**EXERCISE 3**

1. SELECT FEATURES

Tất cả chúng ta có thể đã phải đối mặt với vấn đề này khi xác định các tính năng liên quan từ một tập hợp dữ liệu và loại bỏ các tính năng không liên quan hoặc ít quan trọng hơn không đóng góp nhiều vào biến mục tiêu của chúng ta để đạt được độ chính xác tốt hơn cho mô hình của chúng ta.

**Lựa chọn tính năng là một trong những khái niệm cốt lõi trong học máy tác động rất lớn đến hiệu suất của mô hình của bạn.**Các tính năng dữ liệu mà bạn sử dụng để đào tạo các mô hình học máy của mình có ảnh hưởng rất lớn đến hiệu suất mà bạn có thể đạt được.

Các tính năng không liên quan hoặc có liên quan một phần có thể tác động tiêu cực đến hiệu suất của mô hình.

Lựa chọn tính năng và làm sạch dữ liệu phải là bước đầu tiên và quan trọng nhất trong quá trình thiết kế mô hình của bạn.

Lựa chọn tính năng là quá trình bạn chọn tự động hoặc thủ công những tính năng đóng góp nhiều nhất vào biến dự đoán của bạn hoặc kết quả mà bạn quan tâm.

Có các tính năng không liên quan trong dữ liệu của bạn có thể làm giảm độ chính xác của mô hình và khiến mô hình của bạn học dựa trên các tính năng không liên quan.

**Cách chọn tính năng và lợi ích của việc thực hiện lựa chọn tính năng trước khi lập mô hình dữ liệu của bạn là gì?**

· **Giảm Overfitting** : Ít dữ liệu dư thừa hơn có nghĩa là ít cơ hội để đưa ra quyết định dựa trên nhiễu.

· **Cải thiện độ chính xác** : Dữ liệu ít sai lệch hơn có nghĩa là độ chính xác của mô hình được cải thiện.

· **Giảm thời gian huấn luyện** : ít điểm dữ liệu hơn làm giảm độ phức tạp của thuật toán và thuật toán huấn luyện nhanh hơn.

**Phương pháp lựa chọn tính năng:**

Univariate Selection

Feature Importance

Correlation Matrix with Heatmap

1. **Univariate Selection**

Kiểm tra thống kê có thể được sử dụng để chọn những đặc điểm có mối quan hệ chặt chẽ nhất với biến đầu ra.

Thư viện scikit-learning cung cấp lớp [SelectKBest](http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html" \l "sklearn.feature_selection.SelectKBest" \t "_blank) có thể được sử dụng với một bộ các bài kiểm tra thống kê khác nhau để chọn một số tính năng cụ thể.

Ví dụ bên dưới sử dụng kiểm tra thống kê chi bình phương (chi²) cho các tính năng không âm để chọn 10 tính năng tốt nhất từ ​​Tập dữ liệu dự đoán phạm vi giá trên thiết bị di động.

import pandas as pd

import numpy as np

from sklearn.feature\_selection import SelectKBest

from sklearn.feature\_selection import chi2

data = pd.read\_csv("./test.csv")

X = data.iloc[:,0:20]  #independent columns

y = data.iloc[:,-1]    #target column i.e price range

#apply SelectKBest class to extract top 10 best features

bestfeatures = SelectKBest(score\_func=chi2, k=10)

fit = bestfeatures.fit(X,y)

dfscores = pd.DataFrame(fit.scores\_)

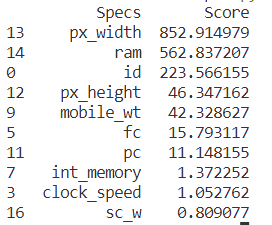
dfcolumns = pd.DataFrame(X.columns)

#concat two dataframes for better visualization

featureScores = pd.concat([dfcolumns,dfscores],axis=1)

featureScores.columns = ['Specs','Score']  #naming the dataframe columns

print(featureScores.nlargest(10,'Score'))  #print 10 best features

****

1. **Feature Importance**

Bạn có thể nhận được mức độ quan trọng của tính năng của từng tính năng trong tập dữ liệu của mình bằng cách sử dụng thuộc tính mức độ quan trọng của tính năng của mô hình.

Mức độ quan trọng của tính năng cung cấp cho bạn điểm số cho từng tính năng trong dữ liệu của bạn, điểm càng cao thì tính năng càng quan trọng hoặc có liên quan là tính năng đối với biến đầu ra của bạn.

Tầm quan trọng của tính năng là một lớp có sẵn đi kèm với Bộ phân loại dựa trên cây, chúng tôi sẽ sử dụng Bộ phân loại cây bổ sung để trích xuất 10 tính năng hàng đầu cho tập dữ liệu.

import pandas as pd

import numpy as np

data = pd.read\_csv("./test.csv")

X = data.iloc[:,0:20]  #independent columns

y = data.iloc[:,-1]    #target column i.e price range

from sklearn.ensemble import ExtraTreesClassifier

import matplotlib.pyplot as plt

model = ExtraTreesClassifier()

model.fit(X,y)

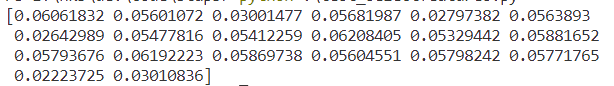
print(model.feature\_importances\_) #use inbuilt class feature\_importances of tree based classifiers

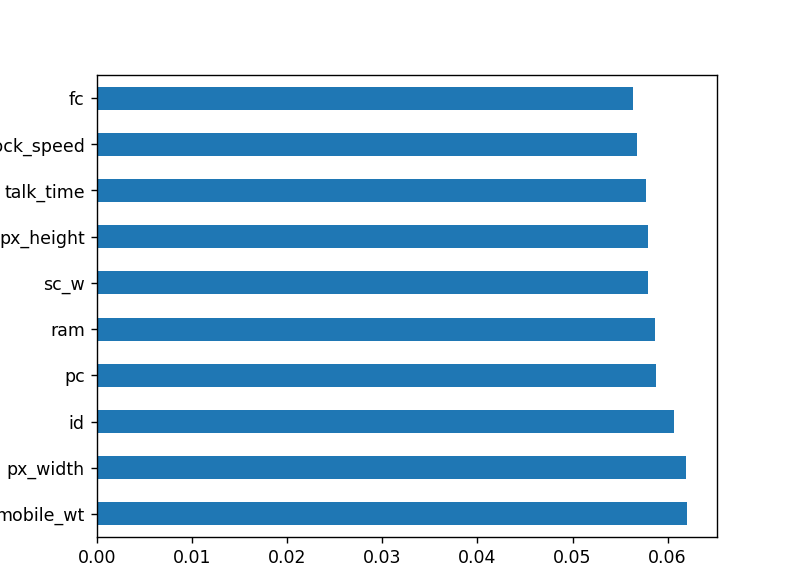
#plot graph of feature importances for better visualization

feat\_importances = pd.Series(model.feature\_importances\_, index=X.columns)

feat\_importances.nlargest(10).plot(kind='barh')

plt.show()



****

1. **Correlation Matrix with Heatmap**

Sự tương quan cho biết các tính năng có liên quan như thế nào với nhau hoặc với biến mục tiêu.

Mối tương quan có thể là dương (tăng một giá trị của đối tượng làm tăng giá trị của biến mục tiêu) hoặc tiêu cực (tăng một giá trị của đối tượng làm giảm giá trị của biến mục tiêu)

Bản đồ nhiệt giúp dễ dàng xác định các đối tượng địa lý nào có liên quan nhiều nhất đến biến mục tiêu, chúng tôi sẽ vẽ bản đồ nhiệt các đối tượng địa lý tương quan bằng cách sử dụng thư viện seaborn.

import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

import seaborn as sns

data = pd.read\_csv("./test.csv")

X = data.iloc[:,0:20]  #independent columns

y = data.iloc[:,-1]    #target column i.e price range

#get correlations of each features in dataset

corrmat = data.corr()

top\_corr\_features = corrmat.index

plt.figure(figsize=(20,20))

#plot heat map

g=sns.heatmap(data[top\_corr\_features].corr(),annot=True,cmap="RdYlGn")

plt.show()

