THE iSCALE

Industries Helping Hands

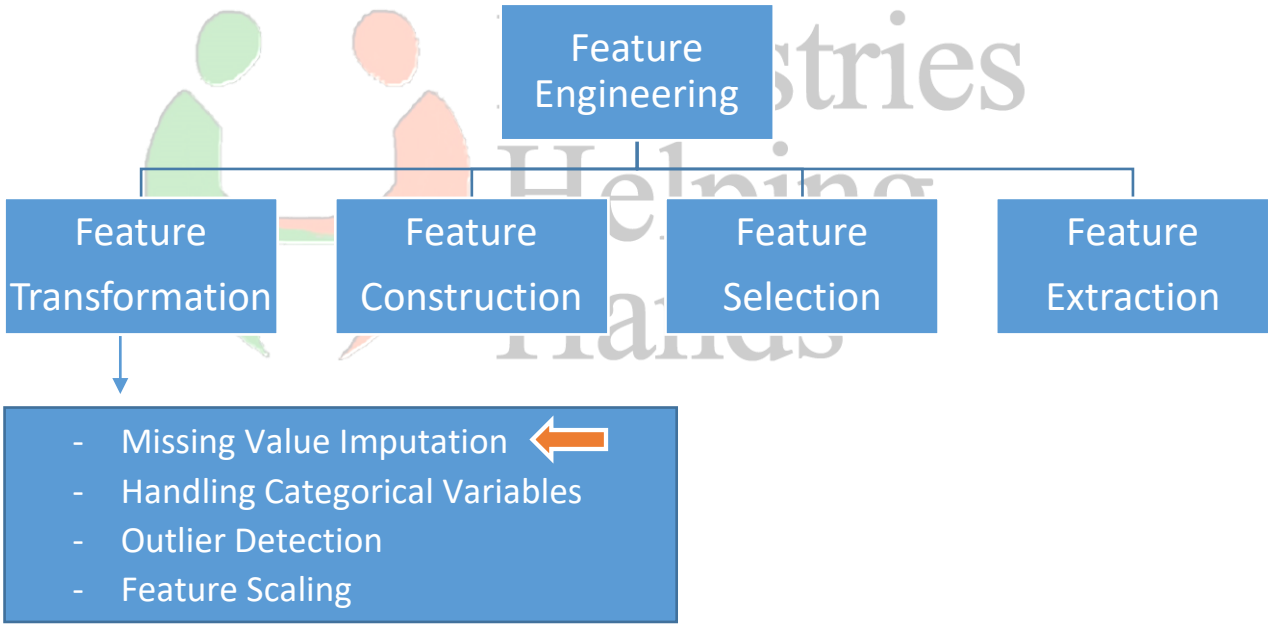
Data Science | 30 Days of Machine Learning | Day - 12

Educator Name: Nishant Dhote
Support Team: **+91-7880-113-112**

----Today Topics | Day 12----

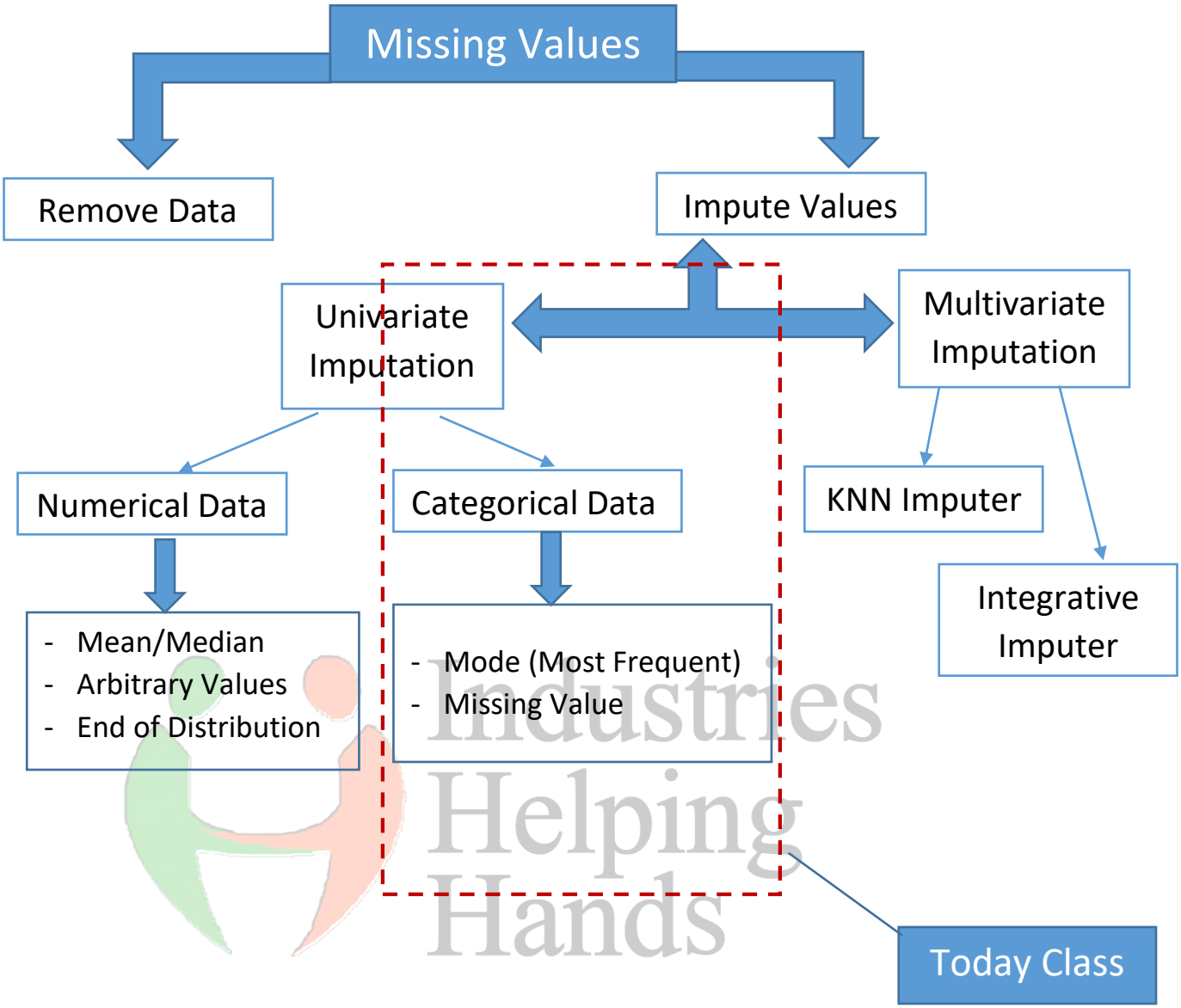
- What is the purpose of using “Mode”?
- What is most frequent imputation?
- Which variables can be imputed with most frequent / mode Imputation?
- When to use mode / most frequent category imputation?
- Missing value imputation?

Dataset Link GitHub: https://github.com/TheiScale/30_Days_Machine_Learning/



- Two Important ways to handle missing values in the dataset:

1. Deleting rows with missing values (Remove Missing Values)
2. Impute missing values (Fill)
   - Univariate (Numerical & Categorical Removal)
   - Multivariate (KNN & Iterative Imputer)

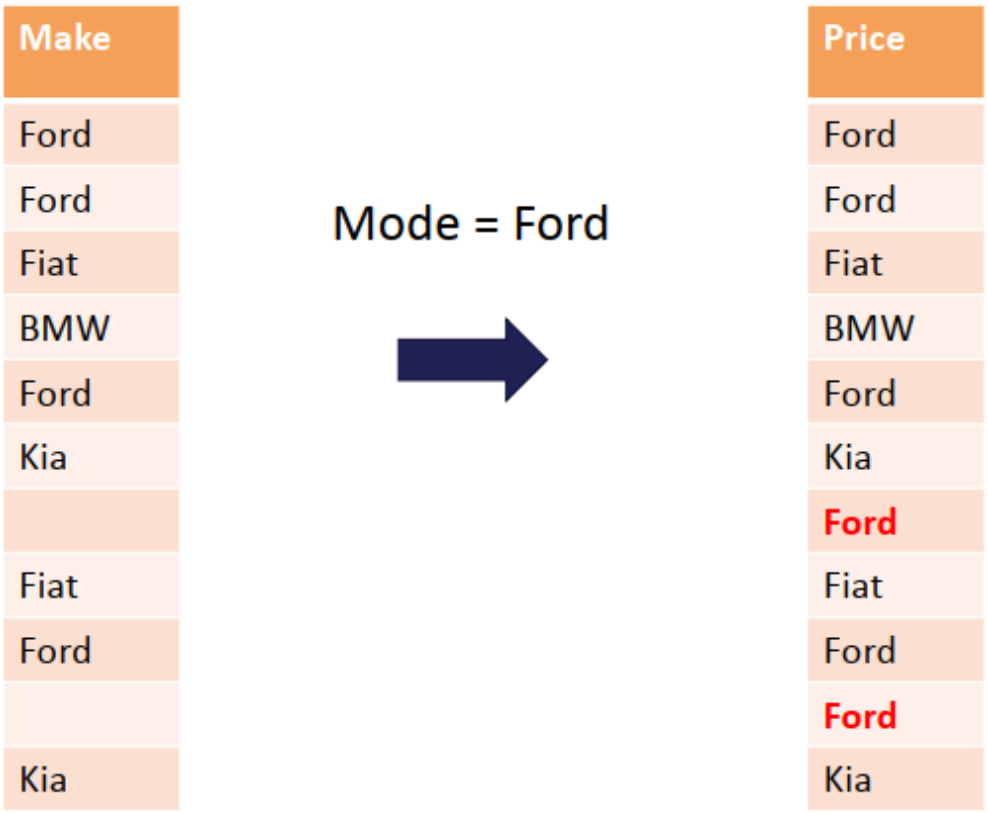


What is the purpose of using "Mode"?

The mode is a statistical measure in machine learning that represents the most frequently occurring value in a dataset. It can be used to identify the central tendency of categorical data.

What is most frequent imputation?

Mode imputation consists of replacing all occurrences of missing values (NA) within a variable by the mode, which in other words refers to the most frequent value or most frequent category.



Which variables can be imputed with most frequent / mode Imputation?

Although the mode, or most frequent value can be calculated for both numerical and categorical variables, in practice, we only use this technique on categorical variables. The reason is that for numerical variables, the mean or the median tend to better represent the average value of the population.

When to use mode / most frequent category imputation?

Data is missing completely at random.
No more than 5% of the variable contains missing data.

House Price Kaggle Datasets: https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data

<Start Coding | Mode - imputation>

#Import Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

#Import Dataset

```
df =
pd.read_csv('train.csv',usecols=['GarageQual','Fir
eplaceQu','SalePrice'])
```

```
----
df.head()
```

#Check missing (null) value

```
df.isnull().mean()*100
```

#Plot Bar Garage Value

```
df['GarageQual'].value_counts().plot(kind='bar')
```

```
----
df['GarageQual'].mode()
```

#kde Plot | Compare Houses with TA | Null

```
fig = plt.figure()
ax = fig.add_subplot(111)

df[df['GarageQual']=='TA']['SalePrice'].plot(kind
='kde', ax=ax)

df[df['GarageQual'].isnull()]['SalePrice'].plot(k
ind='kde', ax=ax, color='red')

lines, labels = ax.get_legend_handles_labels()
labels = ['Houses with TA', 'Houses with NA']
ax.legend(lines, labels, loc='best')

plt.title('GarageQual')
```


#Store variable TA in temp

```
temp = df[df['GarageQual']=='TA']['SalePrice']
```

#Replace missing value with TA

```
df['GarageQual'].fillna('TA', inplace=True)
```

#Review Bar Plot Changes

```
df['GarageQual'].value_counts().plot(kind='bar')
```

#Draw plot again | After Imputation

```
fig = plt.figure()
ax = fig.add_subplot(111)


temp.plot(kind='kde', ax=ax)

# distribution of the variable after imputation
df[df['GarageQual'] ==
'TA']['SalePrice'].plot(kind='kde', ax=ax,
color='red')

lines, labels = ax.get_legend_handles_labels()
labels = ['Original variable', 'Imputed
variable']
ax.legend(lines, labels, loc='best')

# add title
plt.title('GarageQual')
```

#Plot Bar Fire Place

```
df['FireplaceQu'].value_counts().plot(kind='bar')
```

```
df['FireplaceQu'].mode()
```

#kde Plot | Replace House with GD and NA

```
fig = plt.figure()
ax = fig.add_subplot(111)

df[df['FireplaceQu']=='Gd']['SalePrice'].plot(kin
d='kde', ax=ax)

df[df['FireplaceQu'].isnull()]['SalePrice'].plot(
kind='kde', ax=ax, color='red')

lines, labels = ax.get_legend_handles_labels()
labels = ['Houses with Gd', 'Houses with NA']
ax.legend(lines, labels, loc='best')

plt.title('FireplaceQu')
```


#Store Temp Variable

```
temp = df[df['FireplaceQu']=='Gd']['SalePrice']
```

#Replace missing value with TA

```
df['FireplaceQu'].fillna('Gd', inplace=True)
```

#Draw Bar Plot

```
df['FireplaceQu'].value_counts().plot(kind='bar')
```

#Draw plot again | After Imputation

```
fig = plt.figure()
ax = fig.add_subplot(111)


temp.plot(kind='kde', ax=ax)

# distribution of the variable after imputation

df[df['FireplaceQu'] ==
'Gd']['SalePrice'].plot(kind='kde', ax=ax,
color='red')

lines, labels = ax.get_legend_handles_labels()
labels = ['Original variable', 'Imputed
variable']
ax.legend(lines, labels, loc='best')

# add title
plt.title('FireplaceQu')
```

Missing Value Imputation: Data imputation is a method for retaining the majority of the dataset's data and information by substituting missing data with a different value.

<Start Coding | Missing Value - imputation>

#Import Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

#Import Data

```
df =
pd.read_csv('train.csv',usecols=['GarageQual','Fi
replaceQu','SalePrice'])

----
df.head()
```

#Check missing (null) value

```
df.isnull().mean()*100
```

#Plot bar | Garage

```
df['GarageQual'].value_counts().sort_values(ascen
ding=False).plot.bar()
plt.xlabel('GarageQual')
plt.ylabel('Number of houses')
```

#Replace Missing Value in "Missing" Word

```
df['GarageQual'].fillna('Missing', inplace=True
```

#Category Missing Values

```
df['GarageQual'].value_counts().sort_values(ascend
ing=False).plot.bar()
plt.xlabel('GarageQual')
plt.ylabel('Number of houses')
```

Day 12: Curious Data Minds

- **Data Science and AI in the Travel Industry:**

Read Blog: https://www.altexsoft.com/blog/data-science-and-ai-in-the-travel-industry-9-real-life-use-cases/

https://economictimes.indiatimes.com/jobs/government-jobs/ayodhya-tourism-boom-may-create-150000-200000-direct-and-indirect-jobs/articleshow/107124078.cms?from=mdr

How OYO uses Data Analytics

As of January 2020, it has more than 43,000 properties and 10 lakh (1 million) rooms across 800 cities in 80 countries, including

India, Malaysia, the UAE, Nepal, China, Brazil, Mexico, the UK, Philippines, Japan, Saudi Arabia, Sri Lanka, Indonesia, Vietnam, and the United States.

Area served Asia, Europe and Americas
Revenue ₹4,157 crore

“Our data Analysts use natural curiosity and innovative tools to derive deep insights into customer behavior. These insights not only help us improve our service but also take effective business decisions,"

said Ritesh Agarwal, founder & CEO of OYO.

OYO users spent 3,232 years' worth of time on the OYO app in India – the highest in India in FY2021

Subah-Sham. Quite literally. The most popular time to make bookings on the OYO app were 11:00 AM – 1:00 PM and evening 6:00 PM – 9:00 PM

Fan Alert: A travel agent from India made 1193 bookings for an OYO in 2021