Data Preprocessing –

- **Importing dataset**
    1. Getting all the statistical information like mean, max, min etc, of all the columns.
    2. Checking for NA values (missing values)
- **Outliers**

Feature Selection –

- **Multicollinearity (check for independent variables)**
    1. Correlation Matrix
    2. Variance inflation factor (VIF)
    3. Drop column which has collinearity.
    4. Check again with Correlation Matrix.
    5. Check again for VIF values.
- **PCA (Principal component Analysis)**
- **Forward Selection OR Backward elimination**

Model Comparisons

Regression Models –

1. K Nearest Neighbours for regression

- Feature Scaling
- Finding appropriate value of k (neighbours)
- Training model and predicting
- Accuracy (cross validation)
- Improvements

2. Random Forest regression

- Feature Scaling
- Training model and predicting
- Accuracy (cross validation)

3. XGBoost (eXtreme Gradient Boosting)

- Training the model
- Accuracy (cross validation)
- Better Alternative (Light GBM)

4. Multiple Linear Regression – I started with this model, before doing the model comparisons. Ask me why I did not finish the implementation. :-)

CORRELATION MATRIX

Why is Multicollinearity a Potential Problem?

Multicollinearity occurs when independent variables in a regression model are correlated. This correlation is a problem because independent variables should be independent. If the degree of correlation between variables is high enough, it can cause problems when you fit the model and interpret the results.

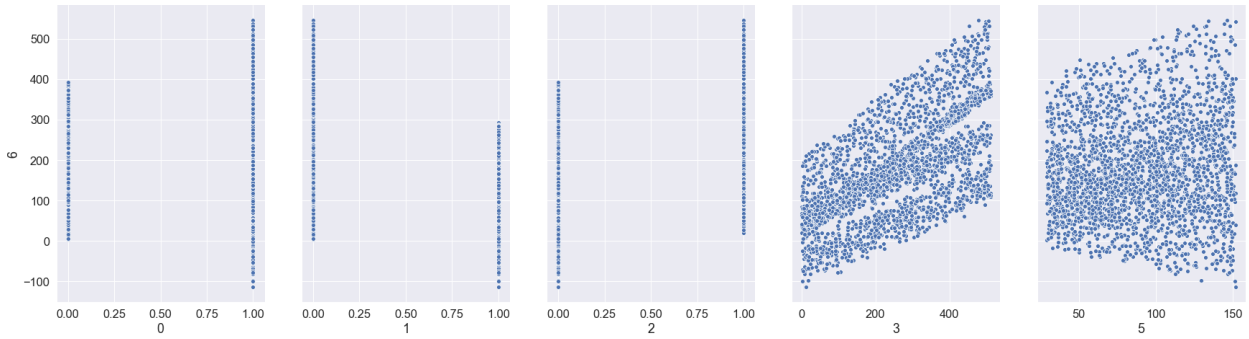
Now in the matrix of features (X), we can check for which variables are independent and which are not. One way is to find the correlation between the variables.

Why Spearman's rank correlation coefficient?

Sometimes two variables can be related in a nonlinear relationship, which can be stronger or weaker across the distribution of the variables and this is where Spearman's rank correlation coefficient comes into play. It's a non-parametric test that's used to measure the degree of association between two variables with a monotonic function, meaning an increasing or decreasing relationship.

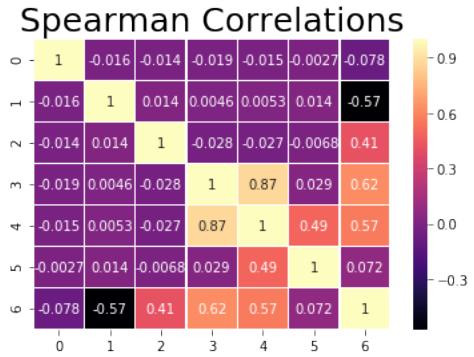
The measured strength between the variables using Spearman's correlation varies between +1 and −1, which occurs when each of the variables is a perfect monotone function of the other. It's a lot like Pearson's correlation, but whereas Pearson's correlation assesses linear relationships, Spearman's correlation assesses monotonic relationships (whether linear or not).
Spearman's coefficient is suitable for both continuous and discrete ordinal variables.
The Spearman's rank correlation test doesn't carry any assumptions about the distribution of the data.

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)}$$



We can also see in this pairplot that column 5 does not have a linear relationship with the target value so we will use **Spearman's correlation coefficient.**

Resultant correlation matrix –



Now we check for which columns are-

- Correlated to each other – We can see column 3 and 4 are highly correlated to each other.
- Correlated to target value - We also calculate which columns are highly correlated to column 6 (target column)

Results –

```
1    0.578472
3    0.627082
4    0.578512
6    1.000000
```

So now we know column 3 has highest correlation with target column, followed by column 4 and then column 1(negative). So, when we have to drop the redundant OR correlated column, it would probably be column 4, as it is less correlated to column 6 then column 3.

Principal Component Analysis

We can use PCA to reduce the dimesion of the data to any number of principal components. Before we do this , we'll need to scale our data so that each feature has a single unit variance. I won't be doing it as usually we use when our dataset has a lot of features. Here, we don't really have many.

.

VARIANCE INFLATION FACTOR (VIF)

Before we start dropping the redundant independent variables, let us check the Variance inflation factor (VIF) among the independent variables.

VIF quantifies the severity of multicollinearity in an ordinary least squares regression analysis. It provides an index that measures how much the variance (the square of the estimate's standard deviation) of an estimated regression coefficient is increased because of collinearity.

In general, we should aim for the VIF of less than 10 for the independent variables.

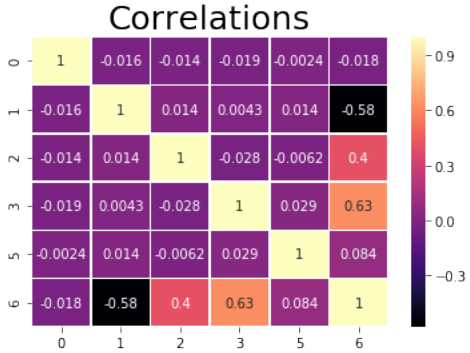
Results –

	Features	VIF_Factor
0	0	1.841500
1	1	1.894908
2	2	1.889648
3	3	200.509319
4	4	601.925706
5	5	133.743486

Drop the column '4'.

Our matrix of coefficients also explained that this column had multicollinearity, which is now proved again by VIF method.

Re-evaluate correlation matrix



This time, we don't get any high correlations between any independent variables, and so we don't really have to drop any more columns.

Re-evaluate VIF

Results –

	Features	VIF_Factor
0	0	1.809689
1	1	1.886215
2	2	1.858545
3	3	3.122326
4	5	4.020826

This shows that VIF_factor is less than 10 for all our columns so we are good here as well.

Model Comparisons

What is k-Fold Cross-Validation?

Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample. The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation.

Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model.

Compare different machine learning algorithms using K-Fold cross validation and measuring them on the basis of **negative mean squared error**.

Random Forest regression performs the best, so we will implement KNN for regression.

KNN for regression and XGB (eXtreme Gradient Boosting) perform the second best and the third best respecyively, so we will implement XGB.

Regression Models

1) **KNN regression, k = 8**

Accuracey – 98.72%
Standard deviation – 0.12%

2) **Random Forest Regression**

Accuracey – 98.65%
Standard deviation – 0.11%

2) **XGBoost**

Accuracey – 98.59%
Standard deviation – 0.13%

K Nearest Neighbours Regression

Feature Scaling (Required) -

Two ways of applying feature scaling - standardization & normalization.
We apply standardization on the X_train and X_test.

Finding appropriate value of k

- Using RMSE with different values of k, we get appropriate k value (comes out to be 8).
- GridSearchCV to find best possible k value (comes out to be 8 again).

Fitting the model and predicting.

Accuracy – calculated using cross validation.

Accuracey – **98.72%**
Standard deviation – **0.12%**

Improvements -

We can try to scale our data using different methods, and then calculate accuracy again.

Random Forest Regression

Feature Scaling (Required) -

Two ways of applying feature scaling - standardization & normalization.
We apply standardization on the X_train and X_test.

Fitting the model and predicting.

Accuracy – calculated using cross validation.

Accuracey – **98.65%**
Standard deviation – **0.13%**

Improvements -

We can try to scale our data using different methods, and then calculate accuracy again.

XGBoost (eXtreme Gradient Boosting) Regression

Feature Scaling (Required) -

Not really required for XGBoost.

Fitting the model and predicting.

Accuracy – calculated using k-fold cross validation.

Accuracey – **98.59%**
Standard deviation – **0.11%**

Improvements -

- We can try to scale our data using different methods, and then calculate accuracy again.
- We can also use Light GBM model, as it decreases the training time by a lot. Also, more useful when the datasets are super big. Here, we don't really require it.