# DEALING WITH MISSING VALUES IN DATASET

MOUSA TAYSEER JAFAR

Abstract. Missing data is defined as the values or data that is not stored (or not present) for some variables in the given dataset.Missing value can bias the results of the machine learning models and reduce the accuracy of the model.In this work we will handle the missing values by Deleting the Missing values, Replacing With Arbitrary Value (Zero), Replacing with Previous Value – Forward Fill, Replacing with Next Value – Backward Fill, Interpolation Method. Random Forest, Decision Tree, and Linear Regression have been used as Machine Learning Methods to measure the Accuracy after handling the missing values.

## Contents



*Date*: 2022-08-07.
2020 *Mathematics Subject Classification.* Artificial Intelligence.
*Key words and phrases.* Machine Learning, Missing Data, Missing Value.

## 1. Introduction

**What is Missing Data (Missing Values)?** Missing data is defined as the values or data that is not stored (or not present) for some variables in the given dataset.Missing value can bias the results of the machine learning models and reduce the accuracy of the model. The Missing Value is classified into three categories:

- Missing Completely At Random (MCAR)
- Missing At Random (MAR)
- Missing Not At Random (MNAR)

**Missing Completely At Random (MCAR):**
In MCAR, the probability of data being missing is the same for all the observations. In this case, there is no relationship between the missing data and any other values observed or unobserved (the data which is not recorded) within the given dataset. The missing values are completely independent of other data. There is no pattern. In the case of MCAR, the data could be missing due to human error, some system/equipment failure, loss of sample, or some unsatisfactory technicalities while recording the values.

**Missing At Random (MAR):**
MAR means that the reason for missing values can be explained by variables on which you have complete information as there is some relationship between the missing data and other values/data. In this case, the data is not missing for all the observations. It is missing only within sub-samples of the data and there is some pattern in the missing values.

**Missing Not At Random (MNAR):**
Missing values depend on the unobserved data. If there is some structure/pattern in missing data and other observed data can not explain it, then it is Missing Not At Random (MNAR). If the missing data does not fall under the MCAR or MAR then it can be categorized as MNAR. It can happen due to the reluctance of people in providing the required information. A specific group of people may not answer some questions in a survey.
Figure-1 show a sample of the missing data from the Tabular Playground Series - June 2022 dataset.

```
In [5]: df=pd.read_csv('data.csv')
        df.head(15)
```

Out[5]:

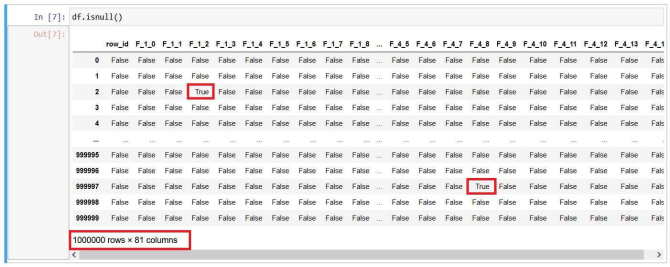
| | row_id | F_1_0 | F_1_1 | F_1_2 | F_1_3 | F_1_4 | F_1_5 | F_1_6 | F_1_7 | F_1_8 | ... | F_4_5 | F_4_6 | F_4_7 | F_4_8 | F_4_9 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | -0.354591 | -0.464038 | 2.304115 | 0.734486 | 1.696395 | 0.136285 | -0.518344 | 0.502640 | -1.852504 | ... | 3.744152 | 0.794438 | 0.265185 | -0.561809 | 0.196480 |
| 1 | 1 | 1.380940 | -0.499626 | -0.418548 | 1.911725 | -0.826130 | -1.715371 | -0.577091 | -1.041486 | 0.596067 | ... | -2.895826 | -0.738275 | 2.361818 | -0.060753 | 0.727249 |
| 2 | 2 | 0.256023 | -1.059874 | NaN | 0.345678 | 1.513814 | 1.243864 | -0.509648 | -0.800481 | -0.115945 | ... | 2.252834 | 0.472496 | 2.491386 | 0.353381 | -0.260682 |
| 3 | 3 | -0.728420 | -2.432399 | -2.453602 | -0.020509 | 0.333397 | 0.086049 | -1.787601 | 0.667011 | 0.761564 | ... | 2.004600 | -4.664806 | -0.847211 | -0.264249 | 0.664334 |
| 4 | 4 | 0.590212 | -0.066127 | 0.468009 | -1.096038 | 0.119399 | -1.809710 | 0.466358 | -0.053196 | -0.580320 | ... | 0.976937 | 2.558883 | 3.377724 | 0.846891 | 0.696032 |
| 5 | 5 | 0.088609 | NaN | 0.270489 | 0.251012 | 0.280627 | -1.705229 | 0.320000 | -0.263370 | 2.066337 | ... | -0.359761 | 1.740050 | 1.927704 | -0.082221 | -0.548425 |
| 6 | 6 | 0.533210 | 1.662285 | 0.601138 | -0.662212 | 0.033856 | -0.056208 | -1.781594 | 0.326768 | 1.583396 | ... | -2.149355 | 4.226821 | -1.136903 | 0.171289 | 0.703419 |
| 7 | 7 | -1.737332 | -1.009690 | 0.294162 | 1.002915 | 0.700551 | -1.071100 | 2.044173 | -0.545110 | -0.950346 | ... | 3.150719 | 1.134666 | 3.063516 | 0.046446 | 0.790186 |
| 8 | 8 | -0.334935 | -0.801940 | -0.754364 | -0.369206 | 0.781484 | 1.024585 | -0.394121 | -0.588350 | 0.092130 | ... | 1.075616 | 2.535868 | -0.617469 | 0.107925 | -1.063796 |
| 9 | 9 | 0.144170 | -0.580150 | -0.896780 | -0.550178 | 1.231237 | -0.474318 | -1.230894 | 0.474766 | 0.367473 | ... | -1.630001 | 1.543487 | 0.823975 | 0.036584 | -1.029416 |
| 10 | 10 | -0.357771 | 1.619492 | 0.241737 | -0.931236 | -0.439494 | 0.544001 | -0.276768 | 0.299616 | 0.031999 | ... | -1.950920 | 2.514465 | 0.133249 | -0.287813 | 0.114724 |
| 11 | 11 | 0.696625 | 0.033878 | -0.552245 | -0.508829 | 1.326049 | 1.225479 | 0.846370 | 0.576243 | -0.598348 | ... | -0.446187 | 2.841221 | 7.281518 | -0.093614 | -0.008082 |
| 12 | 12 | -1.086734 | 0.186637 | -0.226690 | -1.586084 | -0.855886 | -0.234057 | 1.269481 | 0.816826 | -0.242076 | ... | -2.315479 | 1.522740 | -1.669253 | 0.705616 | 0.633991 |
| 13 | 13 | 0.322598 | 1.323594 | 1.655550 | 0.790347 | 0.457020 | -0.536445 | 0.777910 | -0.298853 | -2.251969 | ... | 1.135202 | 1.809612 | -0.424931 | 0.224817 | -0.845311 |
| 14 | 14 | 0.745005 | 0.566602 | 0.017445 | 1.927276 | 0.285437 | 0.906739 | 0.462676 | 0.782369 | -1.389328 | ... | -3.328979 | -5.713877 | -1.104388 | 0.517917 | -0.221236 |

15 rows × 81 columns

(A) Missing vlaues

In Pandas, missing values are represented by **NaN** in the dataset as we can show in Figure A-B.


(A) in each row




(B) in all Dataset

## 2. Data Processing

The handling of missing data is very important during the preprocessing of the dataset, as many machine learning algorithms do not support missing values. Real-world data often has a lot of missing values. The cause of missing values can be data corruption or failure to record data, but there are three main problems associated with the missing values: Loss of efficiency, Complications in handling and analyzing the data. Bias results from differences between missing and complete data.

For these reasons, we will analyze the dataset with missing values carefully to understand the reasons behind the missing values, as it is crucial to find out the strategy for handling the missing values. There are many ways of handling missing values. In this work, the following method will be used to handle missing data in our dataset: Deleting the Missing values, Replacing With Arbitrary Value (Zero), Replacing with Previous Value – Forward Fill, Replacing with Next Value – Backward Fill, Interpolation Method.

- **A-Deleting the Missing values:**
  Missing values can be handled by deleting the rows or columns having null values.
- **B-Replacing With Arbitrary Value (Zero):**
  We will replace the missing values with some arbitrary value using the following code.

- **C-Replacing with Previous Value – Forward Fill:**
  The missing value is imputed using the previous value and Imputing the values with the previous value is more appropriate.
- **D-Replacing with Next Value – Backward Fill**
  In backward fill, the missing value is imputed using the next value.
- **E-Interpolation Method**
  Missing values can also be imputed using interpolation.
  Pandas interpolate method can be used to replace the missing values with different interpolation methods like 'polynomial', 'linear', 'quadratic'. Default method is 'linear'.

## 3. Data Extraction

1- We will read the dataset Tabular Playground Series - June 2022 from the CSV file [2] as shown in the figure below .

```
In [1]: import pandas as pd

In [2]: df=pd.read_csv('data.csv')
        df.head()

Out[2]:
```

| | row_id | F_1_0 | F_1_1 | F_1_2 | F_1_3 | F_1_4 | F_1_5 | F_1_6 | F_1_7 | F_1_8 | ... | F_4_5 | F_4_6 | F_4_7 | F_4_8 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | -0.354591 | -0.464038 | 2.304115 | 0.734486 | 1.696395 | 0.136285 | -0.518344 | 0.502640 | -1.852504 | ... | 3.744152 | 0.794438 | 0.265185 | -0.561809 |
| 1 | 1 | 1.380940 | -0.499626 | -0.418548 | 1.911725 | -0.826130 | -1.715371 | -0.577091 | -1.041486 | 0.596067 | ... | -2.895826 | -0.738275 | 2.361818 | -0.060753 |
| 2 | 2 | 0.256023 | -1.059874 | NaN | 0.345678 | 1.513814 | 1.243864 | -0.509648 | -0.800481 | -0.115945 | ... | 2.252834 | 0.472496 | 2.491386 | 0.353381 |
| 3 | 3 | -0.728420 | -2.432399 | -2.453602 | -0.020509 | 0.333397 | 0.086049 | -1.787601 | 0.667011 | 0.761564 | ... | 2.004600 | -4.664806 | -0.847211 | -0.264249 |
| 4 | 4 | 0.590212 | -0.066127 | 0.468009 | -1.096038 | 0.119399 | -1.809710 | 0.466358 | -0.053196 | -0.580320 | ... | 0.976937 | 2.558883 | 3.377724 | 0.846891 |

5 rows × 81 columns

```
In [3]: df.shape

Out[3]: (1000000, 81)
```

(A) dataset for Tabular Playground Series - June 2022

2- We will Check for missing values in the dataset as shown in the figures below.

```
In [8]: df.isnull().sum()

Out[8]: row_id      0
        F_1_0     18397
        F_1_1     18216
        F_1_2     18008
        F_1_3     18250
                  ...
        F_4_10    18225
        F_4_11    18119
        F_4_12    18306
        F_4_13    17995
        F_4_14    18267
        Length: 81, dtype: int64
```

(A) The Number of missing values for Each row

```
In [9]: df.isnull().sum().sum()

Out[9]: 1000000
```

(B) The Number of missing values in the dataset

3- Visualization Missing Values in the dataset:

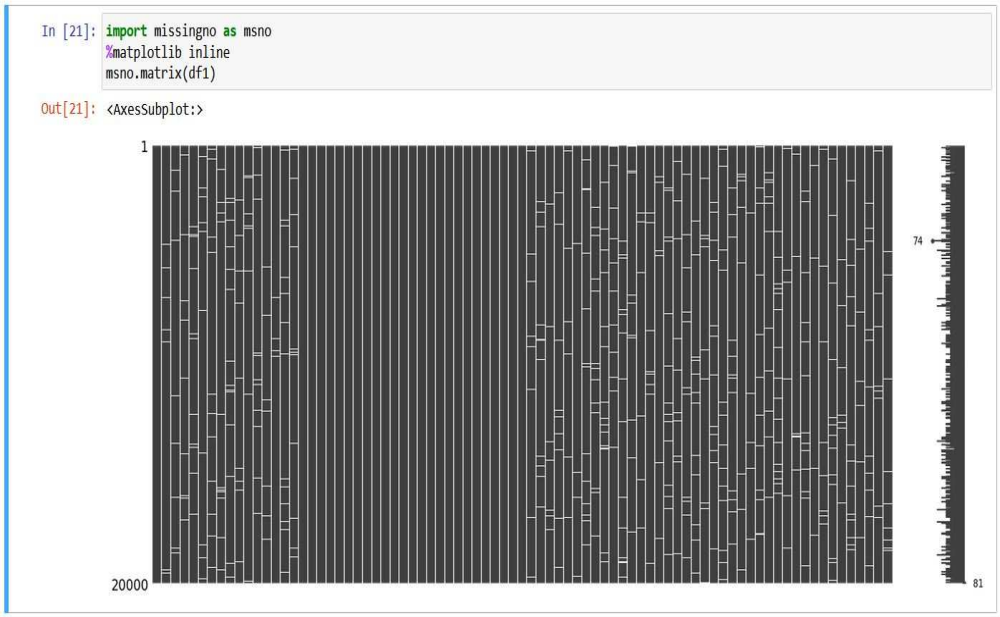



FIGURE 5. Missing Values


FIGURE 6. After Removing NAN vlaues

4- Selecting Data:
The dataset (Tabular Playground Series - June 2022) has huge data 1000000 Rows , and 81 columns.

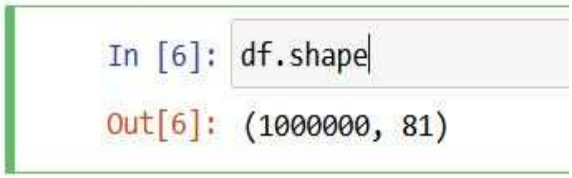



FIGURE 7. Tabular Playground Series - June 2022 dataset

We will select 5 % of dataset.

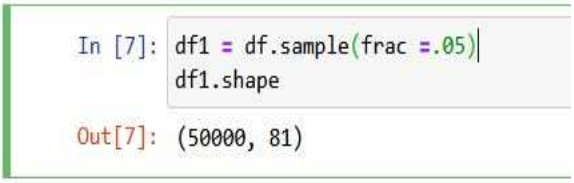



FIGURE 8. Selecting 5 % percent of data

The number of Missing Values in Selecting Data

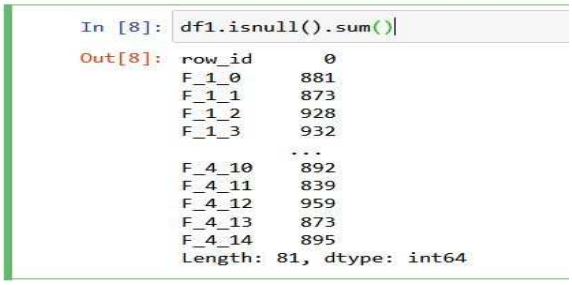



FIGURE 9. The Number of missing values for Each row

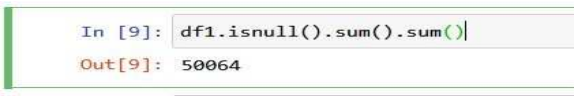



FIGURE 10. The Number of missing values in Selecting Data

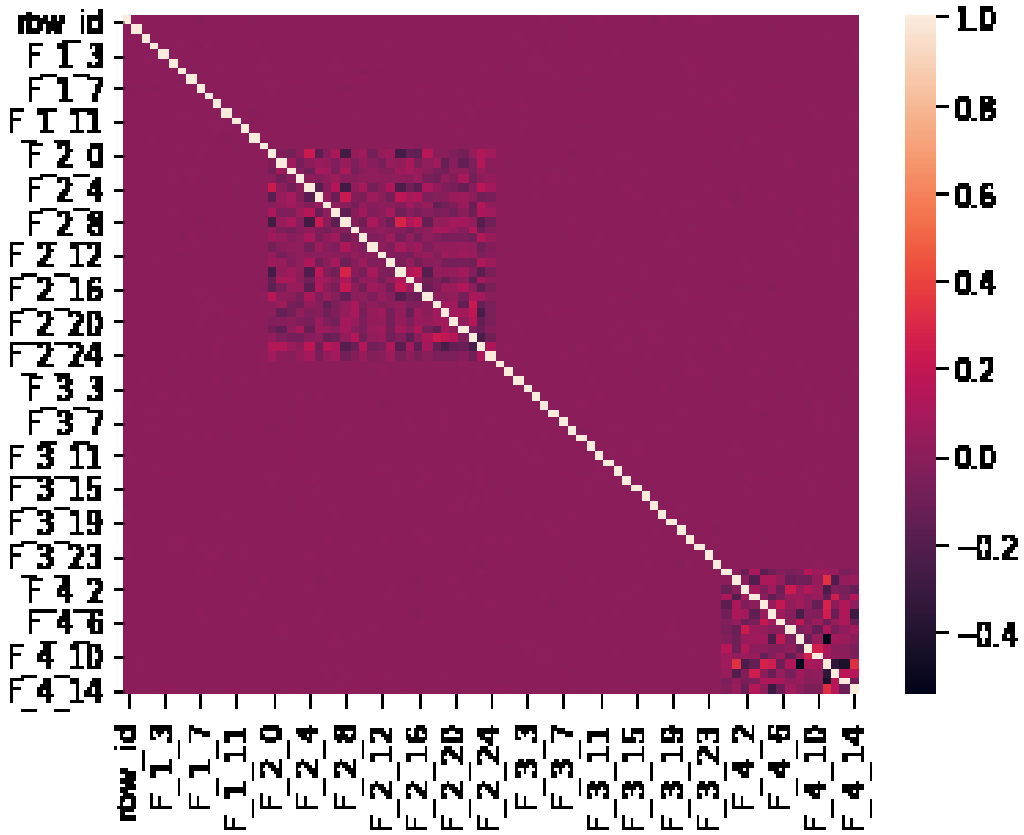



FIGURE 11. correlation data

## 4. BUILD MACHINE LEARNING MODELS

We will apply three Machine Learning Models to measure accuracy after handling the missing values.

- **Random Forest**
- **Linear Regression**
- **Decision Tree**

1- Decision Tree Model(DT)

DT is a supervised learning technique and used in different fields such as statistics, data mining, and ML [1]. it predicts responses values using learning decision rules derived from features. it can be used for decision making in both regression and classification tasks.DT consists of several components; the root node and branch node. or sub-tree, splitting, decision node, leaf or terminal node, pruning. DT is a powerful algorithm it is easy to understand, requires little data preparation, also able to handle numerical and categorical data, and deal with multiple output problems.

**Decision Tree for Interpolate Method**

```
In [62]: from sklearn.tree import DecisionTreeClassifier
         tree=DecisionTreeClassifier(criterion='entropy',random_state=0)
         tree.fit(X_train,y_train)

Out[62]: DecisionTreeClassifier(criterion='entropy', random_state=0)

In [63]: print('The accuracy of Decision Tree: ',tree.score(X_train,y_train))

         The accuracy of Decision Tree:  1.0
```

(A) Interpolate Method

**Decision Tree for Replacing with Previous Value – Forward Fill**

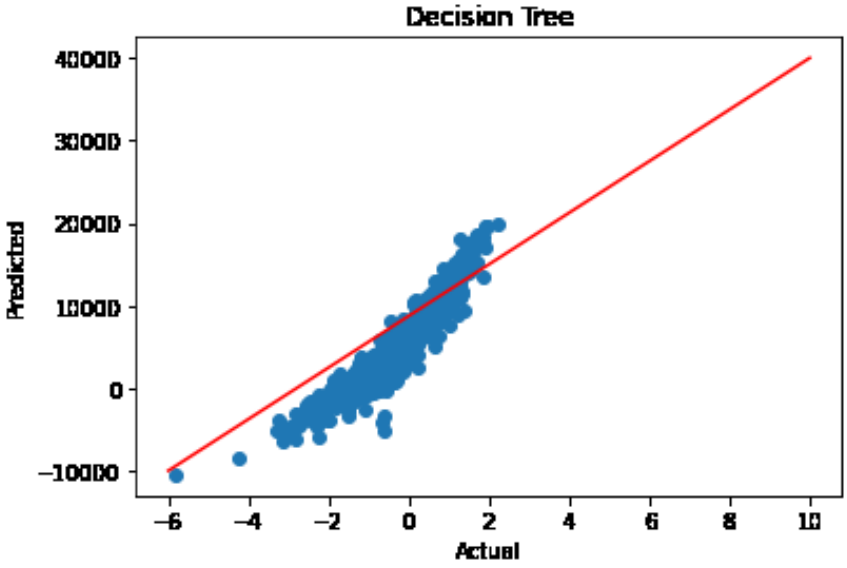
```
In [30]: from sklearn.tree import DecisionTreeClassifier
         tree=DecisionTreeClassifier(criterion='entropy',random_state=0)
         tree.fit(X_train,y_train)

Out[30]: DecisionTreeClassifier(criterion='entropy', random_state=0)

In [31]: print('The accuracy of Decision Tree: ',tree.score(X_train,y_train))

         The accuracy of Decision Tree:  1.0
```

(B) Replacing with Previous Value – Forward Fill




(C) shows the predicted result for the DT

2- Linear Regression Model(LR)
LR is another technique of ML for regression algorithms. LR finds the relationships and dependencies between variables and transforms the output using the Linear Regression function[1].

**Linear Regression for Interpolate Method**

```
In [66]: from sklearn.linear_model import LinearRegression
         Lin=LinearRegression()
         Lin.fit(X_train,y_train)

Out[66]: LinearRegression()

In [67]: print('The accuracy of LinearRegression: ',Lin.score(X_train,y_train))

         The accuracy of LinearRegression:  0.6587766661021908
```

(A) Interpolate Method

**Linear Regression for Replacing with Previous Value – Forward Fill**

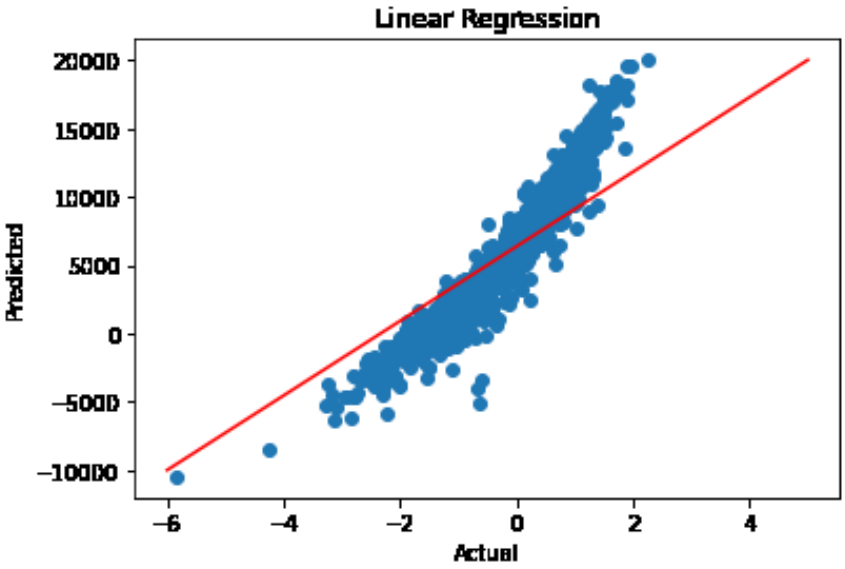
```
In [96]: from sklearn.linear_model import LinearRegression
         Lin=LinearRegression()
         Lin.fit(X_train,y_train)

Out[96]: LinearRegression()

In [97]: print('The accuracy of LinearRegression: ',Lin.score(X_train,y_train))

         The accuracy of LinearRegression:  0.6336116694335745
```

(B) Replacing with Previous Value – Forward Fill




(C) shows the predicted result for the LR

3- Random Forest Model (RF)
This algorithm is used in classification and regression problems. It is a type of a supervised classification algorithm and represents a collection of randomly selected DTs[1]. RF is considered a powerful algorithm since it has a lot of features that improve the results in the random forest such as runs efficiently on large databases, handles thousands of input variables without variable deletion, offers an experimental method for detecting variable interactions.

**Random Forest for Interpolate Method**

```
In [64]: from sklearn.ensemble import RandomForestClassifier
         forst=RandomForestClassifier(n_estimators=10,criterion='entropy',random_state=0)
         forst.fit(X_train,y_train)

Out[64]: RandomForestClassifier(criterion='entropy', n_estimators=10, random_state=0)

In [65]: print('The accuracy of Random Forest : ',forst.score(X_train,y_train))

         The accuracy of Random Forest :  0.9994375
```

(A) Interpolate Method

**Random Forest for Replacing with Previous Value – Forward Fill**

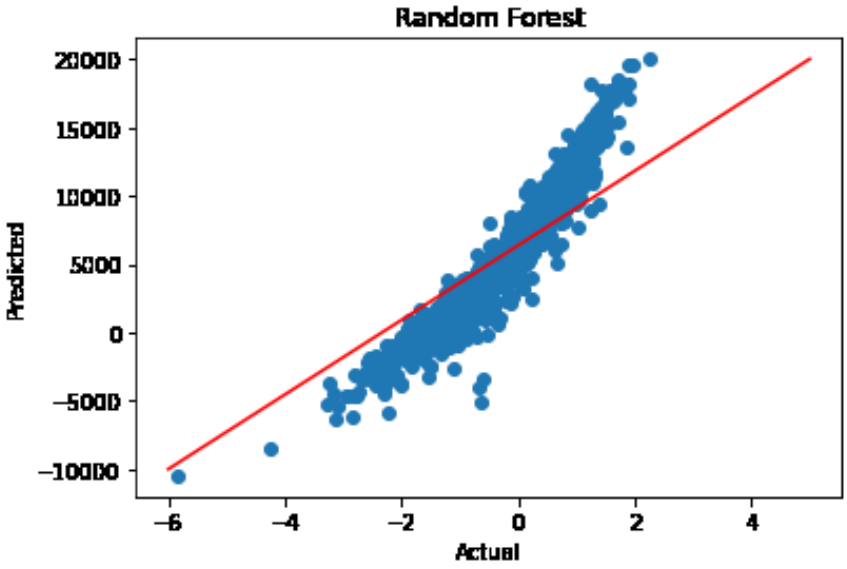
```
In [34]: from sklearn.ensemble import RandomForestClassifier
         forst=RandomForestClassifier(n_estimators=10,criterion='entropy',random_state=0)
         forst.fit(X_train,y_train)

Out[34]: RandomForestClassifier(criterion='entropy', n_estimators=10, random_state=0)

In [35]: print('The accuracy of Random Forest : ',forst.score(X_train,y_train))

         The accuracy of Random Forest :  0.9993125
```

(B) Replacing with Previous Value – Forward Fill




(C) shows the predicted result for the RF

## 5. Experiment and Analysis

Missing data is a common problem in statistical analysis.
Random Forest, Decision Tree, and Linear Regression, were used as Machine Learning Methods to measure the Accuracy after handling the missing values.

Accuracy is generated using the number of correct predictions on the test dataset to find the actual class label against the predicted class label for each category. The accuracy represents the total correct prediction overall the total prediction, as shown in equation 1.

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \tag{5.1}$$

Table 1. Precision Comparison on Event Detection Methods

| | Decision Tree | Random Forest | Linear Regression |
|---|---|---|---|
| Drop Missing Values | 1.0 | 0.9994858 | 0.9047421 |
| Missing Values Zero | 1.0 | 0.9993125 | 0.7113801 |
| Interpolate Method | 1.0 | 0.9994375 | 0.6587766 |
| Forward Fill | 1.0 | 0.9993125 | 0.6336116 |
| Backward Fill | 1.0 | 0.9993125 | 0.6207112 |

The presence of missing values in a dataset can affect the performance of a classifier constructed using that dataset as a training sample. Several methods have been proposed to treat missing data.

## 6. Conclusions

This work has introduced an approach for identifying and handling the missing values in the dataset. Several experiments have been conducted with three ML algorithms (DT, RF, and LR) to evaluate the efficiency and the performance of these approaches. All tests are experimented based on the Tabular Playground Series - June 2022 dataset. Experiments have shown that the Decision Tree Model cannot be affected by handling missing values where the accuracy for all Experiments equals 100 %.

[1] Jafar, Mousa Tayseer, et al. "Analysis and investigation of malicious DNS queries using CIRA-CIC-DoHBrw-2020 dataset." Manchester Journal of Artificial Intelligence and Applied Sciences (MJAIAS) (2021).

[2] https://www.kaggle.com/competitions/tabular-playground-series-jun-2022/data