Data Reduction

Sometimes we have our data collected with as much information as possible. However, some attributes do not contribute to our analysis, and we may need to do dimension elimination to focus in the attributes we need. Dimension elimination is one way of reducing the complexity of your data, and you can use your domain knoweldge to justify the reasons.

We could also do feature extraction, such as Priciple Component Analysis (PCA). We will learn that technique in Data Analysis, unsupervised learning course.

Setup

```
import numpy as np
import pandas as pd
```

```
df = pd.read_csv('/content/sample_data/california_housing_train.csv')
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17000 entries, 0 to 16999
Data columns (total 9 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   longitude           17000 non-null  float64
 1   latitude            17000 non-null  float64
 2   housing_median_age  17000 non-null  float64
 3   total_rooms         17000 non-null  float64
 4   total_bedrooms      17000 non-null  float64
 5   population          17000 non-null  float64
 6   households          17000 non-null  float64
 7   median_income       17000 non-null  float64
 8   median_house_value  17000 non-null  float64
dtypes: float64(9)
memory usage: 1.2 MB
```

Dimension Elimination

```
df_sample1 = df[df.columns[2:]]
df_sample1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17000 entries, 0 to 16999
Data columns (total 7 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   housing_median_age  17000 non-null  float64
 1   total_rooms         17000 non-null  float64
 2   total_bedrooms      17000 non-null  float64
 3   population          17000 non-null  float64
 4   households          17000 non-null  float64
 5   median_income       17000 non-null  float64
 6   median_house_value  17000 non-null  float64
dtypes: float64(7)
memory usage: 929.8 KB
```

```
df_sample2 = df.drop(df.columns[:2], axis = 1)
df_sample2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17000 entries, 0 to 16999
Data columns (total 7 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   housing_median_age  17000 non-null  float64
 1   total_rooms         17000 non-null  float64
 2   total_bedrooms      17000 non-null  float64
 3   population          17000 non-null  float64
 4   households          17000 non-null  float64
 5   median_income       17000 non-null  float64
 6   median_house_value  17000 non-null  float64
dtypes: float64(7)
memory usage: 929.8 KB
```

```
needed_cols = ['total_rooms', 'total_bedrooms', 'population', 'households']
df_sample3 = df[needed_cols]
df_sample3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17000 entries, 0 to 16999
Data columns (total 4 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   total_rooms     17000 non-null  float64
 1   total_bedrooms  17000 non-null  float64
 2   population      17000 non-null  float64
 3   households      17000 non-null  float64
dtypes: float64(4)
memory usage: 531.4 KB
```

```
dontneeded_cols = ['latitude', 'longitude', 'median_income', 'median_house_value']
df_sample4 = df.drop(dontneeded_cols, axis = 1)
df_sample4.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17000 entries, 0 to 16999
Data columns (total 5 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   housing_median_age  17000 non-null  float64
 1   total_rooms         17000 non-null  float64
 2   total_bedrooms      17000 non-null  float64
 3   population          17000 non-null  float64
 4   households          17000 non-null  float64
dtypes: float64(5)
memory usage: 664.2 KB
```

Colab paid products - Cancel contracts here