In [3]:

PHITRON AI/ML

অধ্যায় ৬ঃ ডেটাসেট থেকে গুরুত্বপূর্ন ফিচার সিলেক্ট করা

Author: MD. Emdadul Hoque Tareque (Lead, Phitron)

Provide by: **Phitron AI/ML**

Community Link: **Phitron AI/ML**

একটা মেশিন লার্নিং মডেল তৈরির আগে সবচেয়ে গুরুত্বপূর্ন হচ্ছে ডেটা প্রিপ্রসেসিং। ডেটা প্রিপ্রসেসিং হচ্ছে ডেটাসেটের উপর এমন কিছু কাজ করা যেন ডেটাসেটের মধ্যে থাকা মিসিং ভ্যালু, নয়েজ, ডুপ্লিকেট, স্কেলিং সমস্যা, ক্যাটেগোরিকাল ভেরিয়েবল, আউটলাইয়ারস, অদরকারী ফিচার ইত্যাদিকে হ্যান্ডেল করা যায়। **মেশিন লার্নিং-এ Feature মানে হলো – ডেটার ইনপুট ভ্যারিয়েবল, যেগুলো ব্যবহার করে মডেল টার্গেট (output) প্রেডিক্ট করতে শেখে।**

এই অধ্যায়ে আমরা ডেটা সেটের মধ্যে অদরকারী ফিচারগুলোকে ডিটেক্ট করে সেগুলোকে কিভাবে হ্যান্ডেল করার যায় তা জানব। **এখানে অদরকারী ফিচার বলতে বুঝানো হচ্ছে যে ফিচারগুলো টার্গেট ভ্যারিয়েবল প্রেডিক্ট করতে সাহায্য করছে না সেগুলোকে অদরকারী ফিচার বলা হয়।**

ফিচার সিলেকশন কেন গুরুত্বপূর্ণ?

একটি ML মডেলে ডেটা সেটে অনেক ফিচার থাকতে পারে, কিন্তু সব ফিচার সমানভাবে গুরুত্বপূর্ণ নয়। কিছু ফিচার অপ্রাসঙ্গিক বা রিডান্ডান্ট হতে পারে, যা মডেলের পারফরম্যান্সের উপর নেতিবাচক প্রভাব ফেলে। ফিচার সিলেকশনের প্রধান সুবিধাগুলো হলো:

- **মডেলের নির্ভুলতা বৃদ্ধি:** প্রাসঙ্গিক ফিচারগুলো নির্বাচন করলে মডেলের প্রেডিক্টিভ ক্ষমতা বাড়ে।
- **ওভারফিটিং হ্রাস:** অপ্রয়োজনীয় ফিচার বাদ দিয়ে মডেলের জেনারালাইজেশন ক্ষমতা বৃদ্ধি পায়।
- **কম্পিউটেশনাল দক্ষতা:** কম ফিচার ব্যবহার করলে ট্রেনিং এবং টেস্টিং সময় কমে।
- **ইন্টারপ্রেটেবিলিটি:** কম ফিচার থাকলে মডেলের ফলাফল বোঝা এবং ব্যাখ্যা করা সহজ হয়।

এই অধ্যায়ে আমরা ফিচার সিলেকশনের যে যে টেকনিকগুলো শিখব--

1. Univariate Selection.
2. Recursive Feature Elimination.
3. Principle Component Analysis.
4. Feature Importance.

৬.১ ফিচার সিলেকশন

যে ফিচারগুলো টার্গেট ভ্যারিয়েবল প্রেডিক্ট করতে সাহায্য করে সেই ফিচারগুলোকে সিলেক্ট করাই হচ্ছে ফিচার সিলেকশন। তিনটি কারণে ফিচার সিলেকশন খুবই জরুরী।

ওভারফিটিং কমানোঃ ওভারফিটিং মানে হচ্ছে মেশিন ট্রেনিং ডেটার উপর এমন ভালোভাবে শিখে যাকে মুখস্ত করার মত বলতে পারি যে মেশিন ট্রেনিং ডেটার উপর ভালো পারফর্মেন্স করে কিন্তু টেস্ট ডেটার উপর পারফর্মেন্স খারাপ করে যা আমরা একমত চাই না । একই ধরনের ফিচার বারবার থাকলে এমনটা হয়। তাই ফিচার সিলেকশনের মাধ্যমে যদি ঐ ফিচার বাদ দেওয়া যায় তাহলে ওভারফিটিং কমবে।

মডেলের একুরেসি বাড়ায়ঃ কম অদরকারি ডেটা মানে মেশিন ভালো ভাবে শিখতে পারে অর্থাৎ মডেলের একুরেসি বেশি হয়।

মডেল ট্রেইন্ড হতে সময় কম লাগেঃ কম ডেটা মানে মেশিনকে ট্রেইন করাতে সময় কম লাগে।

৬.১.১ Univariate Selection

Statistical tests এর মাধ্যমে যে ফিচারগুলো টার্গেট ভ্যারিয়েবলকে প্রেডিক্ট করার জন্য দায়ী সেগুলোকে সিলেক্ট করা যায়। The scikit-learn লাইব্রেরির SelectKBest ক্লাস বিভিন্ন Statistical tests এর মাধ্যমে ফিচার সিকেকশন করে । আমরা এখানে ch2 Statistical tests এর মাধ্যমে বেস্ট ৪টা ফিচার সিলেক্ট করব

In [1]:
```
import pandas as pd
from numpy import set_printoptions
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2

#load dataset
dataframe = pd.read_csv('diabetes.csv')
array = dataframe.values
X = array[:,0:8]
Y = array[:,8]
# feature extraction
test = SelectKBest(score_func=chi2, k=4)
fit = test.fit(X, Y)
# summarize scores
set_printoptions(precision=3)
scores = list(fit.scores_)
print(fit.scores_)
features = fit.transform(X)
# summarize selected features
print(features[0:5,:])
```

```
[ 111.52  1411.887   17.605   53.108 2175.565  127.669    5.393  181.304]
[[148.    0.   33.6  50. ]
 [ 85.    0.   26.6  31. ]
 [183.    0.   23.3  32. ]
 [ 89.   94.   28.1  21. ]
 [137.  168.   43.1  33. ]]
```

In [4]:
```
feature_name = dataframe.columns
# i = 0
# for name in feature_name:
#     print(name, '=', scores[i])
#     i += 1
for name, score in zip(dataframe.columns[:8], scores):
    print(name, '=', score)
```

```
Pregnancies = 111.51969063588255
Glucose = 1411.887040644141
BloodPressure = 17.605373215320718
SkinThickness = 53.10803983632434
Insulin = 2175.5652729220137
BMI = 127.669343331037
DiabetesPedigreeFunction = 5.392681546971454
Age = 181.30368904430023
```

আমরা এখানে দেখতে পাচ্ছি বেস্ট ৪টা ফিচার সিলেকশন স্কোর হচ্ছে যথাক্রমে Glucose, Insulin, BMI, Age।

৬.১.২ Recursive Feature Elimination

Recursive Feature Elimination টেকনিক হচ্ছে মেশিন লার্নিং এলগরিদম দিয়ে মেশিনকে ট্রেইন্ড করিয়ে যে যে ফিচার বা ফিচারগুলো টার্গেট ভ্যারিয়েবলকে প্রেডিক্ট করতে সাহায্য করে তাদেরকে চিহ্নিত করে এব যে যে ফিচার বা ফিচারগুলো টার্গেট ভ্যারিয়েবলকে প্রেডিক্ট করতে কম সাহায্য করে তাদেরকেও চিহ্নিত করে একটি লিস্ট করা হয় তারপর কম সাহায্যকারী ফিচারগুলোকে বাদ দিয়ে দেওয়া হয় সেই লিস্ট থেকে বের করে দেওয়া হয় । এই কাজটি Recursivly হয় বলে একে Recursive Feature Elimination বলা হয়। scikit learn এ RFE ক্লাস ব্যবহার করে এই কাজটি করা যায়। আমরা এখানে এলগরিদম হিসেবে logistics regression ব্যবহার করব।

In [11]:
```
from sklearn.preprocessing import StandardScaler
from sklearn.feature_selection import RFE
from sklearn.linear_model import LogisticRegression

# Scale the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# feature extraction with RFE on scaled data
model = LogisticRegression()
rfe = RFE(model, n_features_to_select=3)
fit = rfe.fit(X_scaled, Y)

print("Num Features: %d" % fit.n_features_)
print("Selected Features: %s" % fit.support_)
print("Feature Ranking: %s" % fit.ranking_)
```

```
Num Features: 3
Selected Features: [ True  True False False False  True False False]
Feature Ranking: [1 1 3 6 5 1 2 4]
```

আমারে এখানে দেখতে পাচ্ছি RFE তিনটি ফিচার সিলেক্ট করেছে Pregnancies, BMI, Age

৬.১.৩ Principal Component Analysis (PCA)

Principal Component Analysis (or PCA) লিনিয়ার এলজেবরা ব্যবহার করে ডেটাসেটকে কম্প্রেস করে। একে ডেটা রিডাকশন টেকনিক বলা হয়ে থাকে। আমরা পরবর্তি কোন চ্যাপ্টারে এ ব্যপারে বিস্তারিত আলোচনা করব। এখানে PCA ব্যবহার করে ফিচার সিলেকশন দেখব। scikit learn এ PCA ক্লাস ব্যবহার করে ফিচার সিলেকশন টেকনিক ব্যবহার করা যায়। আমরা এখানে তিনটি দরকারি ফিচার সিলেক্ট করব।

In [12]:
```
from sklearn.decomposition import PCA
array = dataframe.values
X = array[:,0:8]
Y = array[:,8]
# feature extraction
pca = PCA(n_components=3)
pca.fit(X)
x_pca = pca.transform(X)
print("Before feature selection", X.shape)
print("After feature selection", x_pca.shape)
```

```
Before feature selection (768, 8)
After feature selection (768, 3)
```

আমরা দেখতে পাচ্ছি আমাদের ফিচার আগে ছিলো ৮ টি PCA করার পরার পর ফিচার হয়ে গেছে ৩ টি ।

৬.১.৪ Feature Importance

Bagged decision trees যেমন Random Forest and Extra Trees অথবা decission tree ব্যবহার করে গুরুতপূর্ন ফিচার বাঁচাই করতে পারি। আমরা এখানে ExtraTreesClassifier এবং DecisionTreeClassifier ব্যবহার ফিচার ইম্পোর্টেন্স দেখব।

In [13]:
```
from sklearn.ensemble import ExtraTreesClassifier
array = dataframe.values
X = array[:,0:8]
Y = array[:,8]
# feature extraction
model = ExtraTreesClassifier()
model.fit(X, Y)
for name, score in zip(dataframe.columns,model.feature_importances_ ):
    print(name, score)
```

```
Pregnancies 0.11147571694232372
Glucose 0.2296124041260251
BloodPressure 0.10188234169399393
SkinThickness 0.07945826922744019
Insulin 0.0756038696642209
BMI 0.1420388147069889
DiabetesPedigreeFunction 0.12231309018095063
Age 0.13761549345805665
```

আমরা এখানে বিভিন্ন ফিচারে স্কোর দেখতে পাচ্ছি। যে ফিচারের স্কোর বেশি তার গুরুত্ব বেশি। এখানে যথাক্রমে Glucose,BMI,Age এর গুরত্ব বেশি দেখতে পাচ্ছি।

In [14]:
```
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier(max_depth=2, random_state=42)
model.fit(X, Y)
for name, score in zip(dataframe.columns,model.feature_importances_ ):
    print(name, score)
```

```
Pregnancies 0.0
Glucose 0.6564199687624899
BloodPressure 0.0
SkinThickness 0.0
Insulin 0.0
BMI 0.19253848413716962
DiabetesPedigreeFunction 0.0
Age 0.15104154710034046
```

DecisionTreeClassifier এর মাধ্যমেও দেখতে পাচ্ছি Glucose,BMI,Age এর গুরত্ব বেশি।

৬.২ অধ্যায় শেষে যা যা শিখলাম

এই অধ্যায়ে মেশিন লার্নিং এর জন্য ডেটা প্রসেসিংয়ের আরেকটা ধাপ ফিচার সিলেকশন শিখে ফেললাম। আমরা ৪টা টেকনিক ব্যবহার করে ফিচার সিলেকশন করেছি --

1. Univariate Selection.
2. Recursive Feature Elimination.
3. Principle Component Analysis.
4. Feature Importance

In [4]:
```
```