Linear Regression on Diabetes dataset

In [9]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [10]:

```
pdata= pd.read_csv('C:\\Users\\Jai mata di\\Downloads\\diabetes.csv')
pdata.head()
```

Out[10]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunctio
0	6	148	72	35	0	33.6	0.62
1	1	85	66	29	0	26.6	0.35
2	8	183	64	0	0	23.3	0.67
3	1	89	66	23	94	28.1	0.16
4	0	137	40	35	168	43.1	2.28

In [11]:

```
pdata.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Pregnancies               768 non-null    int64  
 1   Glucose                   768 non-null    int64  
 2   BloodPressure             768 non-null    int64  
 3   SkinThickness             768 non-null    int64  
 4   Insulin                   768 non-null    int64  
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64  
 8   Outcome                   768 non-null    int64  
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

In [30]:

```
class_counts = pdata.groupby('Outcome').size()
print("Class breakdown of the data:\n")
print(class_counts)
```

```
Class breakdown of the data:

Outcome
0    500
1    268
dtype: int64
```

In [12]:

```
pdata.corr()
```

Out[12]:

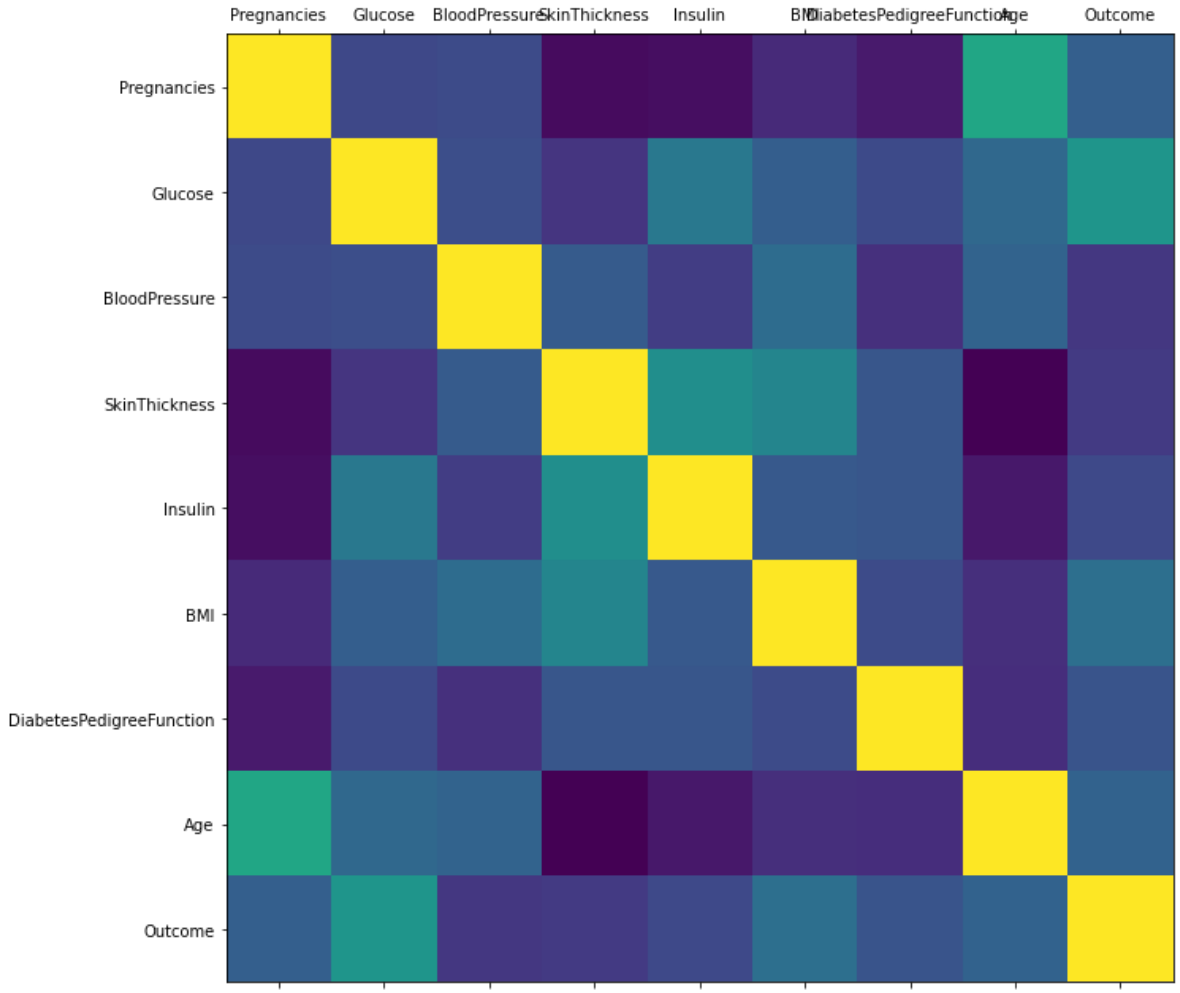
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	
Pregnancies	1.000000	0.129459	0.141282	-0.081672	-0.073535	0.0
Glucose	0.129459	1.000000	0.152590	0.057328	0.331357	0.2
BloodPressure	0.141282	0.152590	1.000000	0.207371	0.088933	0.2
SkinThickness	-0.081672	0.057328	0.207371	1.000000	0.436783	0.3
Insulin	-0.073535	0.331357	0.088933	0.436783	1.000000	0.1
BMI	0.017683	0.221071	0.281805	0.392573	0.197859	1.0
DiabetesPedigreeFunction	-0.033523	0.137337	0.041265	0.183928	0.185071	0.1
Age	0.544341	0.263514	0.239528	-0.113970	-0.042163	0.0
Outcome	0.221898	0.466581	0.065068	0.074752	0.130548	0.2

In [17]:

```
# However we want to see correlation in graphical representation so below is function
def plot_corr(pdata, size=11):
        corr = pdata.corr()
        fig, ax = plt.subplots(figsize=(size, size))
        ax.matshow(corr)
        plt.xticks(range(len(corr.columns)), corr.columns)
        plt.yticks(range(len(corr.columns)), corr.columns)
```

In [18]:

```
plot_corr(pdata)
```



In [22]:

```
# from sklearn.model_selection import train_test_split
X = pdata.drop('Outcome',axis=1) # Predictor feature columns (8 X m)
Y = pdata['Outcome'] # Predicted class (1=True, 0=False) (1 X m)
x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.3, random_state=1)
# 1 is just any random seed number
x_train.head()
```

Out[22]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunct
88	15	136	70	32	110	37.1	0.
467	0	97	64	36	100	36.8	0.
550	1	116	70	28	0	27.4	0.
147	2	106	64	35	119	30.5	1.
481	0	123	88	37	0	35.2	0.

In [23]:

```
print(x_train.shape)
```

```
(537, 8)
```

In [24]:

```
print(y_train.shape)
```

```
(537,)
```

In [31]:

```
skew = pdata.skew()
print("Skew of attribute distributions in the data:\n")
skew
```

```
Skew of attribute distributions in the data:
```

Out[31]:

```
Pregnancies                0.901674
Glucose                    0.173754
BloodPressure             -1.843608
SkinThickness              0.109372
Insulin                    2.272251
BMI                       -0.428982
DiabetesPedigreeFunction   1.919911
Age                        1.129597
Outcome                    0.635017
dtype: float64
```

Visualizing data using Pandas

Now, we visualize data using Python's Pandas library. We discuss both univariate plots and multivariate plots using Pandas.

Univariate plots:

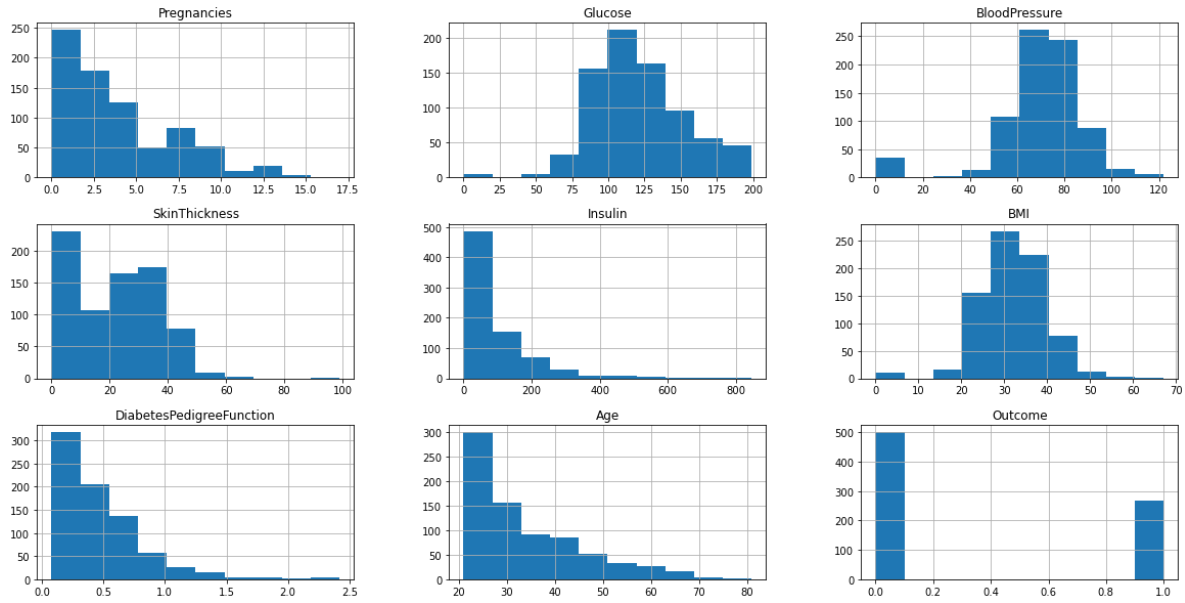
Histograms. Density Plots. Box and Whisker Plots.

In [32]:

```
# Import required package
from matplotlib import pyplot
pyplot.rcParams['figure.figsize'] = [20, 10] # set the figure size
```

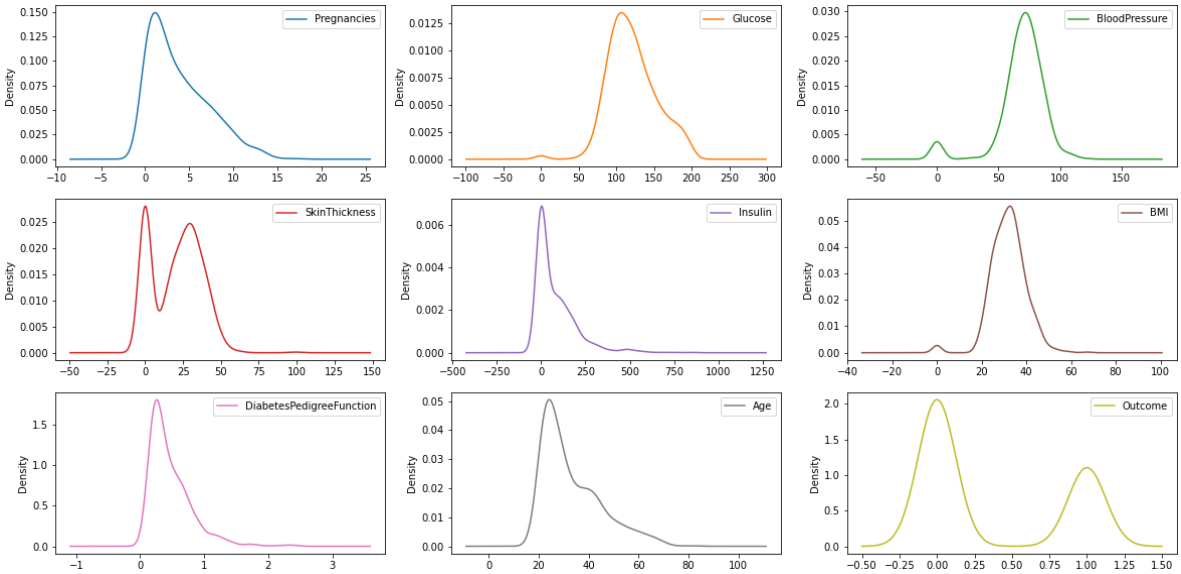
In [33]:

```
# Draw histograms for all attributes
pdata.hist()
pyplot.show()
```



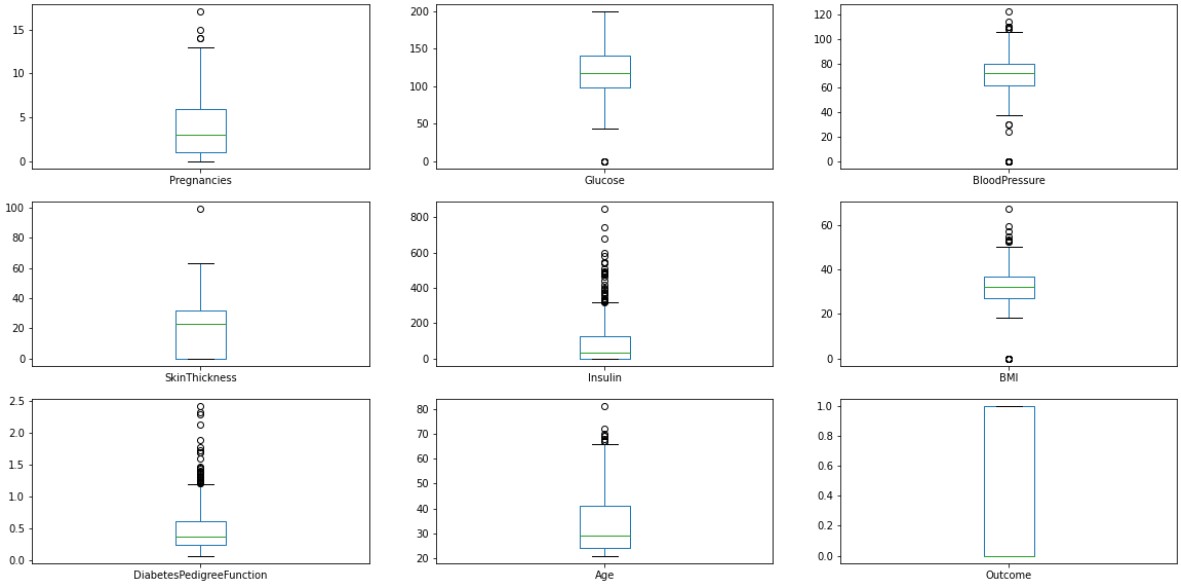
In [34]:

```
# Density plots for all attributes
pdata.plot(kind='density', subplots=True, layout=(3,3), sharex=False)
pyplot.show()
```



In [35]:

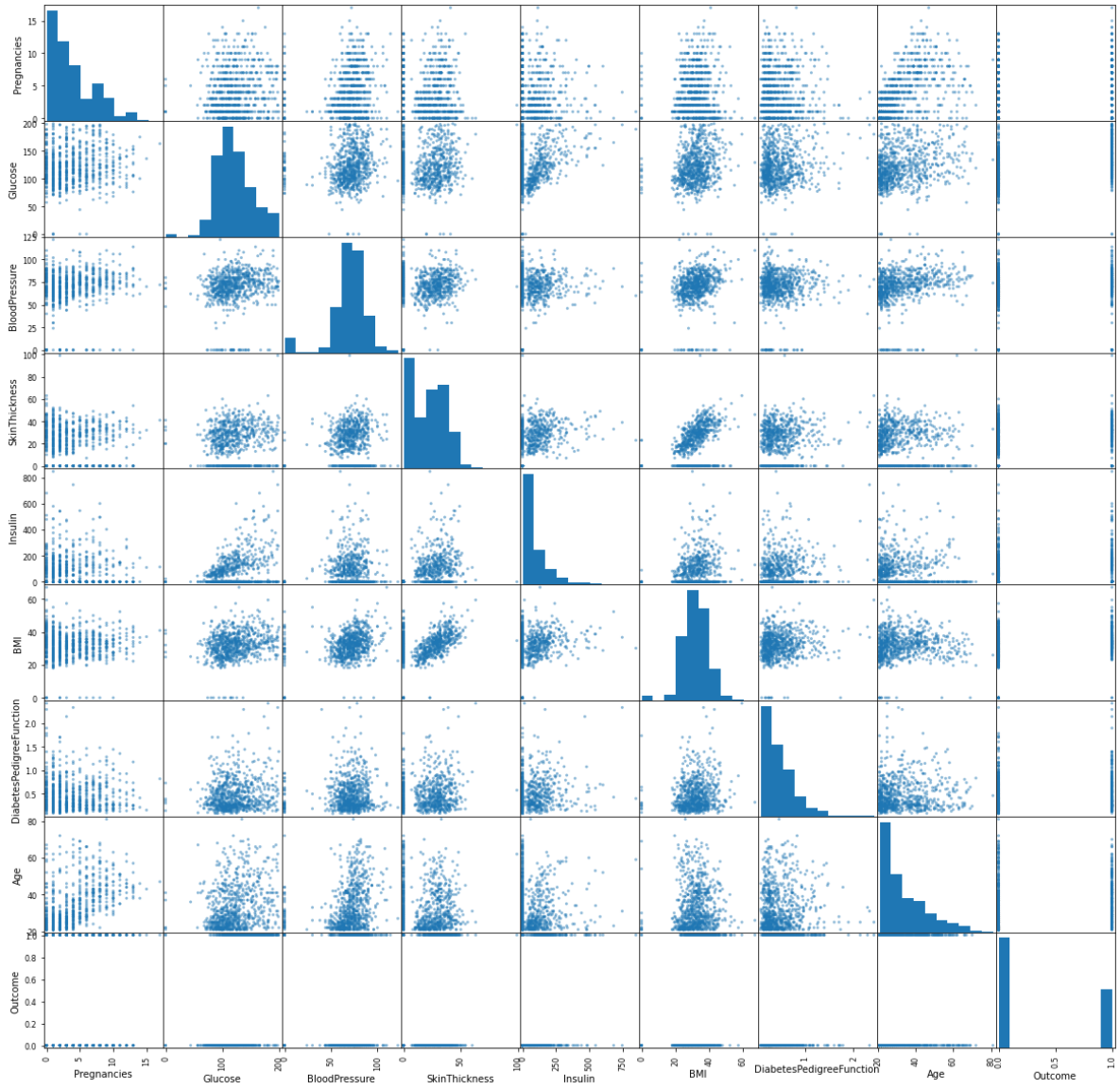
```
# Draw box and whisker plots for all attributes
pdata.plot(kind= 'box', subplots=True, layout=(3,3), sharex=False, sharey=False)
pyplot.show()
```



In [36]:

```
# Import required package
from pandas.plotting import scatter_matrix
pyplot.rcParams['figure.figsize'] = [20, 20]

# Plotting Scatterplot Matrix
scatter_matrix(pdata)
pyplot.show()
```



In [ ]: