In [3]:

PHITRON AI/ML

অধ্যায় ৪ঃ ডেটা এনালাইসিস (ভিজ্যুয়ালাইজেশন ব্যবহার করে ডেটা সম্পর্কে পরিষ্কার ধারণা নেওয়া)

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

Provide by: **Phitron AI/ML**

Community Link: **Phitron AI/ML**

মেশিন লার্নিং এলগরিদমগুলোতে মূলত ডেটা থেকে শেখে। তাই ডেটার ভেতরে কী আছে, সেটা বোঝা ছাড়া মডেল বানানো মানে চোখ বেঁধে হাঁটার মতো। আর ডেটাকে ভালোভাবে বোঝার সবচেয়ে কার্যকর উপায় হলো ডেটা ভিজ্যুয়ালাইজেশন।

মেশিন লার্নিং মডেল তৈরির আগে কেন ডেটা ভিজুয়ালাইজেশন করা দরকার

1. ডেটার কোয়ালিটি চেক করা

অনেক সময় ডেটাসেটে মিসিং ভ্যালু, আউটলায়ার (অস্বাভাবিক মান) বা ডুপ্লিকেট থাকে। টেবিল দেখে এগুলো ধরা কঠিন, কিন্তু বক্সপ্লট, হিস্টোগ্রাম বা স্ক্যাটারপ্লট ব্যবহার করলে সঙ্গে সঙ্গেই চোখে পড়ে। উদাহরণ: যদি কারও বয়স কলামে -5 বা 250 দেখা যায়, সেটা গ্রাফে সহজেই বোঝা যাবে।

2. ফিচার আর টার্গেটের সম্পর্ক বোঝা

ভিজ্যুয়ালাইজেশন ছাড়া বোঝা কঠিন কোন ভেরিয়েবল টার্গেট ভ্যালুর সাথে কতটা সম্পর্কিত। উদাহরণ: বাড়ির দাম প্রেডিক্ট করার আগে স্ক্যাটারপ্লট দেখে বোঝা যাবে—House Size আর Price এর মধ্যে পজিটিভ সম্পর্ক আছে।

3. ডেটার ডিস্ট্রিবিউশন দেখা

প্রতিটি ফিচারের মান কীভাবে ছড়ানো আছে (Normal, Skewed, Uniform) সেটা মডেলের জন্য গুরুত্বপূর্ণ। উদাহরণ: যদি কোনো ফিচারের ডেটা একদিকে বেশি ঝুঁকে থাকে (skewed), তাহলে আগে থেকে Normalization/Transformation করতে হবে।

4. আউটলায়ার শনাক্তকরণ

আউটলায়ার থাকলে মডেলের প্রেডিকশন একপাশে হেলে যেতে পারে। উদাহরণ: ইনকাম ডেটাতে যদি কিছু মানুষের আয় কোটি টাকার বেশি হয়, সেটা বক্সপ্লটে ধরা যাবে, এবং প্রয়োজনে হ্যান্ডেল করা যাবে।

5. সঠিক ফিচার সিলেকশন

সব ফিচার মডেলে ব্যবহার করা প্রয়োজন হয় না। কোন ফিচার ইনফরমেটিভ, আর কোনটা নয় সেটা ভিজ্যুয়ালাইজেশন করে বোঝা যায়। উদাহরণ: কাস্টমারের জুতার সাইজ যদি কেনাকাটার সাথে সম্পর্ক না রাখে, সেটা ফিচার থেকে বাদ দেওয়া যায়।

6. মডেলের একুরেসি বাড়ানো

ডেটা যদি ভালোভাবে বোঝা না হয়, তবে ভুলভাবে প্রিপ্রসেস করা হতে পারে, আর তাতে মডেল ভালো ফল দেবে না। ভিজ্যুয়ালাইজেশন এই ভুলগুলো কমায়।

এই অধ্যায় থেকে আমরা যা যা শিখব

এই অধ্যায়ে আমরা জানব কিভাবে পাইথনের বিভিন্ন লাইব্রেরি ব্যবহার করে ডেটা ভিজ্যুয়ালাইজেশন করতে হয় এবং ডেটা সম্পর্কে ধারনা নিতে হয়। এই অধ্যায়ে আমরা যা যা শিখব ---

1. ইউনিভ্যারিয়েট প্লট (Univariate Plots)

   ক) হিস্টোগ্রাম প্লট (Histogram plots)

   খ) ডেনসিটি প্লট (Density plots)

   গ) বক্স এবং হুইস্কার প্লট (Box & Whisker Plots)

2. মাল্টি-ভ্যারিয়েট প্লট (Multivariate Plots)

   ক) কোরিলেশন ম্যাট্রিক্স প্লট (Correlation metrix plots)

   খ) স্ক্যাটার প্লট ম্যাট্রিক্স (Scatter Plot Matrix)

৪.১ ইউনিভ্যারিয়েট প্লট (Univariate Plots)

Univariate Plots হচ্ছে প্রতিটা ফিচারকে আলাদা আলাদাভাবে প্লট করা। অর্থাৎ একটি লেখচিত্রে শুধুমাত্র একটি ভ্যারিয়েবলের ডেটাগুলো প্লট হবে। Univariate Plots এর মাধ্যমে প্রতিটা ফিচারের ডেটা ডিস্ট্রিবিউশন সম্পর্কে ধারনা পাওয়া যায়। ডেটার বিস্তার এবং কোন কোন ডেটাগুলো outlier অর্থাৎ দলছুট ভ্যালু তা জানা যায়। আউটলায়ার ভ্যালু ডেটার পুরো সামারাইজেশন চেঞ্জ করে দেয়। তাই আউটলায়ার ভ্যালুগুলোকে ডেটা থেকে ফেলে দিতে হবে।

আউটলায়ার (Outlier) বলতে এমন ডেটা পয়েন্ট বা মান বোঝায় যেগুলো ডেটাসেটের অন্যান্য মানের তুলনায় উল্লেখযোগ্যভাবে আলাদা বা অস্বাভাবিক। এগুলো সাধারণ প্যাটার্ন বা ডিস্ট্রিবিউশন থেকে অনেক দূরে অবস্থান করে যেমনঃ মনে করি ক্লাস ৫ এ পড়া একটা ছাত্রের বয়স ১০ থেকে ১১ বছরের মধ্যে হবে এখন যদি সেই ক্লাসে ৩০ বছরের এক আদুভাই পড়ে তাহলে এই আদুভাই হচ্ছে outlier।

আমার এখানের তিনটি Univariate Plots টেকনিক নিয়ে আলোচনা করব --

ক) হিস্টোগ্রাম প্লট (Histogram plots)

খ) ডেনসিটি প্লট (Density plots)

গ) বক্স এবং হুইস্কার প্লট (Box & Whisker Plots)

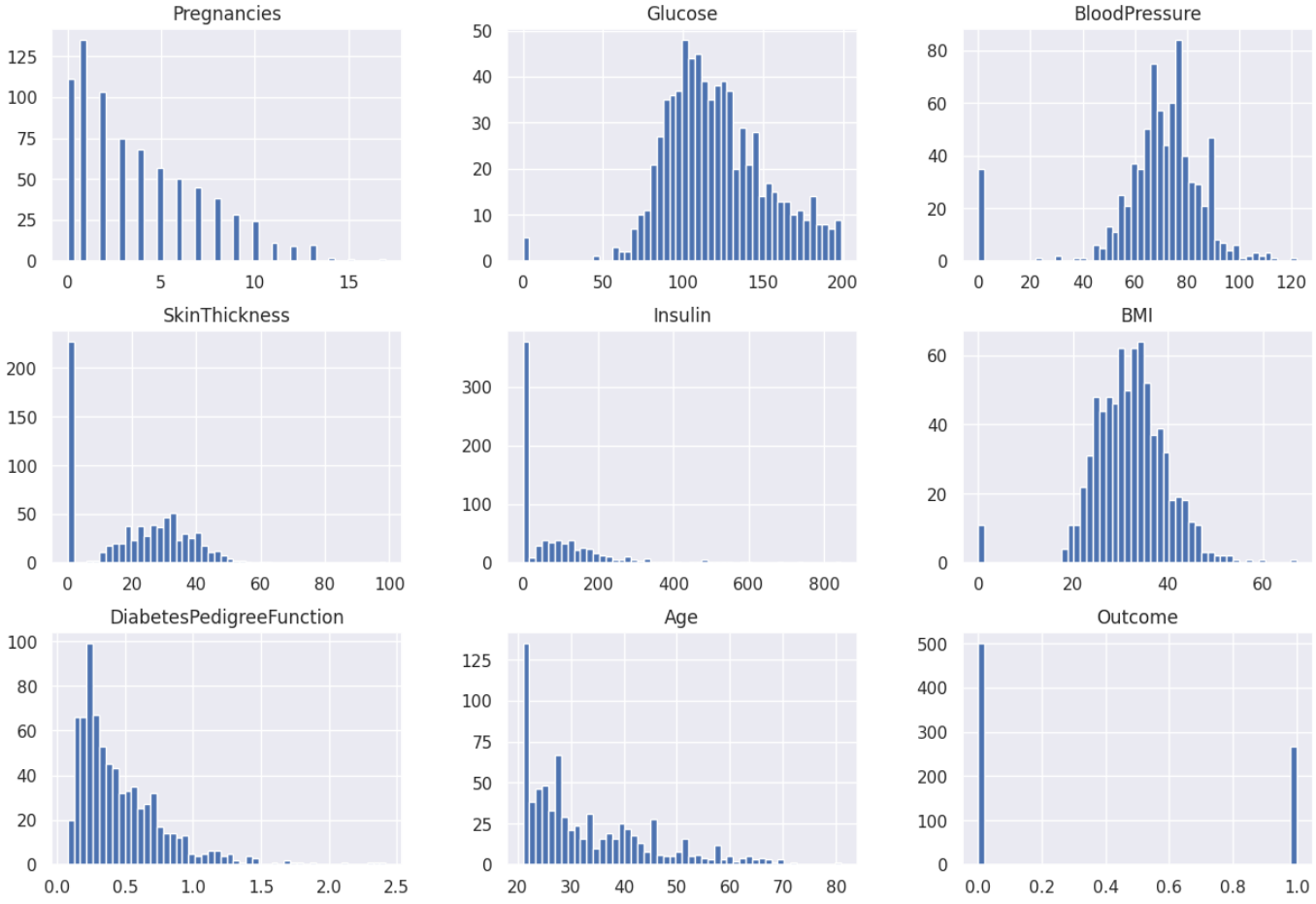
৪.১.১ হিস্টোগ্রাম প্লট (Histogram plots)

প্রতিটা ফিচারের ডিস্ট্রিবিউশন আলাদা আলাদাভাবে দেখার জন্য হিস্টোগ্রাম প্লট (Histogram plots) একটি ভালো টেকনিক। হিস্ট্রোগ্রাম গ্রুপ ডেটাকে একটা Bin এ নিয়ে আসে এবং প্রতিটা bin এর কয়টা অবজারভেশন আছে তা কাউন্ট করে প্লট করে Bar আকারে। এই bar এর আকার থেকে বুঝা যায় এই ফিচারের ডেটা ডিস্ট্রিবিউশন কোন ধরনের (নারমাল ডিস্ট্রিবিউশন, skew, এক্সপোনেনশিয়াল (exponential)।

Pandas DataFrame এর মাধ্যমে ডেটা ভিজ্যুয়ালাইজেশন করা যায়। hist() ফাংশন ব্যবহার করে হিস্টোগ্রাম প্লট (Histogram plots) করা যায় ।

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
dataset = pd.read_csv('diabetes.csv')
sns.set(style='darkgrid')
dataset.hist(bins=50, figsize = (15, 10))
plt.show()
```

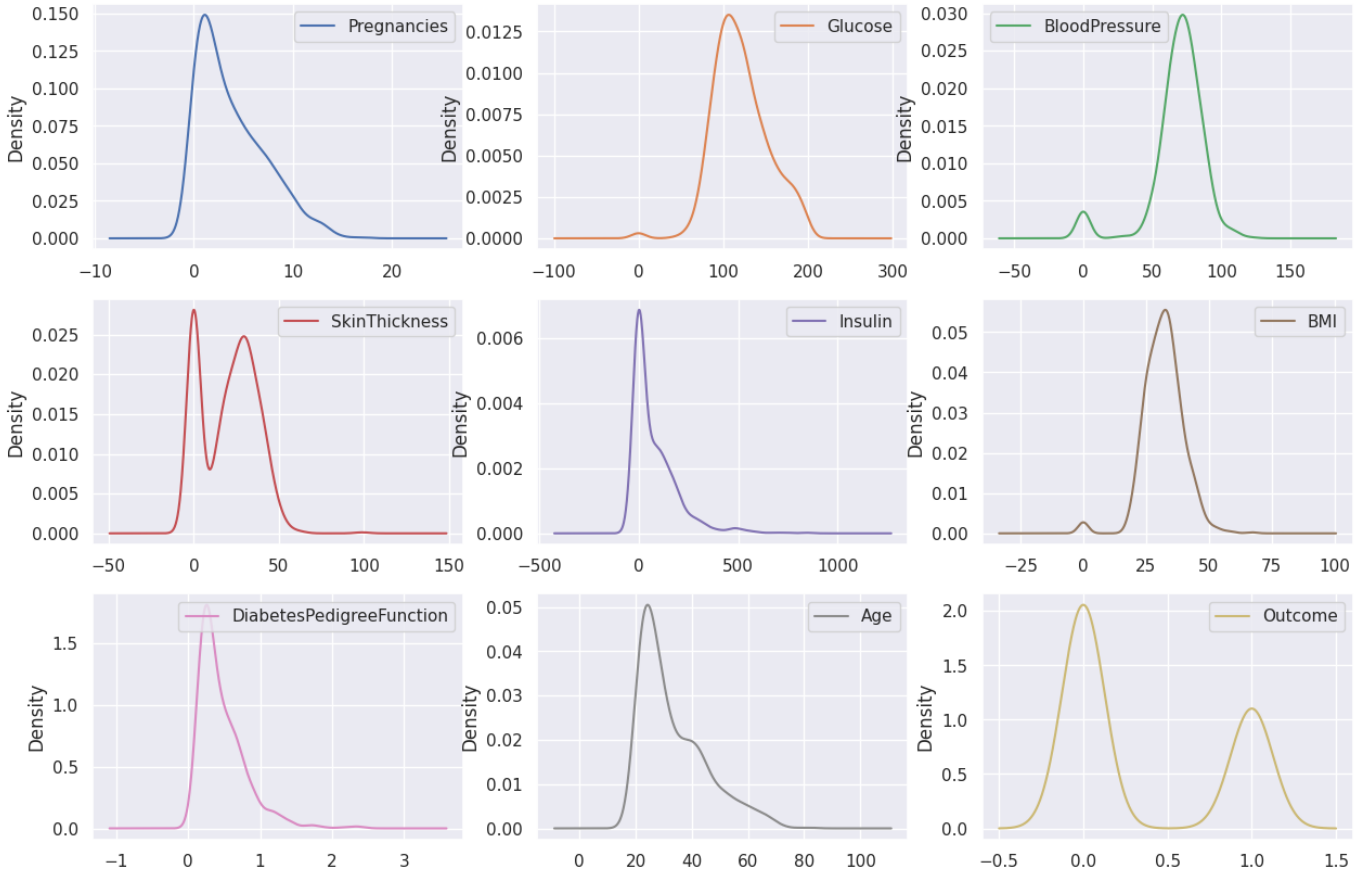


আমরা histogram থেকে দেখতে পাচ্ছি age, DiabetesPedigreeFunction, Insulin, Pregancies এর ডেটা ডিস্ট্রিবিউশন হচ্ছে exponential distribution। এবং BMI, Blood pressure, Glucose এর ডেটা ডিস্ট্রিবিউশন নরমাল / গসিয়ান ডিস্ট্রিবিউশন বা এর কাছাকাছি ডিস্ট্রিবিউশন।

৪.১.২ ডেনসিটি প্লট (Density plots)

ডেটা ডিস্ট্রিবিউশন সম্পর্কে জানার জন্য আরেকটি ভালো টেকনিক হচ্ছে ডেনসিটি প্লট (Density plots)। পাইথনে কিভাবে ডেনসিটি প্লট (Density plots) আঁকতে হয় দেখা যাক

In [4]:
```
dataset.plot(kind='kde', subplots=True, layout=(3,3), sharex=False, figsize=(15, 10))
plt.show()
```



histogram plot থেকে density plot এ ডেটা ডিস্ট্রিবিউশন আরো ভালোভাবে দেখা যাচ্ছে। আমরা Density plot থেকে দেখতে পাচ্ছি age, DiabetesPedigreeFunction, Insulin, Pregancies এর ডেটা ডিস্ট্রিবিউশন হচ্ছে exponential distribution। এবং BMI, Blood pressure, Glucose এর ডেটা ডিস্ট্রিবিউশন নরমাল / গসিয়ান ডিস্ট্রিবিউশন বা এর কাছাকাছি ডিস্ট্রিবিউশন।

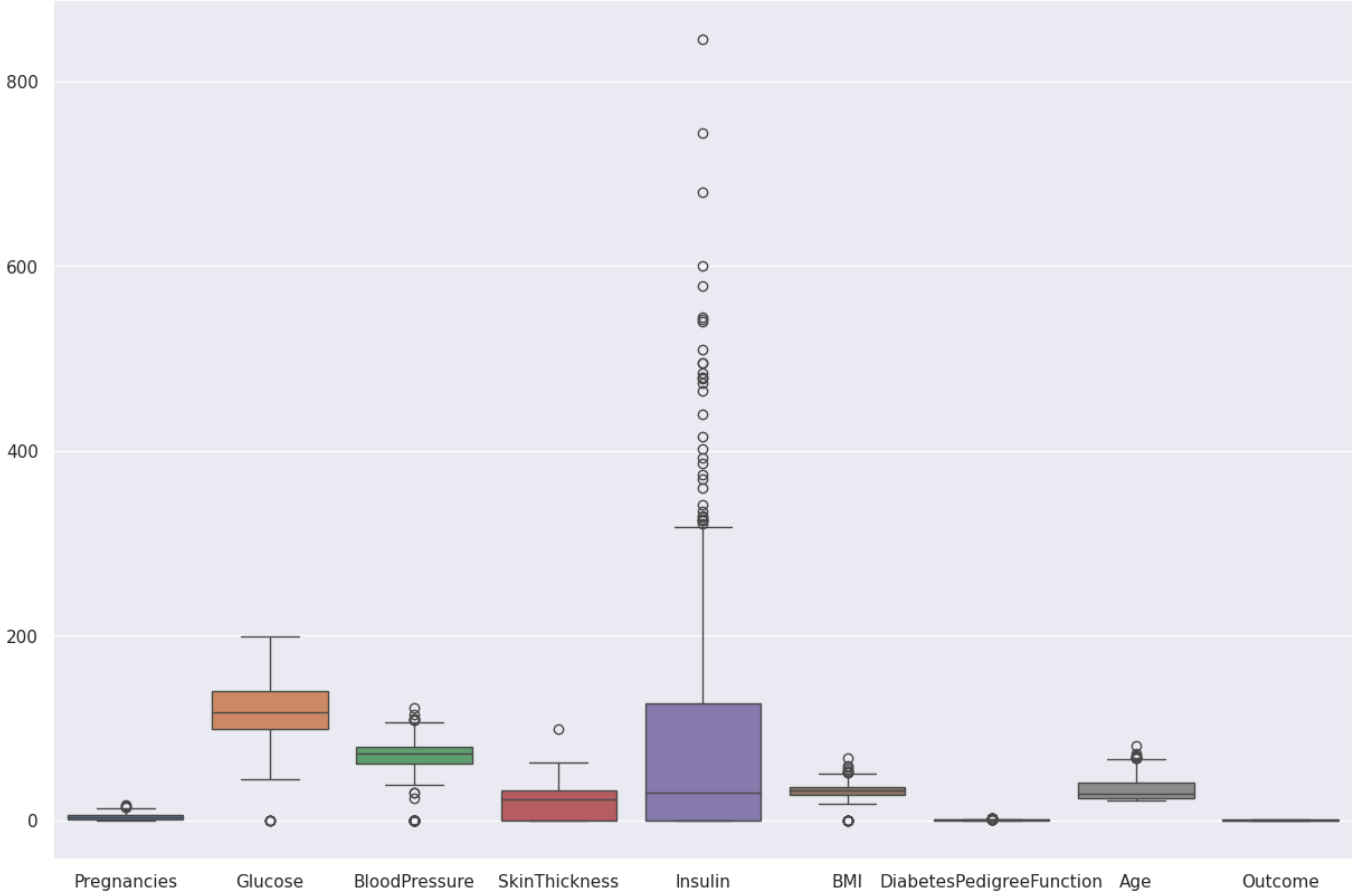
৪.১.৩ বক্স এবং হুইস্কার প্লট (Box and Whisker Plots)

বক্স এবং হুইস্কার প্লটের মাধ্যমে ডেটার বিস্তার সম্পর্কে ধারনা পাওয়া যায়। বক্সপ্লটের মাধ্যমে ডেটার skew সম্পর্কেও পরিষ্কার ধারনা পাওয়া যায়। ডেটার outlier অর্থাৎ দলছুট ভ্যালু সম্পর্কে স্পষ্ট ধারনা পাওয়া যায়। আমরা এখানে cufflinks এর মাধ্যমে বক্সপ্লট আঁকব।

In [6]:
```
# import cufflinks as cf
# cf.go_offline()

# fig = dataset.iplot(asFigure=True, kind="box")
# fig.show()

# Use seaborn for boxplot instead due to potential compatibility issues with cufflink
plt.figure(figsize=(15, 10))
sns.boxplot(data=dataset)
plt.show()
```



আমরা বক্স অ হুইস্কার প্লট থেকে দেখতে পাচ্ছি ইনসুলিনের ডেটাগুলো বিস্তার অনেক বেশি বিশেষ করে q3 থেকে ম্যাক্সিমাম পর্যন্ত। এবং অন্যান্য ফিচারের বক্স প্লট থেকে ডেটার skew সম্পর্কে একটা ধারনা পাওয়া যাচ্ছে। অধ্যায় ২ এ বক্স এবং হুইস্কার প্লট নিয়ে বিস্তারিত আলোচনা করা হয়েছে।

৪.২ মাল্টি-ভ্যারিয়েট প্লট (Multivariate Plots)

মাল্টি-ভ্যারিয়েট প্লট (Multivariate Plots) এর মাধ্যমে ভ্যারিয়েবলদের মধ্যকার সম্পর্ক জানা জানা। আমারা এখানে দুই ধরনের প্লট নিয়ে আলোচনা করব।

ক) কোরিলেশন ম্যাট্রিক্স প্লট (Correlation metrix plots)

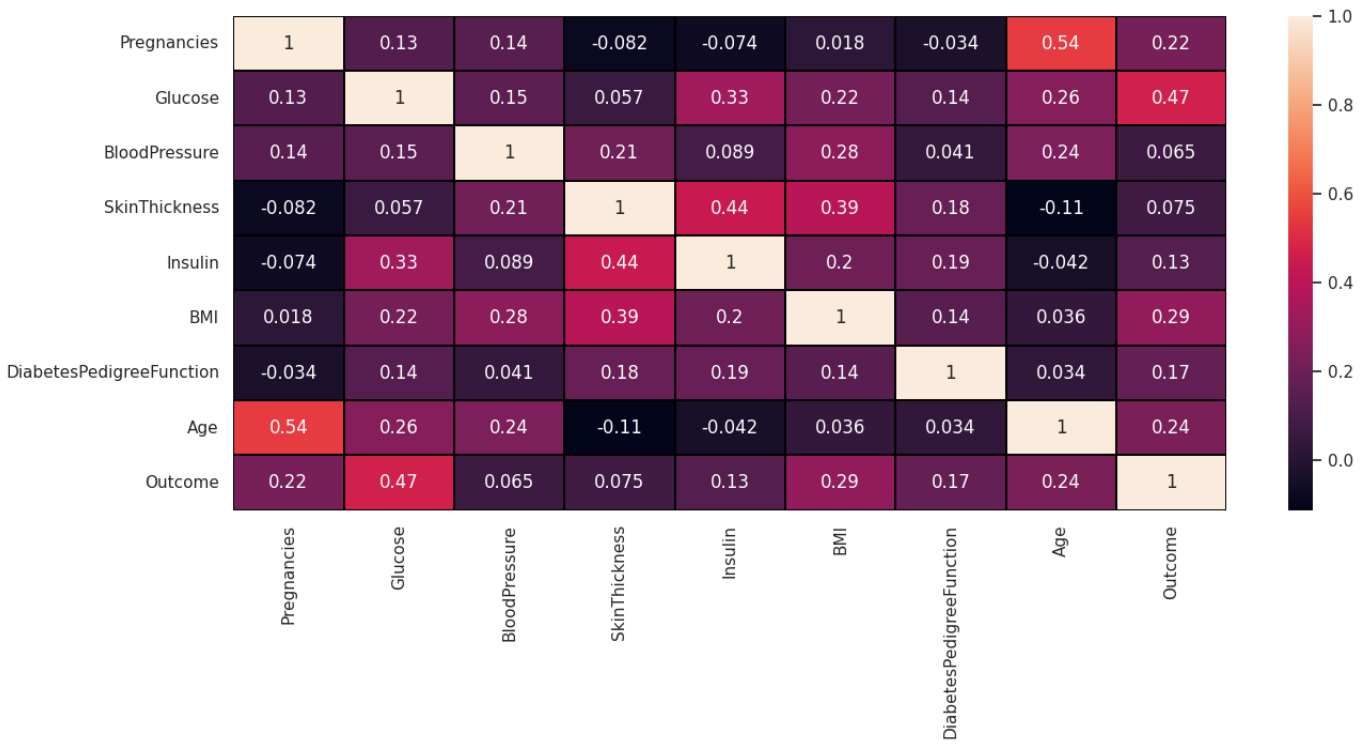
খ) স্ক্যাটার প্লট ম্যাট্রিক্স (Scatter Plot Matrix)

৪.২.১ কোরিলেশন ম্যাট্রিক্স প্লট (Correlation metrix plots)

কোরিলেশন হচ্ছে একটি ভ্যারিয়েবলের পরিবর্তনের ফলে আরেকটি ভ্যালু কিরূপ পরিবর্তন হয় । এই সম্পর্কে অধ্যায় দুইইয়ে বিস্তারিত আলোচনা করা হয়েছে। ফিচারগুলোর মধ্যে কোরিলেশন জানা জরুরী কেননা ভ্যারিয়েবলগুলোর মধ্যে যদি highly correlation থাকে তাহলে কিছু কিছু মেশিন লার্নিং এলগোরিদম (linear & Logistic regression) আছে যেগুলো ডেটা থেকে ভালো ভালো শিখতে পারে না। আমরা seaborn ব্যবহার করে Correlation metrix plots আঁকব।

In [7]:
```
corr = dataset.corr()
fig = plt.figure(figsize=(15,6))
sns.heatmap(corr, linecolor='black', linewidths=.1, annot=True)
```

Out[7]: <Axes: >



আমরা Correlation metrix plots থেকে দেখতে পাচ্ছি এট্রিবিউটগুলোর মধ্যে(নিজের সাথে ছাড়া) highly correlation নেই। highly correlation থাকলে খয়েরি কালার হতো।

৪.২.২ স্ক্যাটার প্লট ম্যাট্রিক্স (Scatter Plot Matrix)

স্ক্যাটার প্লটের মাধ্যমে দুটি নিউমেরিক্যাল ভ্যারিয়েবলের মধ্যে কোরিলেশন এবং আউটলায়ার ভ্যালুগুলোকে দেখা যায়। ডেটা সেটের প্রতিটা ভ্যারিয়েবলের সাথে অপর আরেকটি ভ্যারিয়েবলের স্ক্যাটার প্লট করে তাদের মধ্যে সম্পর্ক দেখা যায়। এই সবগুলো প্লট যখন একসাথে দেখানো হয় তখন তাকে স্ক্যাটার প্লট ম্যাট্রিক্স (Scatter Plot Matrix) বলে। আমরা seaborn ব্যবহার করে স্ক্যাটার প্লট ম্যাট্রিক্স (Scatter Plot Matrix) তৈরি করব।

In [8]:
```
sns.pairplot(data=dataset, hue='Outcome')
```

Out[8]:
```
<seaborn.axisgrid.PairGrid at 0x7ffa5e03b380>
```

আমরা এখানে দেখতে পাচ্ছি ভ্যারিয়েবলগুলোর মধ্যে কোন স্ট্রং কোরিলেশন নেই।

৪.৩ এই অধ্যায়ে আমরা যা যা শিখেছি

1. ইউনিভ্যারিয়েট প্লট (Univariate Plots)

ক) হিস্টোগ্রাম প্লট (Histogram plots)

খ) ডেনসিটি প্লট (Density plots)

গ) বক্স এবং হুইস্কার প্লট (Box & Whisker Plots) ২) মাল্টি-ভ্যারিয়েট প্লট (Multivariate Plots)

ক) কোরিলেশন ম্যাট্রিক্স প্লট (Correlation metrix plots)

খ) স্ক্যাটার প্লট ম্যাট্রিক্স (Scatter Plot Matrix)

In [4]: