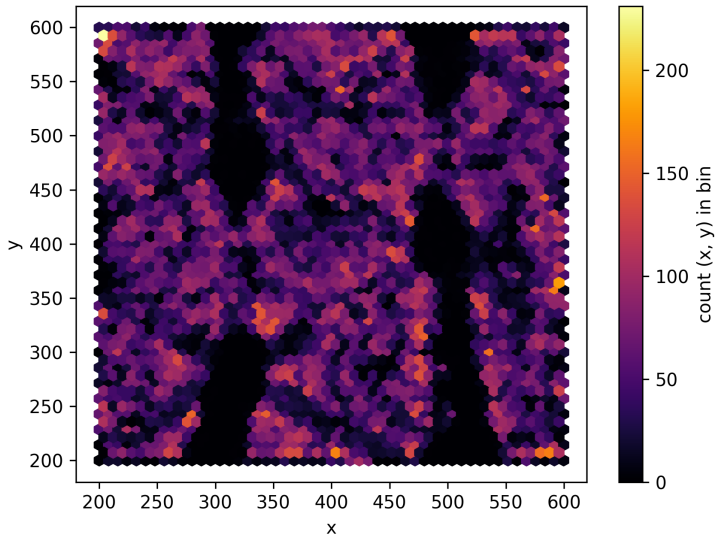

Figure 6: The distribution of ”x” and ”y” predicted features in the evaluation set.

IV. Discussion

The whole approach was designed by having as the main optimization goal the minimization of the distance. Thus, the selected preprocessing techniques and the models were chosen to bear that goal in mind. For further development, it might be interesting to approach this problem by also considering the interpretability, scalability, etc of the task. Furthermore, due to limitations in terms of computational capabilities only a small set of experiments could be conducted. However, regarding future work, doing more experiments with larger models might prove beneficial to minimize the distance further. Last but not least, in this task, domain expertise might prove extremely useful in understanding the relationships between the variables and outlier detection.

References

[1] A. S. Cornell et al. “Boosted decision trees in the era of new physics: a smuon analysis case study”. In: *Journal of High Energy Physics* 2022.4 (2022), p. 015. DOI: 10.1007/jhep04(2022)015. URL: https://doi.org/10.1007/jhep04(2022)015.

[2] *Multiple Regression*. URL: https://home.csulb.edu/~msaintg/ppa696/696regmx.htm.