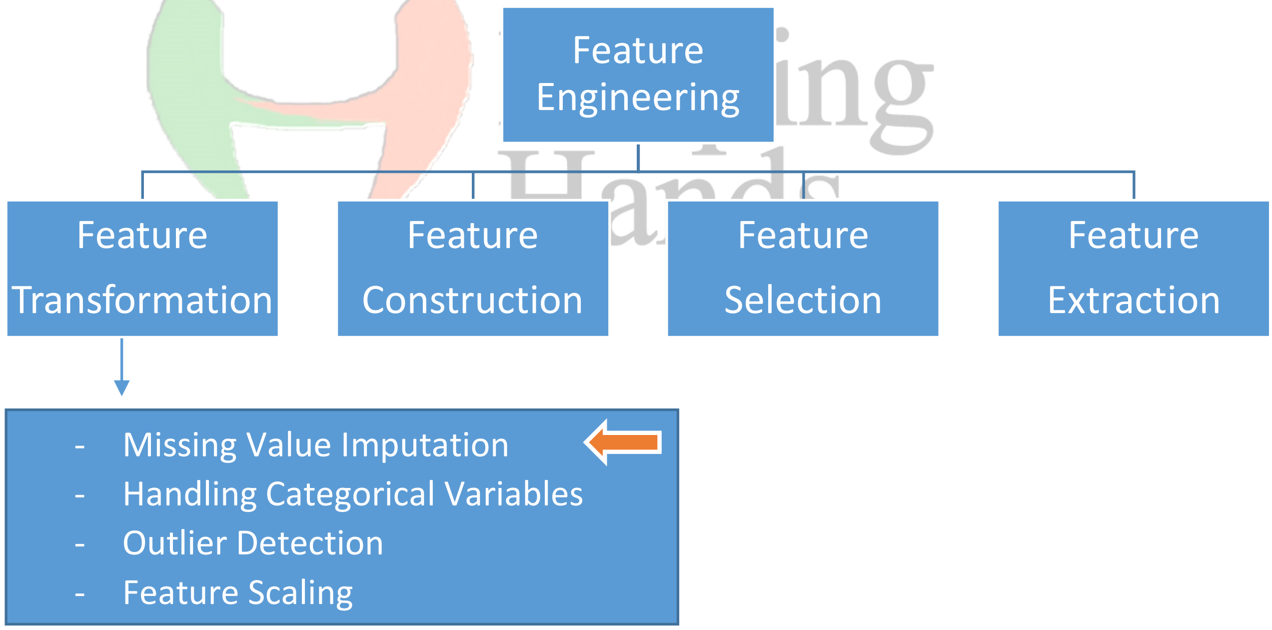
Data Science | 30 Days of Machine Learning | Day - 10

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

----Today Topics | Day 10----

- **Handling Missing Data**
- What are the problems with missing data?
- Remove Missing Values
- What is a missing completely at random (MCAR)?
- Pro and Corns CCA?
- When we use CCA?

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



- What are the problems with missing data?

The real-world data often has a lot of missing values. The cause of missing values can be data corruption or failure to record data. The handling of missing data is very important during the pre-processing of the dataset as many machine learning algorithms do not support missing values.

- 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)

- **Remove Missing Values**
  - CCA: Complete Case Analysis
    The standard treatment of missing data in most statistical packages is complete case analysis (CCA) done by case wise deletion. Any observation that has a missing value for any variable is automatically discarded and only complete observations are analysed.

    This means analysing only those observations for which all variables in the dataset have information.

  - Acceptance for CCA
    - What is a missing completely at random (MCAR)?
    - Pro and Corns CCA?
    - When we use CCA?

<Start Coding>

#Import Libraries

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

#Import Dataset

```
df = pd.read_csv('data_science_job.csv')
```

```
df.head()
```

#Finding Missing Data Column Wise

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

```
df.shape
```

#Selected column name (Below 5%)

```
cols = [var for var in df.columns if
df[var].isnull().mean() < 0.05 and
df[var].isnull().mean() > 0]
cols
```

```
df[cols].sample(5)
```


#Calculated drop rows

```
len(df[cols].dropna()) / len(df)
```

#Create: New Data Frame

```
new_df = df[cols].dropna()
df.shape, new_df.shape
```

#Plot Histogram (Before Applying CCA: Numerical Data)

```
new_df.hist(bins=50, density=True, figsize=(12,
12))
plt.show()
```

#Plot Histogram: Training Hours

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

# original data
df['training_hours'].hist(bins=50, ax=ax,
density=True, color='red')

# data after cca, the argument alpha makes the
color transparent, so we can
# see the overlay of the 2 distributions
new_df['training_hours'].hist(bins=50, ax=ax,
color='green', density=True, alpha=0.8)
```

#Plot Probability Density Function (PDF): Training Hours

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

# original data
df['training_hours'].plot.density(color='red')

# data after cca
new_df['training_hours'].plot.density(color='gree
n')
```

#Plot Histogram: City Development

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

# original data
df['city_development_index'].hist(bins=50, ax=ax,
density=True, color='red')

# data after cca, the argument alpha makes the
color transparent, so we can
# see the overlay of the 2 distributions
new_df['city_development_index'].hist(bins=50,
ax=ax, color='green', density=True, alpha=0.8)
```


#Plot PDF: City Development

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

# original data
df['city_development_index'].plot.density(color='r
ed')

# data after cca
new_df['city_development_index'].plot.density(colo
r='green')
```

#Plot Histogram: Experience

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

# original data
df['experience'].hist(bins=50, ax=ax,
density=True, color='red')

# data after cca, the argument alpha makes the
color transparent, so we can
# see the overlay of the 2 distributions
new_df['experience'].hist(bins=50, ax=ax,
color='green', density=True, alpha=0.8)
```

#Plot PDF: Experience

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

# original data
df['experience'].plot.density(color='red')

# data after cca
new_df['experience'].plot.density(color='green')
```

#CCA in Categorical Data

```
df['education_level'].value_counts()
```

```
df['enrolled_university'].value_counts()
```

#CCA Apply in Enrolled University

```
temp = pd.concat([
# percentage of observations per category,
original data

df['enrolled_university'].value_counts() /
len(df),

# percentage of observations per category, cca
data

new_df['enrolled_university'].value_counts() /
len(new_df)
        ],
        axis=1)

# add column names
temp.columns = ['original', 'cca']

temp
```

#CCA Apply in Education Level

```
temp = pd.concat([
 # percentage of observations per category,
original data
            df['education_level'].value_counts() /
len(df),

  # percentage of observations per category, cca
data

new_df['education_level'].value_counts() /
len(new_df)
        ],
        axis=1)

# add column names
temp.columns = ['original', 'cca']

temp
```

Day 10: Curious Data Minds

Dhanurjay "DJ" Patil

https://en.wikipedia.org/wiki/DJ_Patil

Dhanurjay "DJ" Patil (born August 3, 1974) is an American mathematician and computer scientist, White House announced Patil would be the first U.S. Chief Data Scientist

Dr. Patil public policy work includes being appointed by President Obama to be the first U.S. Chief Data Scientist where his efforts led to the establishment of nearly 40 Chief Data Officer roles across the Federal government.

Head of Data Products and Chief Scientist of LinkedIn

We’re all products of failure.

While growing up in California, to simply say I was bad at Math would have been an understatement. My freshman year of high school, I was kicked out of my algebra class.

By the time high school graduation came around, I almost didn’t graduate. For the record, I did actually graduate, but it was only because a very kind administrator took pity on me and changed my failing grade in chemistry to a passing one.