Phase 3: Development Part 1

Developing a diabetes prediction system involves several key steps, including data preparation and feature selection. Here's a step-by-step guide to help you get started:

1.Importing the necessary packages:

```
import pandas as pd
from sklearn.model_selection import train_test_split
```

Pandas-Pandas is a Python library used for working with data sets.
sklearn-Scikit-Learn, also known as sklearn is a python library to implement machine learning models and statistical modelling.

2.Loading the dataset:

```
df=pd.read_csv('/content/diabetes.csv')
```

Read_csv-read_csv is a method in pandas module, which is used to read the csv files.

3.Exploratory Data Analysis:

```
df.head()
```

```
[10] df.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

```
df.shape
```

```
[12] df.shape
```

```
(768, 9)
```

```
df.info
```

```
df.info

<bound method DataFrame.info of      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]>
```

```
df.describe
```

```
df.describe

<bound method NDFrame.describe of      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]>
```

4.Separating Dataset into X and Y:

```
X=data.drop('Outcome',axis=1)
Y=data['Outcome']
```

X-Which doesn't store the 'outcome' field

Y-It stores only 'Outcome' field

5.Checking for Null values

```
print(data.isnull().sum())
```

```
print(data.isnull().sum())
```

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

There is no null values in the dataset.

6. Spliting dataset into test and training data:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
[24] X_train
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
60	2	84	0	0	0	0.0	0.304	21
618	9	112	82	24	0	28.2	1.282	50
346	1	139	46	19	83	28.7	0.654	22
294	0	161	50	0	0	21.9	0.254	65
231	6	134	80	37	370	46.2	0.238	46
...	...	...	...	...	...	...	...	...
71	5	139	64	35	140	28.6	0.411	26
106	1	96	122	0	0	22.4	0.207	27
270	10	101	86	37	0	45.6	1.136	38
435	0	141	0	0	0	42.4	0.205	29
102	0	125	96	0	0	22.5	0.262	21

614 rows × 8 columns