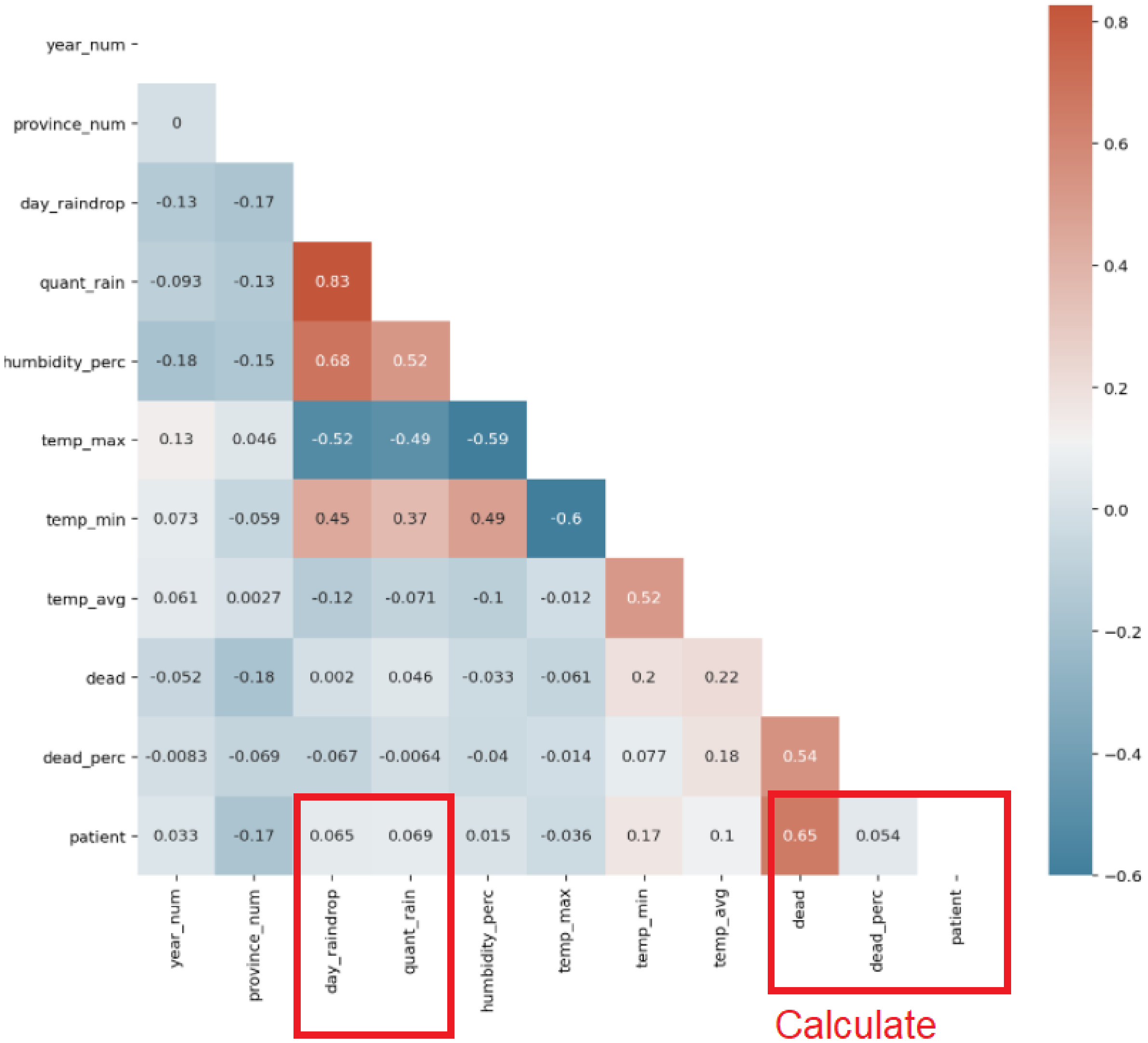
Regression Model Analysis Report

Dataset: dengue_preprocessing.csv

Features:

1. year_num: code of year 2016 – 2020 (1-5)
2. province_num: code of province 1- 77
3. day_raindrop: how many days of rain drop in 365 days
4. quant_rain: total quantity rainwater in unit of millimeters
5. humbidity_perc: average % humidity in each province, each year
6. temp_max: maximum temp in each province, each year
7. temp_min: minimum temp in each province, each year
8. temp_avg: average temp in each province, each year
9. dead: amount of dead person from dengue
10. dead_perc: percent of dead person from dengue
11. patient: amount of patient from dengue (target variable)

What we want to know? **Rain has affected to dengue patient or not?**



Calculate

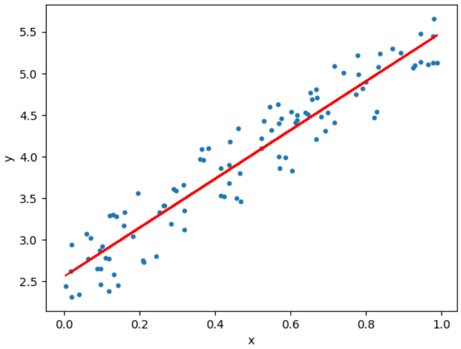
Hyperthesis

Model testing: Linear Regression Base line

We tried to test model in 5 ways

1. Multiple Linear Regression (Baseline)
2. Repeat K-folds Cross Validation
3. Features selection: Recursive Feature Elimination Method (RFE)
4. Grid Search CV
5. Polynomial regression

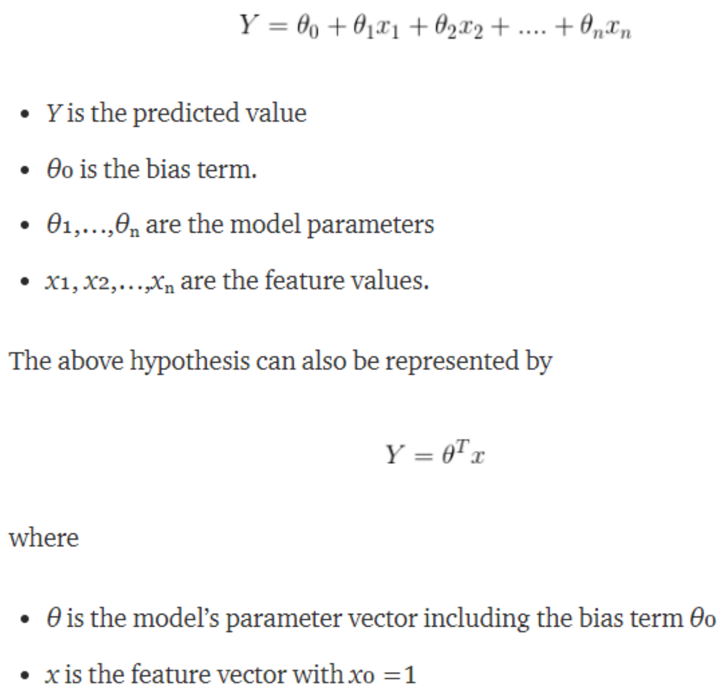
1. Multiple Linear Regression



Linear regression model is to find a relationship between one or more features (independent variables) and a continuous target variable (dependent variable). When there is only feature it is called Uni-variate Linear Regression and if there are multiple features, it is called Multiple Linear Regression.

Hypothesis of Linear Regression

The linear regression model can be represented by the following equation



$$Y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + ... + \theta_n x_n$$

- Y is the predicted value
- θ_0 is the bias term.
- $\theta_1,\ldots,\theta_n$ are the model parameters
- $x_1, x_2,\ldots,x_n$ are the feature values.

The above hypothesis can also be represented by

$$Y = \theta^T x$$

where

- θ is the model's parameter vector including the bias term θ_0
- x is the feature vector with $x_0 = 1$

Result of testing Linear Regression model

After we applied simple model linear regression to the dataset, we have an R-squared score of 0.57 in the train set and 0.49 in the test set, Root Mean Squared Error (RMSE) of 812.8884 in our prediction result on this dataset. The overall Regression Evaluation Metrics is still less than expected then we will find a new method to make it more efficient.

2. Repeats k-folds Cross Validation

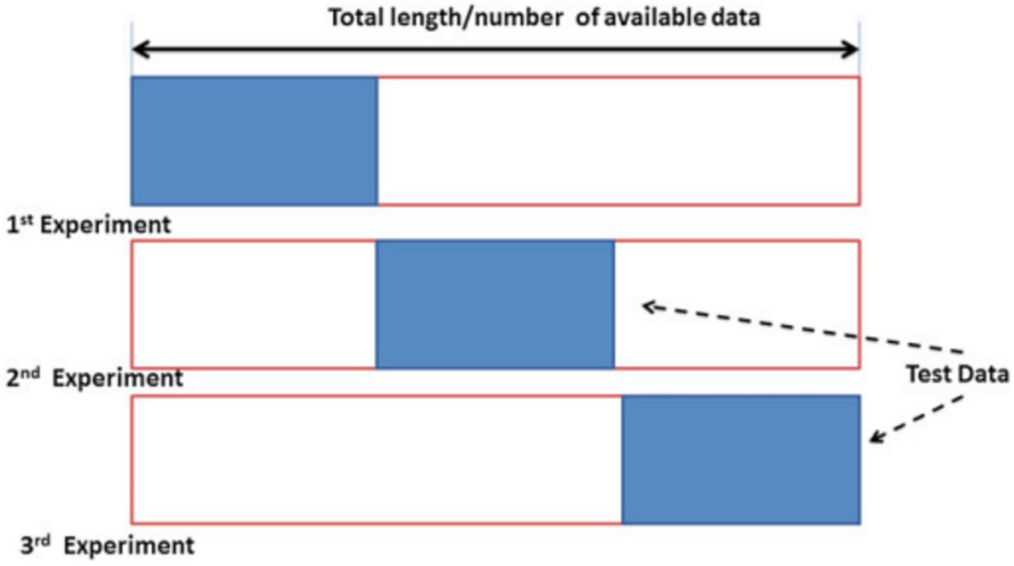


Fig. 3.8 Data splitting in *K*-fold cross-validation

The scikit-learn Python machine learning library provides an implementation of repeated k-fold cross-validation via the Repeated K-Fold class.

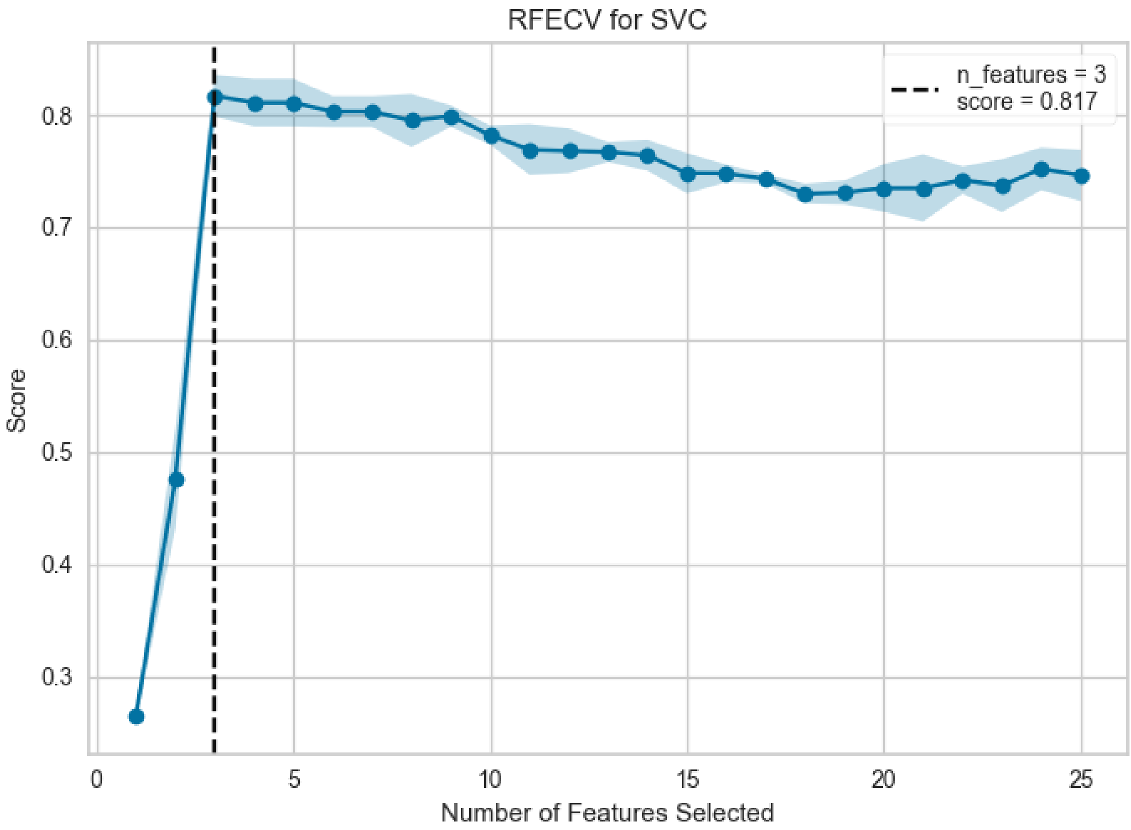
The main parameters are the number of folds (*n_splits*), which is the "*k*" in k-fold cross-validation, and the number of repeats (*n_repeats*).
A good default for k is k=10.

A good default for the number of repeats depends on how noisy the estimate of model performance is on the dataset. A value of 3, 5, or 10 repeats is probably a good start. More repeats than 10 are probably not required.

Result of testing Linear Regression model with Repeats k-folds Cross Validation.

After we applied linear regression with Repeats k-folds cross-validation by a number of folders are 5 folds and a number of repeats are 3. we have an R-squared score of 0.57 in the train set and 0.52 in the test set, Root Mean Squared Error (RMSE) of 975.68 in our prediction result on this dataset. The overall Regression Evaluation Metrics is still less than expected and RMSE also increased, so we will find a new method to make it more efficient.

3.Features selection: Recursive Feature Elimination (RFE)



Recursive feature elimination (RFE) is a feature selection method that fits a model and removes the weakest feature (or features) until the specified number of features is reached. Features are ranked by the model's coef_ or feature_importances_ attributes, and by recursively eliminating a small number of features per loop, RFE attempts to eliminate dependencies and collinearity that may exist in the model.

RFE requires a specified number of features to keep, however it is often not known in advance how many features are valid. To find the optimal number of features cross-validation is used with RFE to score different feature subsets and select the best scoring collection of features. The RFECV visualizer plots the number of features in the model along with their cross-validated test score and variability and visualizes the selected number of features.

Result of testing Linear Regression model: Recursive Feature Elimination method (RFE).

After we applied linear regression that use the Recursive Feature Elimination method (RFE). we have an R-squared score of 0.5634 in the train set and 0.5163 in the test set, Root Mean Squared Error (RMSE) of 790.60 in our prediction result on this dataset. The RFE got an optimal solution they dropped 5 columns that 1. Percent of humidity, 2. Rain quality, 3. Raindrop Day/year, 4. Year number and 5. Province number. If we select this model to deploy it will oppose to our objective of the project, so we will find a new method.

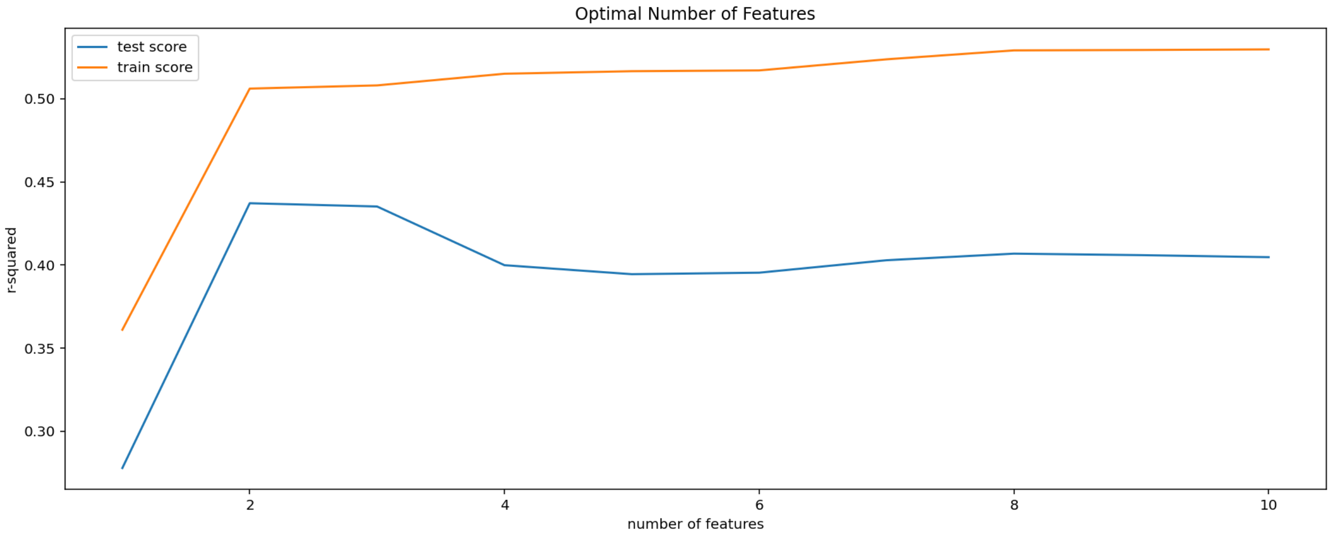
4.Grid Search CV

Grid Search CV is a library function that is a member of sklearn's model selection package. It helps to loop through predefined hyperparameters and fit your estimator (model) on your training set. So, in the end, you can select the best parameters from the listed hyperparameters.

In addition to that, you can specify the number of times for the cross-validation for each set of hyperparameters.

Then all you have to do is create an object of GridSearchCV. Here basically you need to define a few named arguments:

- **estimator**: estimator object you created
- **params_grid**: the dictionary object that holds the hyperparameters you want to try
- **scoring**: evaluation metric that you want to use, you can simply pass a valid string/ object of evaluation metric
- **cv**: number of cross-validation you have to try for each selected set of hyperparameters
- **verbose**: you can set it to 1 to get the detailed print out while you fit the data to GridSearchCV
- **n_jobs**: number of processes you wish to run in parallel for this task if it -1 it will use all available processors.



Result of testing Linear Regression model: Grid Search CV

After we applied linear regression that use the Grid Search CV and set up hyperparameters to tune:

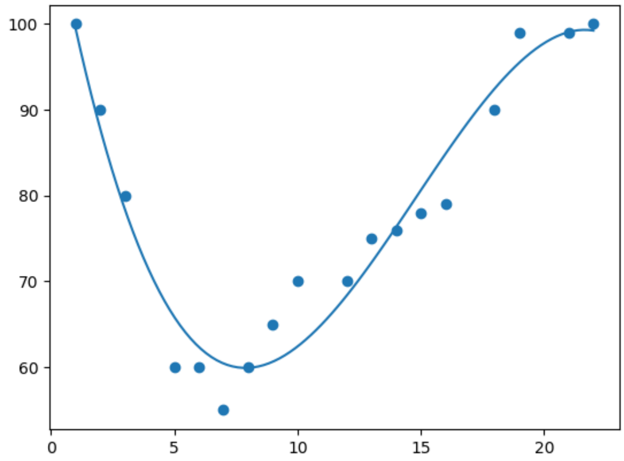
- Estimator = rfe (linear regression with backward method)
- Param_grid = hyper_params (features selection 1-10)
- Scoring= r2 (Coefficient of determination (R2 score))
- Cv = folds (use 5 folds)
- Verbose = 1 (use default value)
- Return_train_score=True

we have an R-squared score of 0.55 in the train set and 0.55 in the test set, Root Mean Squared Error (RMSE) of 864.66 in our prediction result on this dataset. The Grid Search CV got an optimal solution they dropped 8 columns which make our hypothesis of our project fail. If we select this model to deploy, it will oppose our project's objective, so we will find a new method.

5.Polynomial regression (use degree = 2)

If your data points clearly will not fit a linear regression (a straight line through all data points), it might be ideal for polynomial regression.

Polynomial regression, like linear regression, uses the relationship between the variables x and y to find the best way to draw a line through the data points.



Polynomial Regression is a form of linear regression in which the relationship between the independent variable x and dependent variable y is modelled as an nth degree polynomial. Polynomial regression fits a nonlinear relationship between the value of x and the corresponding conditional mean of y, denoted E(y |x)

Result of testing Polynomial Regression model

After we applied simple model polynomial regression to the dataset, we have an R-squared score of 0.83 in the train set and 0.78 in the test set, Root Mean Squared Error (RMSE) of 753.65 in our prediction result on this dataset. The overall regression evaluation metrics of polynomial got the best result that compare with 4 methods before.

Evaluation metrics

Evaluation metrics for regression problems as below:

Coefficient of determination (R2 score) is used to evaluate the performance of a linear regression model. It is the amount of the variation in the output dependent attribute which is predictable from the input independent variable(s). It is used to check how well-observed results are reproduced by the model, depending on the ratio of total deviation of results described by the model.

$$R^2 = 1 - \frac{\sum_i (y_i - f_i)^2}{\sum_i (y_i - \bar{y})^2}$$

Mean Absolute Error (MAE): the mean of the absolute value of the errors:

$$\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Root Mean Squared Error (RMSE): the square root of the mean of the squared errors:

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Comparison:

- **R2 Score of Training set**
- **R2 Score of Testing set**
- **MAE** is the easiest to understand because it's the average error.
- **RMSE** is even more popular than MSE, because RMSE is interpretable in the "y" units.

Conclusion

From the table as below, we will find result of Model testing in each model.

Method	Model	R2 score(Train)	R2 score(Test)	RMSE	MAE
1	Linear Regression (original)	57	49	812.88	622
2	Repeat K-folds Cross Validation	57	52	975.68	671.32
3	Recursive Feature Elimination (RFE)	56	52	790.6	605.22
4	Grid Search CV	55	55	864.66	601.58
5	Polynomial Regression	83	78	753.65	544.95

For the results in this section, we used the default settings for all machine learning approaches imported from the sklearn package.

Overall, we have experimented with various machine learning approaches in predicting dengue patients for each province that have a different environmental condition. We showed that polynomial out-perform other approaches and achieves an r-squared score greater than 0.78. In future work, we would like to improve our collecting data and add more factors that more correlate to a dependent variable for greater and reliable results in prediction.