# Sparse Matrix

A sparse matrix or sparse array is a matrix in which most of the elements are zero. On the contrary, if the most of the elements are non-zero then the matrix is considered as dense.

The number of zero valued elements divided by the total number of elements is called sparsity of the matrix.

# Sparse Features

Features with sparse data are features that have mostly zero values. This is different from features with missing data. Features with dense data have predominantly non-zero values.

# Difference between sparse data and missing data

| Missing Data | Sparse Data |
|---|---|
| Values are usually unkown | Values are known |
| Represented with NA in a dataset | Mostly represented with 0 in a dataset |
| Data has to be filled in manually or computationally | Data does not have to be filled in |

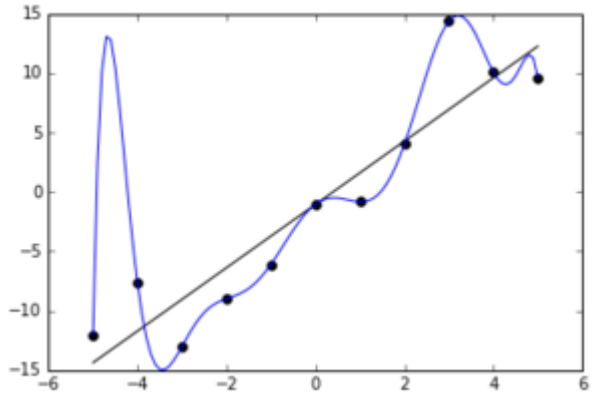
**Summary - Missing data are unknown and absent from a dataset, whereas sparse data are usually known but are rarely present.**

| Row | Feature with sparse data | Feature with missing data |
|---|---|---|
| 1 | 0 | null |
| 2 | 1 | 4 |
| 3 | 0 | 3 |
| 4 | 0 | null |

# Common problems with sparse datasets in Machine Learning

1. **Over-fitting**

If there are too many features included in the training data, then while training a model, the model with tend to follow every step of the training data, results in higher accuracy in training data and lower performance in the testing dataset.




In the above image, we can see that the model is over-fitted on the training data and tries to follow or mimic every trend of the training data. This will result in lower performance of the model on testing or unknown data.

2. **Avoiding Important Data**

Some machine-learning algorithms avoid the importance of sparse data and only tend to train and fit on the dense dataset. They do not tend to fit on sparse datasets.

The avoided sparse data can also have some training power and useful information, which the algorithm neglects. So, it is not always a better approach to deal with sparse datasets.

3. **Space Complexity**

   If the dataset has a sparse feature, it will take more space to store than dense data; hence, the space complexity will increase. Due to this, higher computational power will be needed to work with this type of data.

4. **Time Complexity**

If the dataset is sparse, then training the model will take more time to train compared to the dense dataset on the data as the size of the dataset is also higher than the dense dataset.

5. **Change in behaviour of the algorithms**

   Some of the algorithms might perform badly or low on sparse datasets. Some algorithms tend to perform badly while training them on sparse datasets. Logistic Regression is one of the algorithms which shows flawed behaviour in the best fit line while training it on a space dataset.

## Methods for dealing with sparse features

### 1. Convert the feature to dense from sparse

It is always good to have dense features in the dataset while training a machine learning model. If the dataset has sparse data, it would be a better approach to convert it to dense features.

### 2. Remove the features from the model

It is one of the easiest and quick methods for handling sparse datasets. This method includes removing some of the features from the dataset which are not so important for the model training. However, it should be noted that sometimes sparse datasets can also have some useful and important information that should not be removed from the dataset for better model training, which can cause lower performance or accuracy.

### 3. Use methods that are not affected by sparse datasets

Some of the machine learning models are robust to the sparse dataset, and the behaviour of the models is not affected by the sparse datasets. This approach can be used if there is no restriction to using these algorithms. For example, Normal K means the algorithm is affected by sparse datasets and performs badly, resulting in lower accuracy.