# Feature Engineering

## One Hot Encoding:

Feature Engineering - It refers to manipulation such as addition, deletion, Combination, mutation of your dataset to improve machine learning model training, leading to better performance and greater Accuracy.

## One-Hot Encoding:

One hot encoding is a technique that we use to represent Categorical variables as numerical values in machine learning model.

### Advantages:

→ Straightforward to implement.

→ Does not require hours of variable exploration.

→ Does not expand massively the feature space.

[No of columns in dataset]

### Disadvantages:

→ Does not add any information that may make the variable more predictive.

→ Does not keep the information of the ignored labels.

Suppose, we have the "Gender" feature which includes male and female. We need to change this Category feature to number or numerical values. It can be represented as 0 (or) 1. 0 for male and 1 for female.

Suppose we have the dataset,

| | Employee-ID | Gender | Remarks |
|---|---|---|---|
| 0 | 45 | Male | Nice |
| 1 | 78 | Female | Good |
| 2 | 56 | Female | Great |
| 3 | 12 | Male | Great |
| 4 | 7 | Female | Nice |

After applying One hot encoding,

It looks,

| | Employee_ID | Remarks_Good | Remarks_Great | Remarks_Nic |
|---|---|---|---|---|
| 0 | 45 | 0 | 0 | 1 |
| 1 | 78 | 1 | 0 | 0 |
| - | - | - | - | - |

| Gender_female | Gender_male |
|---|---|
| 0 | 1 |
| 1 | 0 |
| - | - |

# One Hot Encoding - Variables with many Changes

```
import pandas as pd
import numpy as np
data = pd.read_csv('mercedesbenz.csv', usecols = ['x1', 'x2', 'x3',
                   'x4', 'x5', 'x6'])
data.head()
# To find out Unique Categories in each and every column
for col in data.columns:
    print(col, ':', len(data[col].unique()), 'labels')
# Performing One-hot encoding
pd.get_dummies(data, drop_first = True).shape
```

Out: (4209, 117)

↓

117 Columns [It leads to Curse of dimensionality]

According to this, we use to take top 10 features from the each feature. (It's Based on KDD cup in kaggle)

# Selecting top 10

# One-hot encoding perform top 10 features and remaining are zero.

# Let's find the top 10 most frequent categories for the variable $x_2$

```
data.x2.value_counts().sort_values(ascending=False).head(20)

top_10 = [x for x in data.x2.value_counts().sort_values(ascending=False).head(10).index]

top_10
```

# Now make binary variable (0 and 1) for encoding

```
for label in top_10:
    data[label] = np.where(data['x2']==label,1,
```

```
dat[['x2'] + top_10].head(40)
```

# perform one hot encoding for top 10

```
def One_hot_top_x (df, variable, top_x_labels):

    for label in top_x_labels:

        df [variable + '_' + label] = np.where
                                    (data [variables] == label, 1, 0)

# read data again fresh

data = pd.read_csv ('mercedesbenz.csv', usecols =
                                    ['x1', 'x2', 'x3', 'x4', 'x5', 'x6'])

One_hot_top_x (data, 'x2', top_10)

data.head()
```

↑

Similar applying for $x_1$, $x_2$, $x_3$ upto x6. Finally removes all the $x_1$, $x_2$, $x_3$ upto x6 columns.

One-hot Encoding wells for noise data because take most frequent elements and make better prediction.

## Types of Encoding Techniques:

Whenever speak about the encoding techniques, Firstly comes to the mind is Categorical Variable

Because, we need to change Categorical Value to Numerical like, Gender [male, female] like this.

Types,

1. Nominal encoding.

[order does not matter]

| Gender | State |
|---|---|
| Male | Nayework |
| Female | Spain |
| Male | Nayework |

This type of data is known as the Nominal encoding.

2. Ordinal encoding.

[order does matter]

Education dataset,

| | |
|---|---|
| X | BE |
| Y | BCom |

Z  Phd

A  BE

Order does, The rearrangement preference based on order.

Phd - 1

BE - 2

BCom - 3

Nominal Encoding

↳ One hot Encoding.

↳ One hot Encoding with many Categorical.

↳ Mean Encoding.

Ordinal Encoding

↳ Label Encoding.

↳ Target guided Ordinal Encoding.

---

One hot Encoding.

We already discussed the one hot Encoding,

Suppose,

| State | Geno |
|---|---|
| Germany | |
| France | |
| Spain | |

Normally, If applied one hot encoding. (Any of the One Column) may be deleted.

After apply one hot encoding,

| State | Germany | France | Spain |
|---|---|---|---|
| Germany | 1 | 0 | 0 |
| France | 0 | 1 | 0 |
| Spain | 0 | 0 | 1 |

The Spain column removed, after that

1 0 → Represent Germany.

0 1 → Represent France.

0 0 → Represent spain.

Suppose, Consider another feature [Pincode]

Pincode

600001

600002

600004

600008

⋮

600124

(Suppose 100 pincodes)

100 pincodes,

The one hot encoding creates 99 features.

⇓

It leads to Curse of dimensionality.

⇓

So, Some other techniques Used.

## Label Encoding:

Label encoding technique used for Ordinal Categories.

Example,

| Education | Rank |
|---|---|
| BE | 1 |
| ME | 3 |
| Phd | 4 |
| STATS | 2 |

We now put the values, based on the Rank. [It represents the higher most important].

This is known as Label Encoding.

---

## One hot Encoding with multiple Categories:

Suppose the feature, $f_1$ have 50 Categories.

The Kaggle Competition, KDD Cup Orange proves the [select top 10 Categories and apply One hot encoding] is efficient.

This also proves with Researchers.

So, top 10 => Applying one hot encoding => Then it create the 9 columns.

## Target Guided Ordinal Categories: (Encoding)

Suppose we have,

| $f_1$ | o/p | | mean | Rank |
|---|---|---|---|---|
| A | 1 | Compare this ↔ | $1/2 = 0.5$ | 2 |
| B | 1 | ↔ | $1/3 = 0.33$ | 3 |
| C | 0 | | $1/2 = 0.5$ | 2 |
| D | 1 | | $1/1 = 1$ | 1 |
| B | 0 | | $1/3 = 0.33$ | 3 |
| A | 0 | | $1/2 = 0.5$ | 2 |
| C | 1 | | $1/2 = 0.5$ | 2 |
| B | 0 | | $1/3 = 0.33$ | 3 |

⇒

The weighted mean also Calculate with various methods in target encoding.

---

## Mean Encoding:

| $f_1$ | o/p | Mean (Suppose) | |
|---|---|---|---|
| A | 1 | 0.73 | this value is fed into machine learning training. |
| B | 0 | 0.6 | |
| C | 1 | 0.5 | |
| D | 1 | 0.1 | |
| A | 0 | – | |
| B | 1 | – | |

Why use these techniques ?

Suppose, we have pincode feature. It contains 1600 categories. The one hot performing is very large dimensionality. It leads to Curse of dimensionality

So rather used, Mean Encoding.

Based on the dataset, we use different different encoding techniques.

Pincodes:

60001
60002
60003

} mean ⇒ to machine.

FEATURE SCALING:

Why feature Scaling ?

Suppose we have the,

| Height | weight | BMI |
|---|---|---|
| 180 | 78 | |
| 170 | 84 | |

Every features have different different units and magnitude.

Without feature Scaling,

Some algorithm is does not work properly.

Example:

k-nearest neighbour.

Suppose plot the points in this algorithm, there is a huge difference between the datapoints and very long distance points covered.

Some Scaling techniques are,

→ Standard Scaler.

→ Minmax scaler etc...

When apply feature Scaling?

Some algorithm like,

→ Linear Regression for [Gradient descent]

→ Eucleidan distance related algorithm like KNN, k-means [Distance problem avoid]

→ CNN [unit scaling for images] [RGB 0-255

When does not require feature Scaling?

→ Decision Tree algorithms.

→ Random forest.

→ xgboost.

Because, it is tree based decision algorithm. Whenever apply the Scaling or not. Both the Scores are well-performed.

## HANDLE MISSING VALUES IN CATEGORICAL VARIABLE:

There are four steps(or) four techniques are available,

→ Delete the Rows.

→ Replace the Most frequent values.

→ Apply Classification ~~or~~ classifier algorithm to predict.

→ Apply Unsupervised Machine learning algorithm.

Consider,

| $f_1$ | $f_2$ | $f_3$ | O/p |
|---|---|---|---|
| MALE | 23 | 24 | Yes |
| | 24 | 25 | No |
| FEMALE | 25 | 26 | Yes |
| MALE | 26 | 27 | Yes |

Delete the Rows,

→ The very less no of row (missing) in the huge dataset.

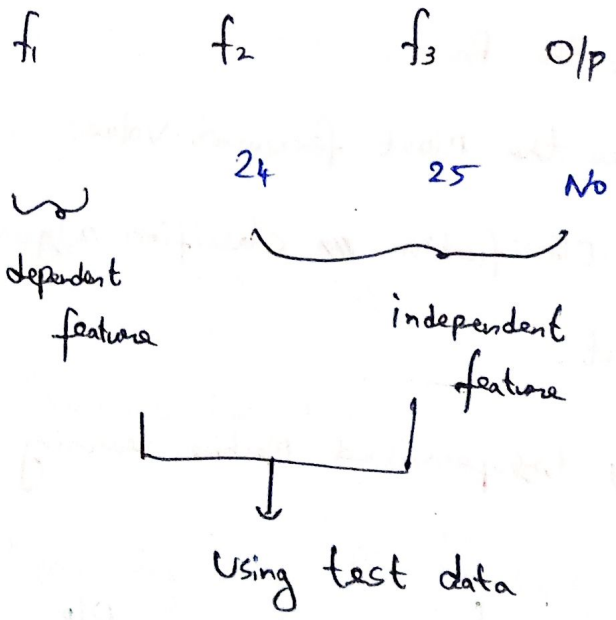
→ Delete the entire row.

Replace with most frequent value,

To fill with most frequent value using "MODE" method in Statistics.

Apply Classifier algorithm,

Take the row,




Remaining things are to be considered as the Training data to predict the outcome.

Apply Unsupervised ML:

Using Clustering technique to group the feature..

| $f_2$ | $f_3$ | |
|---|---|---|
| 24 | 25 | → which groups are related and assign them to predict. |

Using Unsupervised ML algorithm. kmeans, kNN etc.

---

HANDLING CATEGORICAL FEATURES FOR MANY CATEGORIES:

COUNT/ FREQUENCY ENCODING:

Higher No of labels (or) Higher No of Categories is known as High Cardinality.

Definition:

The frequency encoding is replace each label of the Categorical variable by the count, this is the amount of times each label appears in the dataset.

Example:

```
import pandas as pd

import numpy as np


df = pd.read_csv('mercedesbenz.csv', usecols=['x1', 'x2'])

df.head()


df.shape

Out: (4209, 2)

# Applying one hot encoding

pd.get_dummies(df).shape

out: (4209, 71)  # more columns created.
# It tends to may be Curse of dimensionality.

#Applying frequency label encoding

len(df['x1'].unique())

len(df['x2'].unique())

# look how many labels
```

```
for col in df.columns[0:]:
    print(col, ':', len(df[col].unique()), 'labels')
```

Out:

X1 : 27 labels

X2 : 44 labels

# Take X2 because it is high Cardinality (or) take any.

# Count X2 each unique category value count

```
df.X2.value_counts().to_dict()
```

# To dictionary purpose is the need to mapping variable.

```
df_frequency_map = df.X2.value_counts().to_dict()
# Now replace the X2 values in the dataset
df.X2 = df.X2.map(df_frequency_map)
df.head()
```

Before applying:

| $X_1$ | $X_2$ |
|---|---|
| v | at |
| t | av |
| w | n |
| t | n |
| v | n |

After Applying:

| $X_1$ | $X_2$ |
|---|---|
| v | 6 |
| t | 4 |
| w | 137 |
| t | 137 |
| v | 137 |

Advantages:

→ It is Very Simple to implement.

→ Does not increase the feature dimensional space. [Avoids leads to Curse of dimensionality]

Disadvantages:

→ If some of labels have the same Count, then they replaced with same Count and they will loose some Valuable information.

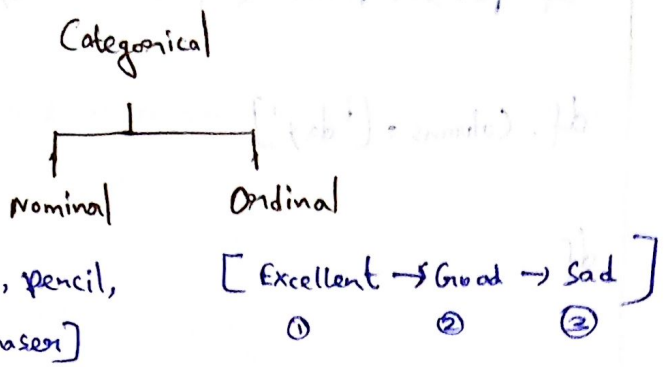
→ Adds somewhat arbitrary numbers, and therefore weights to the different labels, that may not be related to their predictive power.

---

HANDLING ORDINAL CATEGORIES [ORDINAL ENCODING]

Ordinal Categorical Variable:

Ordinal data is a categorical, statistical data type where the variables have natural, ordered categories and the distances between the categories is not known.




Example: [ORDINAL]

→ Student's grade in Exam. [A, B, C or Fail]

→ Educational level [Elementary school, high school, college, graduate, Phd] from rank 1 to 4.

IMPLEMENTATION:-

```
import pandas as pd
import datetime

# Create a variable with dates, and from that extract
  the weekday.
# I create a list of dates with 20 days difference
  from today.
# Then transform into dataframe -

df_base = datetime.datetime.today()
df_date_list = [df_base - datetime.timedelta(days = x)
                for x in range(0, 20)]
```

```
df= pd. DataFrame (df_data_list)

df. Columns = ['day']

df

# Extract the week day name

df ['day_of_week'] = df ['day'] . dt. weekday_name

df.head()

# Categorical variable by ordinal number replacement

weekday_map = { "monday": 1,
                "Tuesday": 2,
                "wednesday": 3,
                "Thursday": 4,
                "friday": 5,
                "Saturday": 6,
                "Sunday": 7 }

df ['day_Ordinal'] = df. day_of_week. map (weekday_map)

df. head (20)
```

Advantages:

Ordinal measurements is normally used for surveys and Questionnaire.

With Respect to machine learning.

→ Keeps the semantical information of the variable.

→ Straightforward approach.

Disadvantages:

With Respect to machine learning,

→ Does not add machine learning valuable information.