**7 Techniques to Handle Multicollinearity that Every Data Scientist Should Know**

**Essential guide to detect and handle multicollinearity in the dataset**

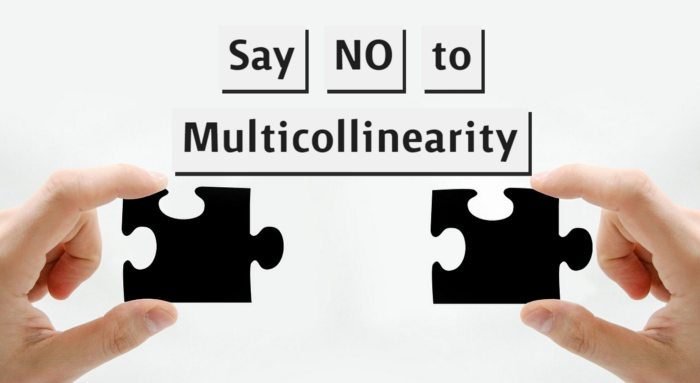


Image by [Gerd Altmann](https://pixabay.com/users/geralt-9301/?utm_source=link-attribution&amp;utm_medium=referral&amp;utm_campaign=image&amp;utm_content=523231) from [Pixabay](https://pixabay.com/?utm_source=link-attribution&amp;utm_medium=referral&amp;utm_campaign=image&amp;utm_content=523231)

Exploratory data analysis and statistical analysis are important components of a data science model development pipeline to generate insights about the data. Before fitting a machine learning model, a data scientist…

**What is Multicollinearity?**

Multicollinearity refers to the condition when two or more independent features are correlated to each other. The change in one of the collinear features may affect the other related features. Multicollinearity in the dataset may be caused due to poor designing of experiments while collecting the data or maybe introduced while creating new features.

Multicollinearity may cause to make the coefficients unstable after training a regression model. The presence of the correlated features may not add any new valuable information to the model.

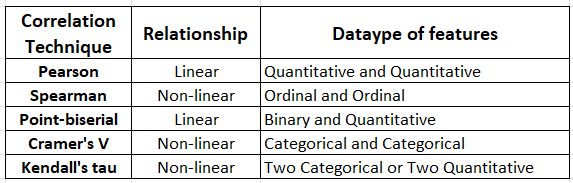
The condition of multicollinearity needs to be detected and handled prior to modeling the dataset. There are various techniques to detect and handle the condition of multicollinearity, we will discuss some of the techniques in this article.

**1.) Correlation Matrix:**

We have various statistical techniques that measure the correlation between two features. A correlation matrix can be formed for the dataset that represents correlation values between each pair of features. The correlation matrix can only measure collinearity between two features and is not able to detect multicollinearity in the dataset.

There are various correlation coefficient techniques include spearman correlation, Spearman correlation, Kendall correlation, Cramer’s V correlation, chi-square test, and many more. Pandas library provides the API to implement correlation matrix using **Dataframe.corr(method=‘pearson’)** function. The ‘pearson’ input in the method hyperparameter can be replaced with ‘kendall’, and ‘spearman’.

Now the question arises, which statistical correlation algorithm can be used for various conditions. The below-mentioned table describes which correlation technique to use in what condition:

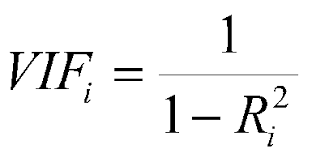


(Image by Author),

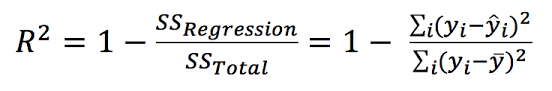
**2.) Variance Inflation Factors (VIF):**

The correlation matrix only works to detect collinearity between two features, but when it comes to detecting multicollinearity of the features, it fails. VIF is another technique that can be used to detect the multicollinearity of the features.

VIF measures the multicollinearity of the feature by computing the R-squared metric.



(Image by Author), VIF computation formula for feature ‘i’



(Image by Author), R-squared metric formulation

R-squared metrics measure how well the data points fit a line or curve. It ranges between 0 and 1, where values close to 1 reflect a good model.

The value of VIF is computed for each feature, where a regression model is trained keeping one feature as dependent variables and other features as independent features.

The VIF numerical real values range between 1 to infinity, where a value of 1 reflects no multicollinearity exists for the given feature. As the VIF value increases 1 upwards, more there exists multicollinearity between the features. Typically, a standard thumb rule says:

* VIF=1: No multicollinearity
* VIF between 1 to 5: Moderate multicollinearity
* VIF > 5: Highly multicollinear

**3.) Lasso Regression:**

Lasso regression is a linear regression technique with L1 prior as a regularize. The idea is to reduce the multicollinearity by regularization by reducing the coefficients of the feature that are multicollinear.

By increasing the alpha value for the L1 regularizer, we introduce some small bias in the estimator that breaks the correlation and reduces the variance.

Scikit-learn package offers API to perform Lasso Regression in a single line of Python code.

Refer to [scikit-learn documentation](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html) for the implementation of [Lasso Regression](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html).

**4.) Principal Component Analysis (PCA):**

PCA is a dimensionality reduction technique that uses matrix factorization under the hood to compute the eigenvalues and eigenvectors. PCA projects the given dataset into a new dimensional space based on eigenvectors.

PCA can be used to handle multicollinearity in the dataset by taking the top eigenvectors that preserve the maximum variance. The number of dimensions can be decided by observing the variance preserved for each eigenvector.

Follow [this ipynb notebook](https://github.com/bhattbhavesh91/pca-multicollinearity/blob/master/multi-collinearity-pca-notebook.ipynb) by Bhawesh Bhatt to get a better understanding of the implementation of PCA to handle multicollinearity.

**5.) Hierarchical clustering:**

Hierarchical clustering is a clustering algorithm groups similar clusters of objects based on certain similarity criteria. There are two types of hierarchical clustering algorithms:

* Agglomerative Clustering: Sequentially merges similar clusters
* Divisive Clustering: Sequentially divides dis-similar clusters

To handle multicollinearity, the idea is to perform hierarchical clustering on the spearman rank order coefficient and pick a single feature from each cluster based on a threshold. The value of the threshold can be decided by observing the dendrogram plots.

Follow [scikit-learn documentation](https://scikit-learn.org/stable/auto_examples/inspection/plot_permutation_importance_multicollinear.html#handling-multicollinear-features) for implementation of the above-discussed idea.

**6.) Multicollinearity for Categorical Feature:**

[pd.get\_dummies()](https://pandas.pydata.org/docs/reference/api/pandas.get_dummies.html), [OneHotEncoder()](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html) are function to vectorize the categorical features in the one-hot encoded format. When we one-hot encode any categorical feature, it introduces multicollinearity in the dataset.

The basic idea is to remove one feature level from the one-hot encoded vectorized data to handle the condition of multicollinearity. pd.get\_dummies() function has a parameter drop\_first that can be set to True, to remove the first feature value from vectorizing. Similarly, the scikit-learn implementation of a one-hot encoder offers a drop parameter that can remove one feature.

**7.) More Data:**

Adding more data to the existing dataset can break the pattern of multicollinearity. This technique is often useful to remove the problem of multicollinearity.

**Conclusion:**

In this article, we have discussed various techniques to handle the condition of multicollinearity. Correlation Matrix and VIF technique can detect multicollinear feature, but the data scientist needs to remove the feature deciding the threshold of coefficients. The alpha value for Lasso Regression can be tuned to decrease multicollinearity.