CE802 - Machine Learning and Data Mining

Design and Application of a Machine Learning System for a Practical Problem: Investigation report

Registration Number: 1802655

Word count: 1498

Investigation report: comparative report P2

Data pre-processing

The dataset provided by the insurance company contains 15 features, 1500 rows of customer data and a column of True/False labels where True indicates a customer has made a claim and False indicates they haven't.

On initial inspection it was identified that half of feature F15's records were null values, which equated to 750 rows of missing data and that the features were on different scales, requiring normalisation through standard scaler[1].

Before pre-processing was applied, the dataset was split with 80% being allocated for fitting and training the models and 20% allocated for evaluating the model performance.

As Scikit-learn[2] is the program used internally by the company, the issue of null values would need to be resolved by either removing or imputing the values. Removing half of the dataset or the feature is likely to lead to a significant loss of information that would be useful for the model so imputation[3] methods have been explored instead.

The chosen models were all trained, tested and evaluated using four different imputation methods comprising of the mean, median, mode and a constant for which zero was selected.

Machine Learning Models

To avoid any human error in building the model, a pipeline[4] was created for each run that fits the selected imputation method, a standard scaler (except the decision tree) and the selected model.
In order to fine tune each model, GridsearchCV[5] was used to measure the model performance on different combinations of hyperparameters.

Decision Trees

The decision tree classifier[6] is the first model implemented in this study. The optimal parameters for each imputation method are shown in table 1.

Table 1: Decision Tree optimal parameters

	Criterion	Max_depth	Splitter
Mean	Gini	20	Best
Median	Entropy	20	Random
Mode	Entropy	20	Best
Constant (Zero)	Entropy	20	Best

As we can see from table 2 the best imputation method was to use the constant zero which gave the model a moderate set of evaluation scores.

Table 2: Decision Tree evaluation results

	Accuracy	Precision	Recall	Kappa score
Mean	0.737	0.713	0.729	0.472
Median	0.76	0.788	0.644	0.513
Mode	0.747	0.735	0.714	0.49
Constant (Zero)	0.77	0.748	0.764	0.539

The following confusion matrices[7] were generated for each imputation. As we can see the decision tree on all imputations still produces a fair number of false negatives and false positives.

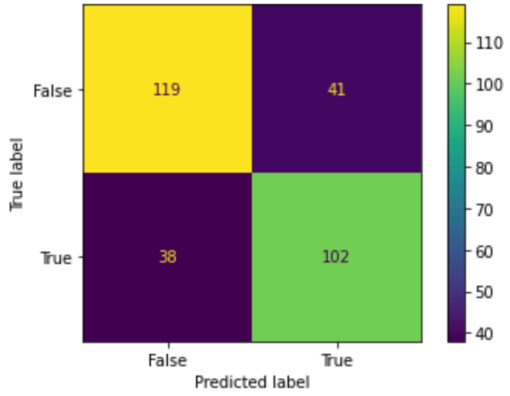

Figure 1: Decision Tree (Imputation = Mean)

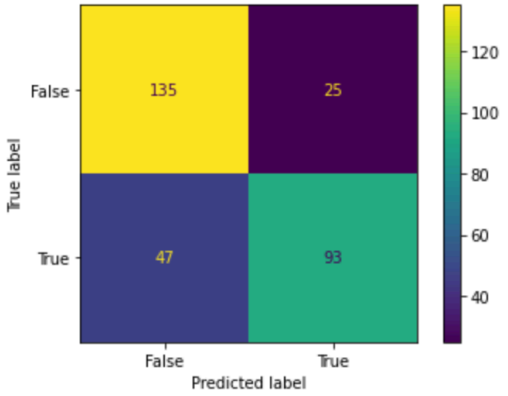

Figure 2: Decision Tree (Imputation = Median)

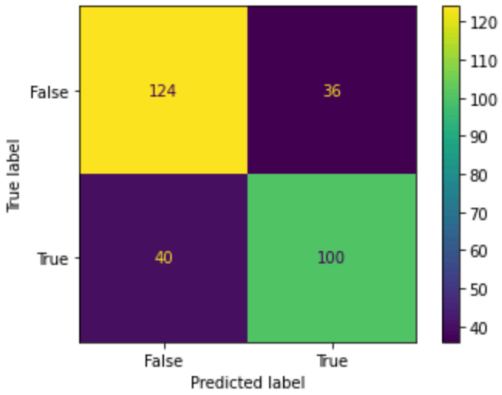

Figure 3: Decision Tree (Imputation = Mode)

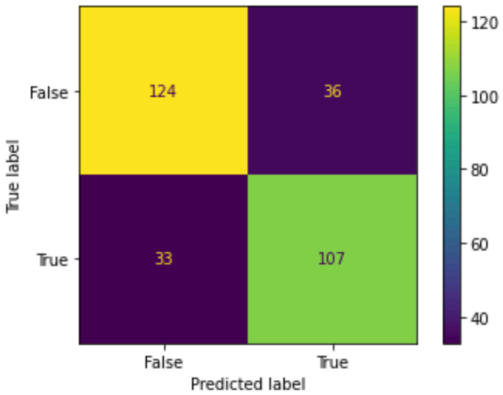

Figure 4: Decision Tree (Imputation = Constant (Zero))

K-Nearest Neighbour

K-Nearest Neighbour[8] is the second model implemented in this study. The optimal parameters for each imputation method are shown in table 3.

Table 3: KNN optimal parameters

	Weights	Metric	n_neighbours
Mean	Distance	Manhattan	100
Median	Distance	Manhattan	100
Mode	Distance	Manhattan	50
Constant (Zero)	Distance	Manhattan	50

As we can see from table 4 the median and mode produce similar accuracy[9] and kappa scores[10]. For this problem the median is the better imputation method as the company aims to give discounts to customers that don't claim so slightly better recall is preferred.

Table 4: KNN evaluation results

	Accuracy	Precision	Recall	Kappa score
Mean	0.77	0.763	0.736	0.537
Median	0.777	0.774	0.736	0.55
Mode	0.777	0.812	0.679	0.547
Constant (Zero)	0.747	0.732	0.721	0.491

Similar to decision trees, when analysing the confusion matrices, KNN has a fair number of false positives and false negatives but tends to be better at predicting true negatives.

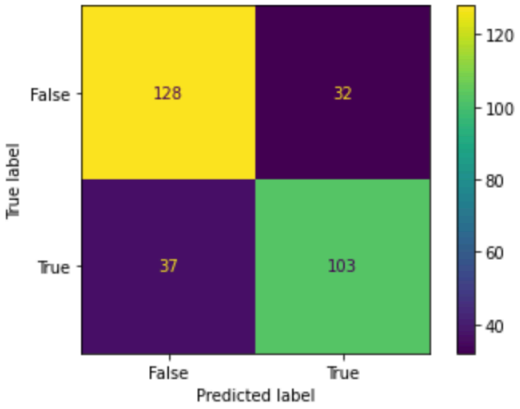

Figure 5: KNN (Imputation = Mean)

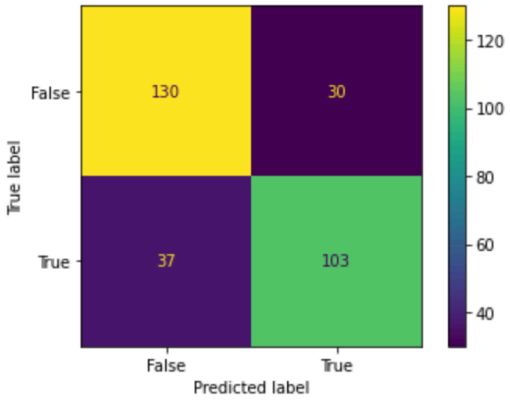

Figure 6: KNN (Imputation = Median)

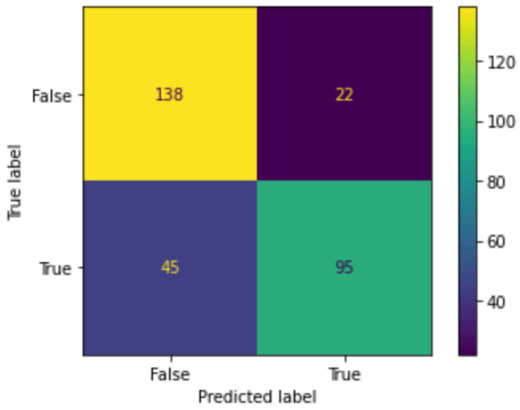

Figure 7: KNN (Imputation = Mode)

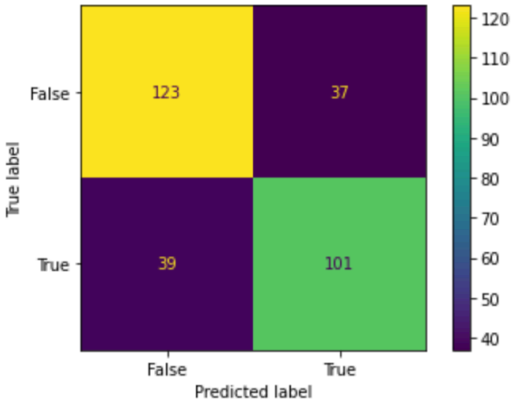

Figure 8: KNN (Imputation = Constant (Zero))

Support Vector Classification (SVC)

SVC[11] is the final model to be implemented in this study. The optimal parameters for each imputation method are shown in table 5.

Table 5: SVC optimal parameters

	Gamma	Kernel	C
Mean	Auto	rbf	100
Median	Auto	rbf	100
Mode	Auto	rbf	100
Constant (Zero)	Auto	rbf	100

As we can see from table 6 all imputation methods perform extremely well against the test set with the mode scoring slightly higher on accuracy.

Table 6: SVC evaluation results

	Accuracy	Precision	Recall	Kappa score
Mean	0.873	0.854	0.879	0.746
Median	0.873	0.854	0.879	0.746
Mode	0.877	0.865	0.871	0.752
Constant (Zero)	0.863	0.851	0.857	0.726

The confusion matrices for SVC show a fairly even split of errors, however it tends to predict slightly less false negatives.

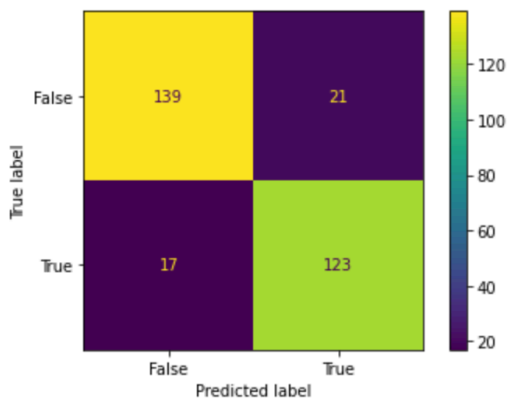

Figure 9: SVC (Imputation = Mean)

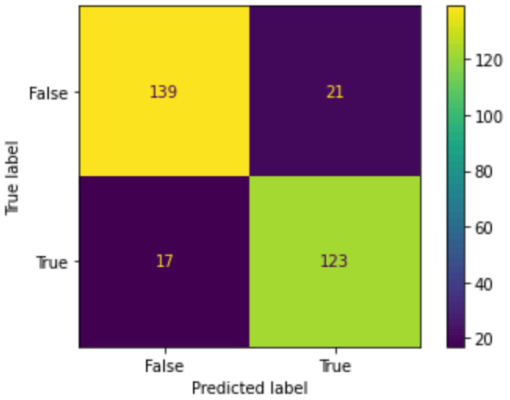

Figure 10: SVC (Imputation = Median)

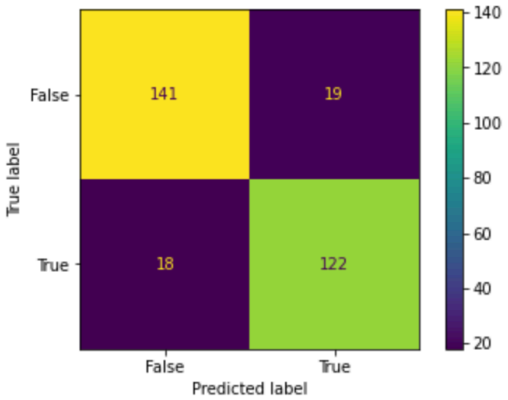

Figure 11: SVC (Imputation = Mean)

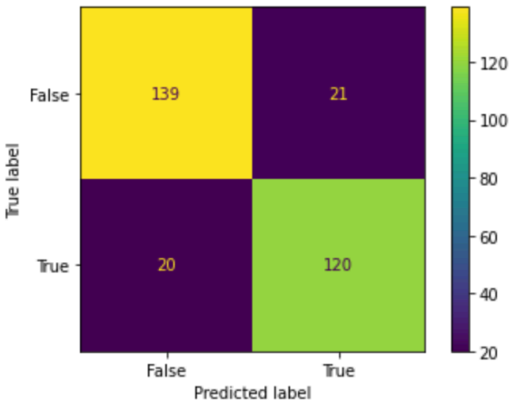

Figure 12: SVC (Imputation = Median)

Compare model performance

As we have seen from the results of evaluating the models on the test set, table 7 shows SVC outperforms Decision trees and KNN on all the evaluation metrics for this particular problem. For this reason, it will be used as the final model for the holdout test set.

Table 7: Comparison of best results from each model

	Accuracy	Precision	Recall	Kappa score
Decision Tree	0.77	0.748	0.764	0.539
KNN	0.777	0.774	0.736	0.55
SVM	0.877	0.865	0.871	0.752

Whilst no definitive boundaries are set for interpreting Kappa scores, Landis and Koch[12] identify that models within 0.41-0.60 are moderate predictors which is the range the decision tree and KNN models fell into whereas a score between 0.61-0.80 are substantial predictors which SVC scored at the higher end.

Each algorithm performs differently on each problem so the lower accuracy and kappa scores for decision tree and KNN models on this dataset may be due to a number of factors including potential outliers or the number of features in the dataset, both of which SVM handles quite well.

Investigation report: comparative report P3

This study is based on the specification of the competitor that has hired us to not only predict whether there will be an insurance claim made but also the numerical value of the claim. As the company is providing a dataset that contains the ground truth, supervised learning techniques can still be applied, however, because the output required is a numerical value this is a regression problem. The evaluation metrics used in this study are the mean squared error (MSE)[13], mean absolute error (MAE)[14] and root mean squared error (RMSE) with the RMSE being used for interpretation due to its proportionality to the data.

Pre-processing

The dataset provided consisted of 16 features, with 1500 rows of customer data comprising of a mixture of categorical and continuous values and a Target label that indicates the value the customer claimed. The features were all on different scales indicating the standard scaler would be required again to normalise the dataset.

After an initial investigation, it was found that the target values ranged from 0 to 3960.01 with a significant inflation of 447 zero values making up almost a third of the dataset as shown in figure 13.

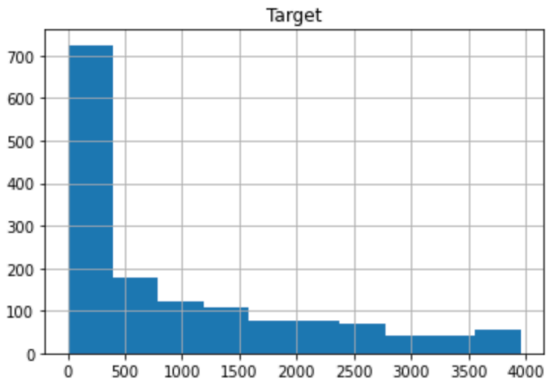

Figure 13: Histogram of Target variable

Linear Regression[15], Support Vector Regression (SVR)[16] and KNN[17] models were selected to solve this problem and initial results in table 8 show the MSE/RMSE were much higher than anticipated. The positively skewed nature of the target variable was investigated as a way to optimise model performance.

Table 8: Initial model results

	Linear Regression	SVR	KNN
Mean Squared Error	217412.33	1395810.76	431498.55
Mean Absolute Error	357.23	821.07	449.04
Root Mean Squared Error	466.27	1181.44	656.89
R Squared	0.76	-4303.99	0.12

The large amount of zero values meant a log transformation on the target variable was unable to be performed. Because the target variable ranged in the thousands it was normalised by applying a division of 1000 which was then rescaled when producing predictions.

As the dataset contained categorical features, they had to be transformed to numerical value before fitting the model. The values for the feature F15 were equally distributed, showing an ordinal relationship and to maintain that, the values were transformed into the range 0-4. The feature F4 had no obvious ordinal relationship as the values were a mixture of countries which meant one hot encoding was applied to ensure no ordinal relationship was inferred by the model.

As the initial results from the model evaluation were not as expected, table 9 shows correlation was used to identify features that had weak linear relationships with the dependent variable and filtered out. F4 was kept in to avoid any issues around the one hot encoding.

Table 9: Correlation with Target

	Target
F1	-0.000308
F2	-0.247543
F3	0.029544
F5	0.035558
F6	-0.27495
F7	-0.285668
F8	-0.020522
F9	0.338018
F10	-0.014475
F11	-0.441578
F12	0.331219
F13	0.02106
F14	0.210751
F15	0.257954
F16	0.026292
F4_Europe	-0.005123
F4_Rest	-0.237815
F4_UK	-0.038231
F4_USA	0.267662
Target	1

Machine learning models

Each model had optimal hyperparameters identified before applying the standard scaler and model through a pipeline. The results in table 10 show that when the Target variable was normalised, SVR had a significant improvement in comparison to the other models.

Table 10: Model results (Target normalised with constant)

	Linear Regression	SVR	KNN
Mean Squared Error	217412.33	119211.16	427316
Mean Absolute Error	357.23	237.5	446.02
Root Mean Squared Error	466.27	345.27	653.69
R Squared	0.76	0.9	0.63

Classification-Regression model

Due to the issue of zero inflation skewing the dataset and impacting on model performance, a two-part model was created using the Target values to classify non-zero/zero values and perform regression on the non-zero labels. The zero label results from the classifier combined with the output from the regression model gives a full set of predictions.

Three models were selected for the classification aspect of the model including decision trees, SVC and KNN. After finding the optimal parameters, table 11 shows SVC outperformed the other models on all evaluation metrics and so was selected as the classifier.

Table 11: Classification scores

	Accuracy	Precision	Recall	Kappa score
SVC	0.993	0.975	1	0.983
Decision Tree	0.85	0.69	0.769	0.624
KNN	0.89	0.785	0.795	0.715

The confusion matrix generated for SVC in figure 14 shows it only classified 2 of the labels incorrectly.

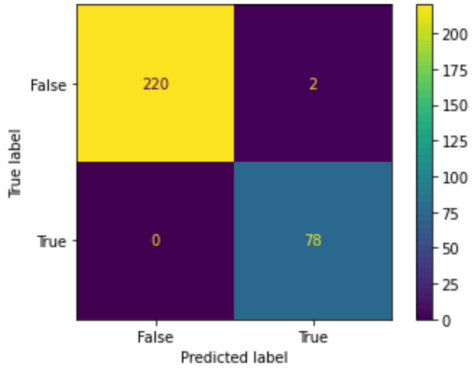

Figure 14: SVR confusion matrix

Linear regression, SVR, and KNN were selected to model the non-zero output of the classifier. Table 12 shows that SVR had significantly lower error values, an R2[18] of 0.94 indicating a very good model and was selected for the final regression model.

Table 12: Non-Zero regression scores

	SVR	Linear Regression	KNN
Mean Squared Error	63875	140508.63	291746.66
Mean Absolute Error	153.38	295.88	416.38
Root Mean Squared Error	252.74	374.84	540.14
R Squared	0.94	0.87	0.74

The results from the SVR final model with parameters (C = 10, gamma = 'auto', kernel = 'rbf') were combined with the zero values predicted by the SVC classifier and all negative values were converted to zero as negative claims cannot be achieved in a real-life scenario.
Table 13 shows the model performance after the adjustments above, indicating very low error values and an R2 score of 0.96 which outperforms all the other models in this study.

Table 13: Final model

	SVR
Mean Squared Error	45816.86
Mean Absolute Error	109.19
Root Mean Squared Error	214.05
R Squared	0.96

Comparison of models

In conclusion, the two-part classification-regression model outperforms the sole use of linear regression and SVR on all the evaluation metrics for this particular problem and will be used as the final model for the hold out test set.

The removal of the zero's in the target variable into a separate classification problem allowed the model to generalise better with the regression problem compared to the other models.

References

[1] sklearn.preprocessing.StandardScaler — scikit-learn 0.24.0 documentation [Internet]. [cited 2021 Jan 19]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

[2]. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in {P}ython. J Mach Learn Res. 2011;12:2825–30.

[3] sklearn.impute.SimpleImputer — scikit-learn 0.24.1 documentation [Internet]. [cited 2021 Jan 19]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.impute.SimpleImputer.html?highlight=simple imputer#sklearn.impute.SimpleImputer

[4] sklearn.pipeline.Pipeline — scikit-learn 0.24.0 documentation [Internet]. [cited 2021 Jan 19]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html#sklearn.pipeline.Pipeline

[5] sklearn.model_selection.GridSearchCV — scikit-learn 0.24.0 documentation [Internet]. [cited 2021 Jan 19]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

[6] sklearn.tree.DecisionTreeClassifier — scikit-learn 0.24.1 documentation [Internet]. [cited 2021 Jan 19]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier

[7] sklearn.metrics.confusion_matrix — scikit-learn 0.24.1 documentation [Internet]. [cited 2021 Jan 19]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html?highlight=confusion matrix#sklearn.metrics.confusion_matrix

[8] sklearn.neighbors.KNeighborsClassifier — scikit-learn 0.24.1 documentation [Internet]. [cited 2021 Jan 19]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html#sklearn.neighbors.KNeighborsClassifier

[9] sklearn.metrics.accuracy_score — scikit-learn 0.24.1 documentation [Internet]. [cited 2021 Jan 19]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html?highlight=accuracy#sklearn.metrics.accuracy_score

[10] sklearn.metrics.cohen_kappa_score — scikit-learn 0.24.1 documentation [Internet]. [cited 2021 Jan 19]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.cohen_kappa_score.html?highlight=kappa score#sklearn.metrics.cohen_kappa_score

[11] sklearn.svm.SVC — scikit-learn 0.24.1 documentation [Internet]. [cited 2021 Jan 19]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC

[12] Landis JR, Koch GG. The Measurement of Observer Agreement for Categorical Data. Biometrics. 1977 Mar;33(1):159.

[13] sklearn.metrics.mean_squared_error — scikit-learn 0.24.1 documentation [Internet]. [cited 2021 Jan 19]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html?highlight=mean squared#sklearn.metrics.mean_squared_error

[14] sklearn.metrics.mean_absolute_error — scikit-learn 0.24.1 documentation [Internet]. [cited 2021 Jan 19]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_absolute_error.html?highlight=absolute#sklearn.metrics.mean_absolute_error

[15] sklearn.linear_model.LinearRegression — scikit-learn 0.24.0 documentation [Internet]. [cited 2021 Jan 19]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html#sklearn.linear_model.LinearRegression

[16] sklearn.svm.SVR — scikit-learn 0.24.1 documentation [Internet]. [cited 2021 Jan 19]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html#sklearn.svm.SVR

[17] sklearn.neighbors.KNeighborsRegressor — scikit-learn 0.24.1 documentation [Internet]. [cited 2021 Jan 19]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html?highlight=neighbor#sklearn.neighbors.KNeighborsRegressor

[18] sklearn.metrics.r2_score — scikit-learn 0.24.0 documentation [Internet]. [cited 2021 Jan 19]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html