

Generate

print hello world using rot13



Close

```
import pandas as pd
df=pd.read_csv("/content/archive (10) (1).zip")
print(df)
```

```
↗
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI   \
0             6      148             72             35         0   33.6
1             1       85             66             29         0   26.6
2             8      183             64             0         0   23.3
3             1       89             66             23        94   28.1
4             0      137             40             35       168   43.1
..          ...      ...             ...             ...       ...   ...
763           10      101             76             48       180   32.9
764             2      122             70             27         0   36.8
765             5      121             72             23       112   26.2
766             1      126             60             0         0   30.1
767             1       93             70             31         0   30.4
```

```

      DiabetesPedigreeFunction  Age  Outcome
0                0.627      50         1
1                0.351      31         0
2                0.672      32         1
3                0.167      21         0
4                2.288      33         1
..          ...      ...      ...
763           0.171      63         0
764           0.340      27         0
765           0.245      30         0
766           0.349      47         1
767           0.315      23         0
```

[768 rows x 9 columns]

```
column_means = df.mean()
print("Mean of each column:")
print(column_means)
```

```
↗ Mean of each column:
Pregnancies           3.845052
Glucose              120.894531
BloodPressure        69.105469
SkinThickness        20.536458
Insulin              79.799479
BMI                 31.992578
DiabetesPedigreeFunction  0.471876
Age                 33.240885
Outcome              0.348958
dtype: float64
```

```
from sklearn.preprocessing import MinMaxScaler, StandardScaler
numerical_cols = df.columns[df.dtypes != 'object'].tolist()
numerical_cols.remove('Outcome')
```

```
min_max_scaler = MinMaxScaler()
df_minmax = df.copy()
df_minmax[numerical_cols] = min_max_scaler.fit_transform(df[numerical_cols])
```

```
print("Dataset after Min-Max Normalization:")
print(df_minmax.head())
```

```
standard_scaler = StandardScaler()
df_standardized = df.copy()
df_standardized[numerical_cols] = standard_scaler.fit_transform(df[numerical_cols])
```

```
print("\nDataset after Standardization:")
print(df_standardized.head())
```

```
↗ Dataset after Min-Max Normalization:
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI   \
0    0.352941  0.743719    0.590164    0.353535  0.000000  0.500745
1    0.058824  0.427136    0.540984    0.292929  0.000000  0.396423
2    0.470588  0.919598    0.524590    0.000000  0.000000  0.347243
3    0.058824  0.447236    0.540984    0.232323  0.111111  0.418778
4    0.000000  0.688442    0.327869    0.353535  0.198582  0.642325

      DiabetesPedigreeFunction  Age  Outcome
0    0.234415  0.483333         1
1    0.116567  0.166667         0
2    0.253629  0.183333         1
3    0.038002  0.000000         0
4    0.943638  0.200000         1
```

Dataset after Standardization:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	\
0	0.639947	0.848324	0.149641	0.907270	-0.692891	0.204013	
1	-0.844885	-1.123396	-0.160546	0.530902	-0.692891	-0.684422	
2	1.233880	1.943724	-0.263941	-1.288212	-0.692891	-1.103255	
3	-0.844885	-0.998208	-0.160546	0.154533	0.123302	-0.494043	
4	-1.141852	0.504055	-1.504687	0.907270	0.765836	1.409746	

	DiabetesPedigreeFunction	Age	Outcome
0	0.468492	1.425995	1
1	-0.365061	-0.190672	0
2	0.604397	-0.105584	1
3	-0.920763	-1.041549	0
4	5.484909	-0.020496	1