Data Science | 30 Days of Machine Learning | Day - 13

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

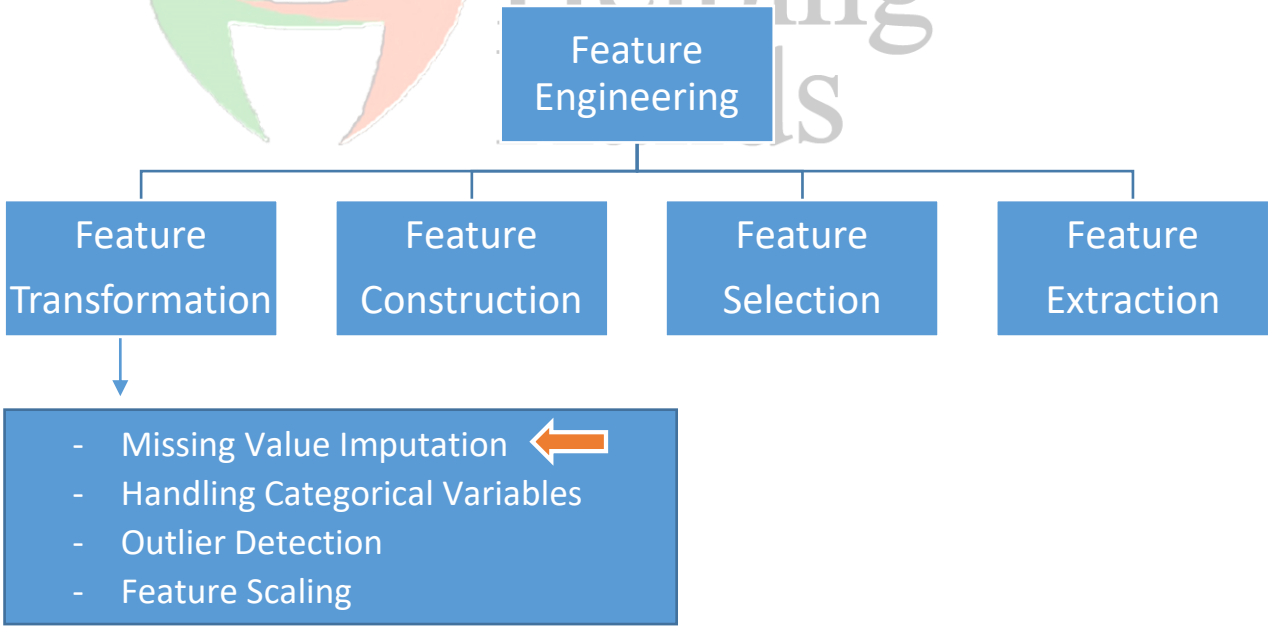
----Today Topics | Day 13----

Feature Engineering (Missing Value Imputation)

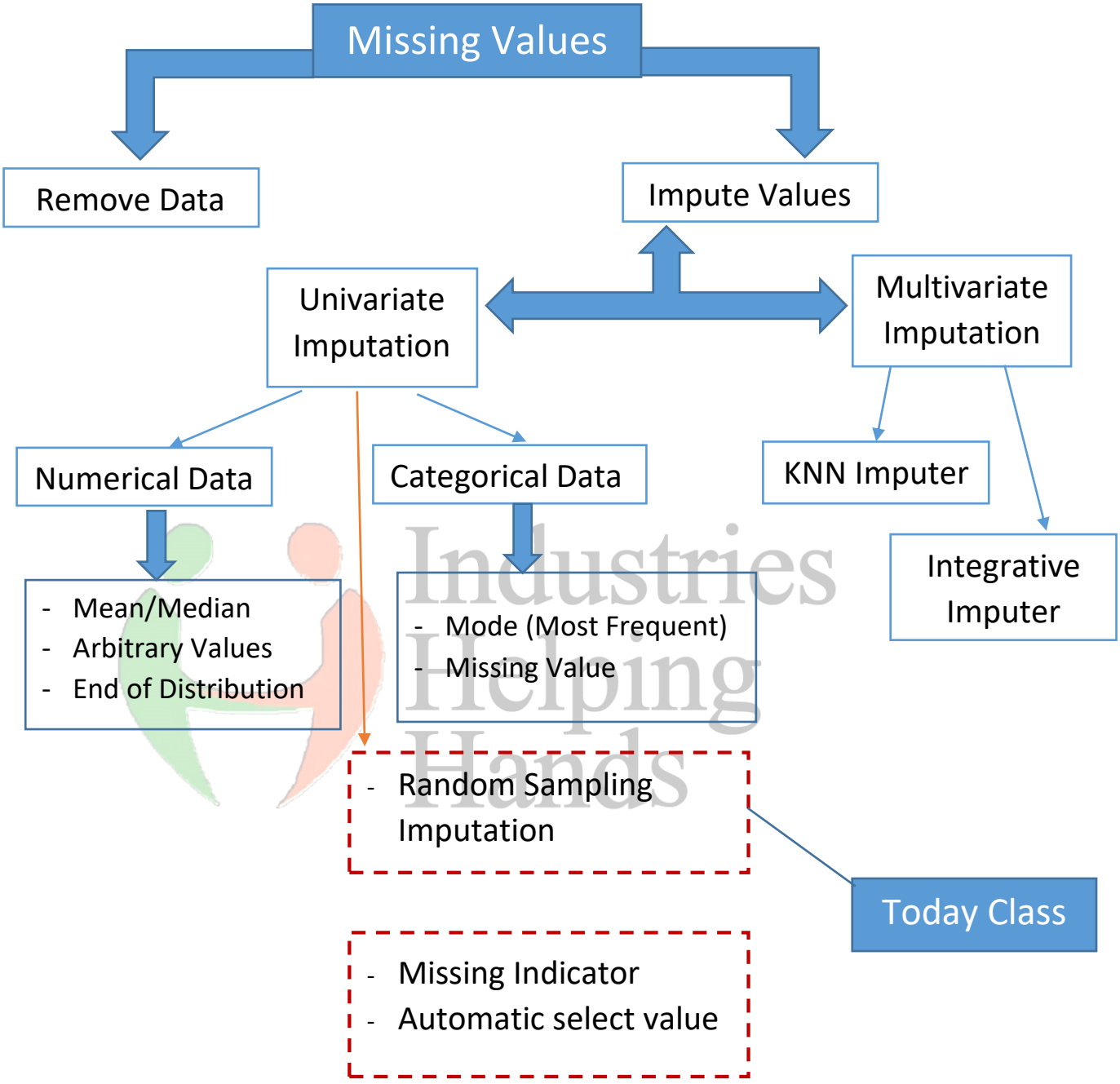
- Random Imputation in Univariate Imputation
- For Numerical Data
- For Categorical Data

- Missing Indicator in Univariate Imputation
- Automatic select value for Imputer parameter

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



Today's Topics:



What is random sampling imputation?

Random sampling imputation consists of extracting random observations from the pool of available values in the variable. Random sampling imputation preserves the original distribution, which differs from the other imputation techniques we've discussed in this chapter and is suitable for numerical and categorical variables alike. In this recipe, we will implement random sample imputation with pandas and Feature-engine.

Example Numerical Data:

Age
15
18
21
N/A
22
15
18

Example Categorical Data:

Gender
M
F
M
N/A
F
M
F

Advantages & Disadvantage:

- Use pandas and implementation easy, not use SK learn uses and good for linear models and not use much in decision tree based algorithm technique.
- Complicated Deployment: Memory heavy for deployment, as we need to store the original training dataset to extract value replace from N/A.
- Well suited for linear algorithms as this does not destroy the distribution, regardless of the % of N/A.

Numeric Data: Titanic & Categorical Data: House Price

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

<Start Coding | Random - Sample - imputation>

#Import Libraries

```
import numpy as np
import pandas as pd

from sklearn.model_selection import
train_test_split

import matplotlib.pyplot as plt
import seaborn as sns
```

#Import Dataset

```
df =
pd.read_csv('train.csv',usecols=['Age','Fare','Su
rvived'])
---
df.head()
```

#Check missing (null) value

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

#Create X & Y

```
X = df.drop(columns=['Survived'])
y = df['Survived']
```

#Apply Train Test Split

```
X_train,X_test,y_train,y_test =
train_test_split(X,y,test_size=0.2,random_state=2
)
```

```
---
```

```
X_train
```

#New column create in Both Train & Test

```
X_train['Age_imputed'] = X_train['Age']
```

```
X_test['Age_imputed'] = X_test['Age']
```

```
----
```

```
X_test.tail()
```

```
X_train.head()
```


#Replace Value Age_imputed

```
X_train['Age_imputed'][X_train['Age_imputed'].isn
ull()] =
X_train['Age'].dropna().sample(X_train['Age'].isn
ull().sum()).values
```

```
X_test['Age_imputed'][X_test['Age_imputed'].isnul
l()] =
X_train['Age'].dropna().sample(X_test['Age'].isnu
ll().sum()).values
```

#Review Sample Random Generate Value

```
X_train['Age'].dropna().sample(1).values
```

```
X_train['Age'].isnull().sum()
```

```
X_train['Age'].dropna().sample(X_train['Age'].isnull().sum()).values
```

```
X_train
```

#Compare Original Age and Imputed Age

```
sns.distplot(X_train['Age'],label='Original',hist
 =False)
sns.distplot(X_train['Age_imputed'],label =
 'Imputed',hist=False)

plt.legend()
plt.show()
```

#Compare Variable Variance

```
print('Original variable variance: ',
 X_train['Age'].var())
print('Variance after random imputation: ',
 X_train['Age_imputed'].var())
```

#Random Sample imputation for Categorical Data

#Import Libraries

```
import numpy as np
import pandas as pd

from sklearn.model_selection import
train_test_split

import matplotlib.pyplot as plt
import seaborn as sns
```

#Import Dataset

```
df = pd.read_csv('house-
train.csv',usecols=['GarageQual','FireplaceQu',
'SalePrice'])
```

---

df.head()

#Check missing (null) value

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

#Create X & Y

```
X = df
```

```
y = df['SalePrice']
```

 # Apply Train Test Split

```
X_train,X_test,y_train,y_test =
train_test_split(X,y,test_size=0.2,random_state=2
)
```

#New column create in Both Garage & Fire place

```
X_train['GarageQual_imputed'] = X_train['GarageQual']
X_test['GarageQual_imputed'] = X_test['GarageQual']

X_train['FireplaceQu_imputed'] =
X_train['FireplaceQu']
X_test['FireplaceQu_imputed'] = X_test['FireplaceQu']

----
X_train.sample(5)
```

#Replace Garage Value and Fireplace Imputed

```
X_train['GarageQual_imputed'][X_train['GarageQual_impu
ted'].isnull()] =
X_train['GarageQual'].dropna().sample(X_train['GarageQ
ual'].isnull().sum()).values
X_test['GarageQual_imputed'][X_test['GarageQual_impute
d'].isnull()] =
X_train['GarageQual'].dropna().sample(X_test['GarageQu
al'].isnull().sum()).values

X_train['FireplaceQu_imputed'][X_train['FireplaceQu_im
puted'].isnull()] =
X_train['FireplaceQu'].dropna().sample(X_train['Firepl
aceQu'].isnull().sum()).values
X_test['FireplaceQu_imputed'][X_test['FireplaceQu_impu
ted'].isnull()] =
```

```
X_train['FireplaceQu'].dropna().sample(X_test['Firepla
ceQu'].isnull().sum()).values
```

#Review Frequency in Garage Original & Imputed

```
temp = pd.concat(
        [
            X_train['GarageQual'].value_counts()
/ len(X_train['GarageQual'].dropna()),

X_train['GarageQual_imputed'].value_counts() /
len(X_train)
        ],
        axis=1)
temp.columns = ['original', 'imputed']
temp
```

#Review Frequency in Fireplace Original & Imputed

```
temp = pd.concat(
        [
            X_train['FireplaceQu'].value_counts()
/ len(X_train['FireplaceQu'].dropna()),

X_train['FireplaceQu_imputed'].value_counts() /
len(df)
        ],
        axis=1)

temp.columns = ['original', 'imputed']

temp
```

#Compare category Fireplace before Imputation

```
for category in
X_train['FireplaceQu'].dropna().unique():
    sns.distplot(X_train[X_train['FireplaceQu'] ==
category]['SalePrice'],hist=False,label=category)
plt.show()
```

#Compare category fireplace after Imputation

```
for category in
X_train['FireplaceQu_imputed'].dropna().unique():

sns.distplot(X_train[X_train['FireplaceQu_imputed
'] ==
category]['SalePrice'],hist=False,label=category)
plt.show()
```

Day 13: 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