3803ICT
Big Data Analysis

Lab 02 – Data Preparation and Preprocessing

Trimester 1 - 2021

Table of Contents

I. General information

All of your workshops and assignment need to be submitted to GitHub and shared with github ID "3803ict" for reviewing.

II. Basic functions

If you are not familiar with Pandas, look at this cheat sheet for your reference: https://pandas.pydata.org/Pandas_Cheat_Sheet.pdf.

Create new jupyter notebook and implement the following features:

- Load PastHires.csv file using read_csv function.
- Visualize top 10 row using head() function.

Out[2]:

	Years Experience	Employed?	Previous employers	Level of Education	Top-tier school	Interned	Hired
0	10	Y	4	BS	N	N	Y
1	0	N	0	BS	Y	Y	Y
2	7	N	6	BS	N	N	N
3	2	Y	1	MS	Y	N	Y
4	20	N	2	PhD	Y	N	N
5	0	N	0	PhD	Y	Y	Y
6	5	Y	2	MS	N	Y	Y
7	3	N	1	BS	N	Y	Y
8	15	Y	5	BS	N	N	Y
9	0	N	0	BS	N	N	N

- Print the list of employees sorted by Years Experience.

Out[13]:

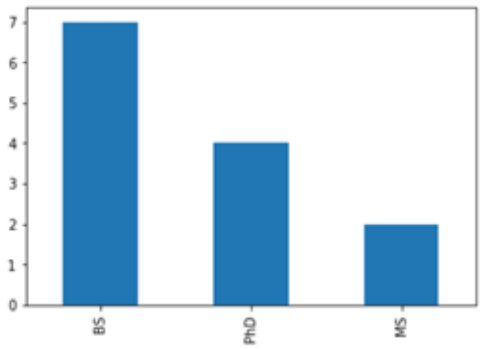
	Years Experience	Employed?	Previous employers	Level of Education	Top-tier school	Interned	Hired
1	0	N	0	BS	Y	Y	Y
5	0	N	0	PhD	Y	Y	Y
9	0	N	0	BS	N	N	N
12	0	N	0	PhD	Y	N	Y
10	1	N	1	PhD	Y	N	N
3	2	Y	1	MS	Y	N	Y
7	3	N	1	BS	N	Y	Y
11	4	Y	1	BS	N	Y	Y
6	5	Y	2	MS	N	Y	Y
2	7	N	6	BS	N	N	N
0	10	Y	4	BS	N	N	Y
8	15	Y	5	BS	N	N	Y
4	20	N	2	PhD	Y	N	N

- Use value_counts function to find the distribution of "Level of Education".

```
Out[14]: BS     7
         PhD    4
         MS     2
         Name: Level of Education, dtype: int64
```

- Visualize above result using plot()

```
Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x28f6d2b0240>
```



III. Missing Data and Data Cleaning

Fill in the TODO cells in sales.ipynb notebook.

- ❖ Fix column datatypes.
- ❖ Drop if duplicated or null.
- ❖ Sanity check for value ranges and to check assumptions.
- ❖ Use regular expression and lambda function to parse data.

IV. Data preprocessing exercise

Given the job market data in csv file. Create your own jupyter notebook and explore the data by:

- ❖ Load the data using Pandas.
- ❖ Visualize top 10 first rows
- ❖ Fix column datatypes.
- ❖ Check and clean the data.

V. Dimensionality Reduction

1. Principal Component Analysis (PCA)

Apply PCA to reduce the dimension of Iris dataset (the dataset has 150 samples individual flowers and three distinct Iris species that each flower is classified into).

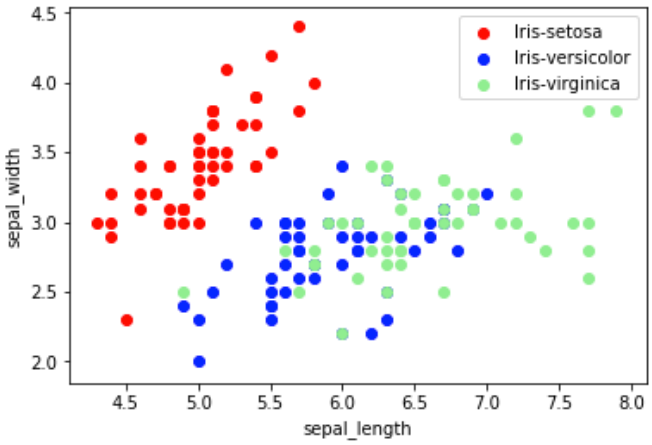
- ❖ Load the dataset

In [68]:
```
import matplotlib.pyplot as plt
import pandas as pd

from sklearn.decomposition import PCA as sklearnPCA
```

In [69]:
```
url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'
data = pd.read_csv(url,header=None)

y = data[4] # Split off classifications
X = data.iloc[:,0:4] # Split off features
```

❖ Visualize the dataset into scatter series.

In [75]:
```
# three different scatter series so the class labels in the legend are distinct
plt.scatter(X[y=='Iris-setosa'].iloc[:,0], X[y=='Iris-setosa'].iloc[:,1], label='Iris-setosa', c='red')
plt.scatter(X[y=='Iris-versicolor'].iloc[:,0], X[y=='Iris-versicolor'].iloc[:,1], label='Iris-versicolor', c='blue')
plt.scatter(X[y=='Iris-virginica'].iloc[:,0], X[y=='Iris-virginica'].iloc[:,1], label='Iris-virginica', c='lightgreen')

# Prettify the graph
plt.legend()
plt.xlabel('sepal_length')
plt.ylabel('sepal_width')

# display
plt.show()
```

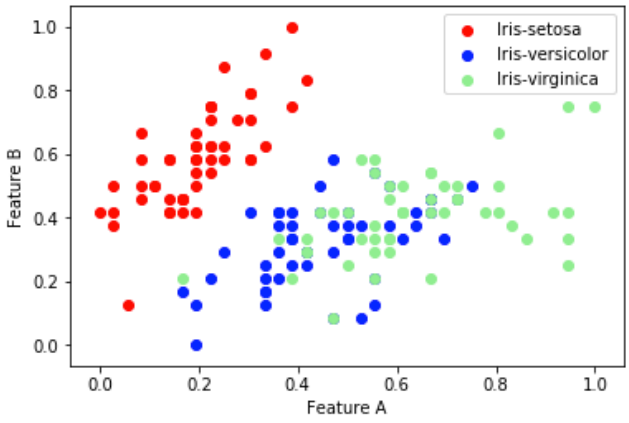


❖ Rescale the data to a [0,1] range.

In [71]:
```
X_norm = (X - X.min())/(X.max() - X.min())
```

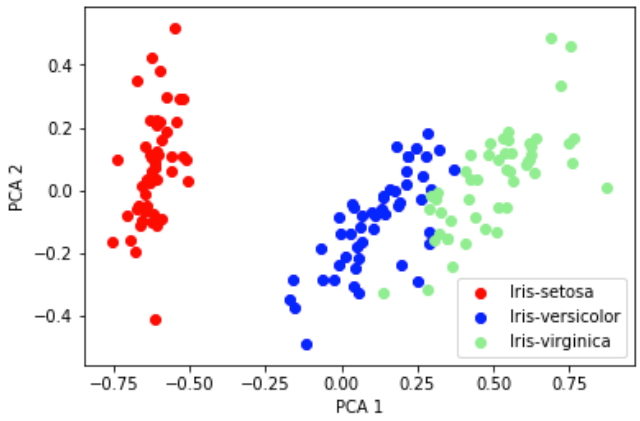
❖ Plot the X_norm again:

In [72]:
```
# three different scatter series so the class labels in the legend are distinct
plt.scatter(X_norm[y=='Iris-setosa'].iloc[:,0], X_norm[y=='Iris-setosa'].iloc[:,1], label='Iris-setosa', c='red')
plt.scatter(X_norm[y=='Iris-versicolor'].iloc[:,0], X_norm[y=='Iris-versicolor'].iloc[:,1], label='Iris-versicolor', c='blue')
plt.scatter(X_norm[y=='Iris-virginica'].iloc[:,0], X_norm[y=='Iris-virginica'].iloc[:,1], label='Iris-virginica', c='lightgreen')

# Prettify the graph
plt.legend()
plt.xlabel('Feature A')
plt.ylabel('Feature B')

# display
plt.show()
```



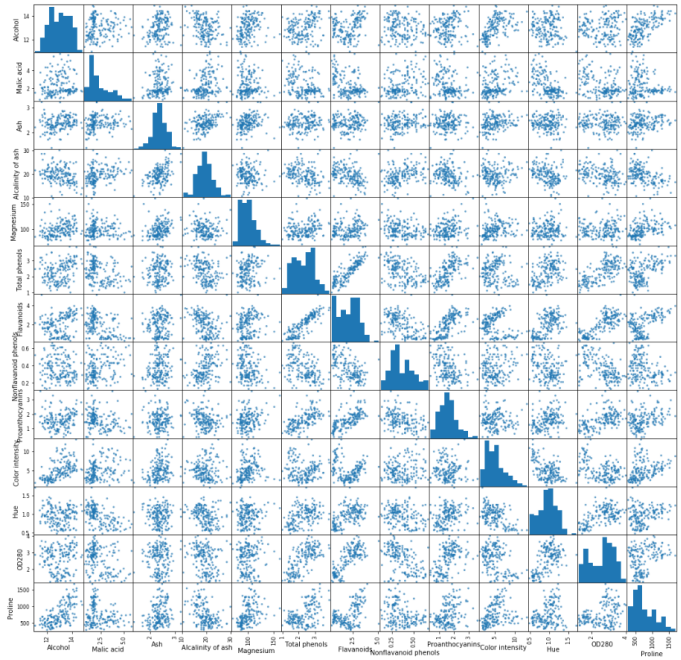
❖ Use sklearn PCA object to the normalized dataset and visualize it again in the plot.

In [73]:
```
pca = sklearnPCA(n_components=2) #2-dimensional PCA
transformed = pd.DataFrame(pca.fit_transform(X_norm))
```

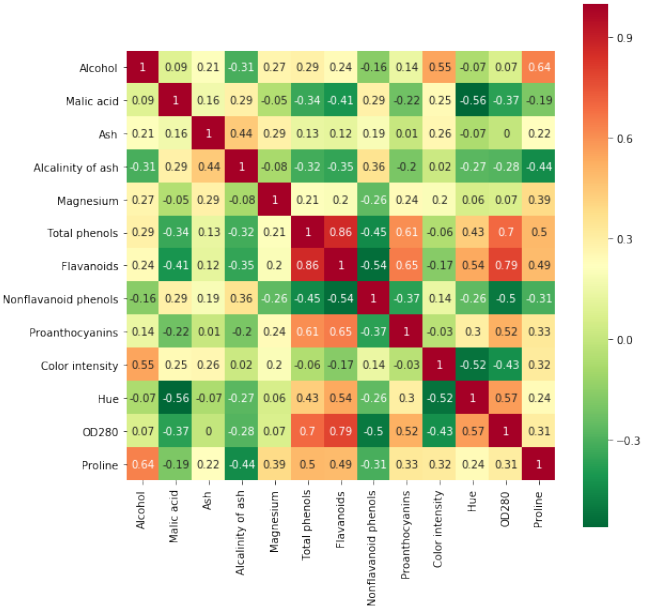


2. Exercise

- Load data from wine.data.csv file. Keep 1st column into a separate variable (label) and remove it from DataFrame.
- Use Scatter plot to learn attributes of data. What is your conclusion?



- Try to visualize data with correlation heatmap? Can you find any pairs of attributes which have large correlation?



- Normalize data by removing the mean and scaling to unit variance using `preprocessing.StandardScaler`.

  Hint:
  standardScaler = preprocessing.StandardScaler()
  standardScaler.fit(wine)
  X_scaled_array = standardScaler.transform(wine)
  normalizedData = pd.DataFrame(X_scaled_array, columns = wine.columns)

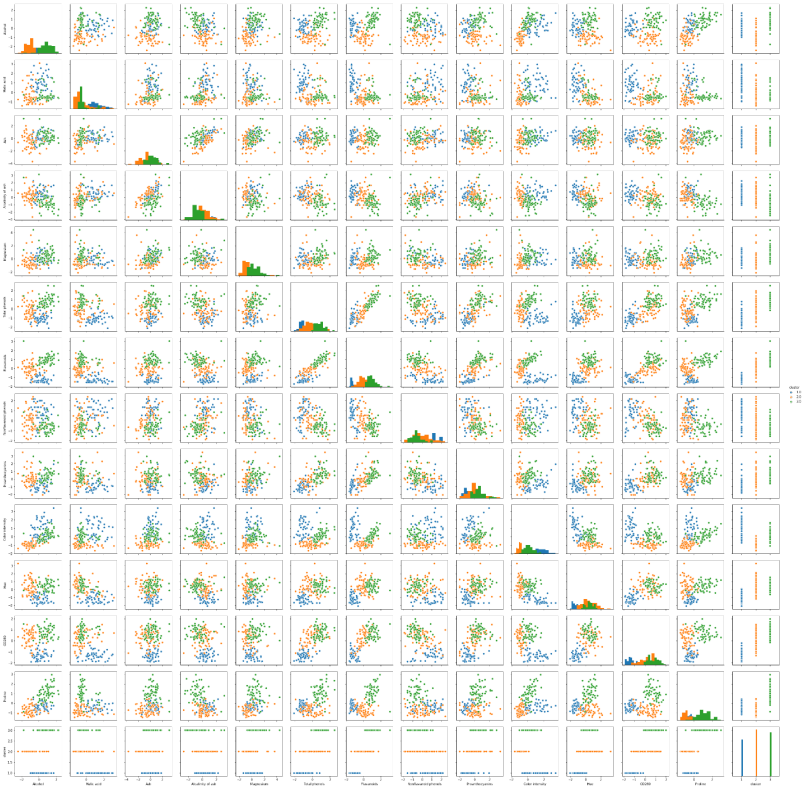
- Use kMeans to cluster the normalized data (By using Elbow method https://en.wikipedia.org/wiki/Elbow_method_(clustering) , the number of clusters should be 3). Use pairplot to visualize the wine attributes with their cluster.

  Hint:
  kMeansClustering = KMeans(n_clusters = 3, random_state=seed)
  res = kMeansClustering.fit_predict(normalizedData)

  ---

  normalizedData ["cluster"] = label_pred_KM.astype('float64')
  sns_plot = sns.pairplot(normalizedData, hue = "cluster",diag_kind="hist")

- By using explained_variance_ratio_ attribute, we know that first 6 PCs explain 85.1% of variance if we reduce the dimension using PCA. You need to apply PCA with 6 components for the above normalized data. Then applying kMeans (3 clusters) to cluster the data after dimensionality reduction.



- Use adjusted_rand_score in sklearn.metrics.cluster to calculate the scores of original kMeans and kMeans after PCA. What is your conclusion?
  Hint:
  adjusted_rand_score(label, label_pred_KM_PCA) with label is kept in 1st step.