Random Forest

YoungWoong Cho

July 2020

In this project, random forest algorithm is used to train the model. Feature importance is plotted for the analysis of the dataset.

Dataset : Breast Cancer Wisconsin

Source : https://www.kaggle.com/uciml/breast-cancer-wisconsin-data

Features : ID Number, Clump Thickness, Uniformity of Cell Size, Uniformity of Cell Shape, Marginal Adhesion, Single Epithelial Cell Size, Bare Nuclei, Bland Chromatin, Normal Nucleoli, Mitoses

Label : Class

0.1 Preliminary works

1. Import libraries

[ ]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
```

2. Get a dataset

[ ]:
```
from google.colab import files
data_to_load = files.upload()
```

```
<IPython.core.display.HTML object>


Saving breast-cancer-wisconsin.csv to breast-cancer-wisconsin.csv
```

3. Create train, test and validation sets

[ ]:
```
data = pd.read_csv("breast-cancer-wisconsin.csv").dropna()
data.columns = ['ID Number', 'Clump Thickness', 'Uniformity of Cell Size',␣
 ↪'Uniformity of Cell Shape', 'Marginal Adhesion', 'Single Epithelial Cell Size'␣
 ↪, 'Bare Nuclei', 'Bland Chromatin', 'Normal Nucleoli', 'Mitoses', 'Class']
```

```
data['Class'] = data['Class'].replace(2,0).replace(4,1) #Replace 2(benign) and␣
 ↪4(malignant) into 0(benign) and 1(malignant)

X = data.iloc[:, 1:-1]
Y = data.iloc[:, -1]
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3)
```

0.2 Random Forest algorithm

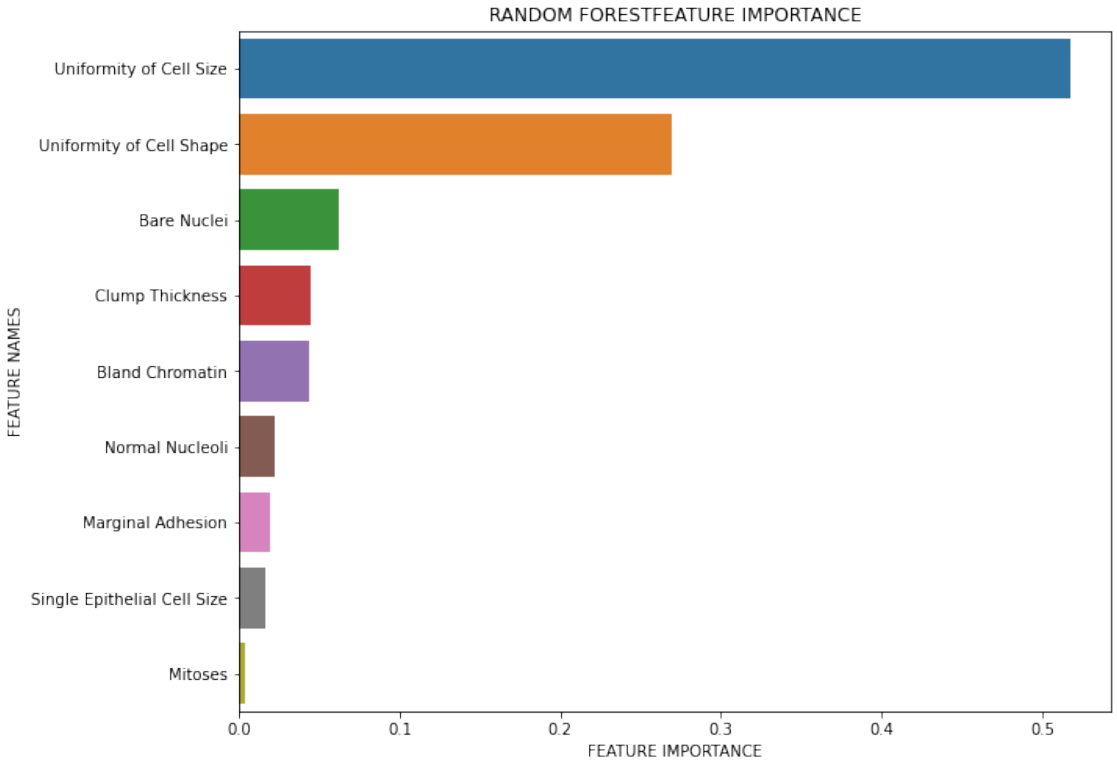
[ ]:
```
rf = RandomForestRegressor(n_estimators=100)
rf.fit(X_train, Y_train)
Y_pred = rf.predict(X_test)
print("%.4f" %rf.score(X_train, Y_train))
print("%.4f" %rf.score(X_test, Y_test))
```

```
0.9832
0.8521
```

Feature importance is plotted for the analysis.

[ ]:
```
def plot_feature_importance(importance,names,model_type):

  #Create arrays from feature importance and feature names
  feature_importance = np.array(importance)
  feature_names = np.array(names)

  #Create a DataFrame using a Dictionary
  data={'feature_names':feature_names,'feature_importance':feature_importance}
  df = pd.DataFrame(data)

  #Sort the DataFrame in order decreasing feature importance
  df.sort_values(by=['feature_importance'], ascending=False,inplace=True)

  #Define size of bar plot
  plt.figure(figsize=(10,8))
  #Plot Searborn bar chart
  sns.barplot(x=df['feature_importance'], y=df['feature_names'])
  #Add chart labels
  plt.title(model_type + 'FEATURE IMPORTANCE')
  plt.xlabel('FEATURE IMPORTANCE')
  plt.ylabel('FEATURE NAMES')
```

[ ]:
```
plot_feature_importance(rf.feature_importances_,X_train.columns,'RANDOM FOREST')
```


RANDOM FORESTFEATURE IMPORTANCE
Uniformity of Cell Size
Uniformity of Cell Shape
Bare Nuclei
Clump Thickness
Bland Chromatin
Normal Nucleoli
Marginal Adhesion
Single Epithelial Cell Size
Mitoses
FEATURE NAMES
0.0
0.1
0.2
0.3
0.4
0.5
FEATURE IMPORTANCE