Significance Of Exploratory Data Analysis (EDA) for the AI-BASED DIABETES PREDICTION SYSTEM

Data Overview

Code:

```
diabetes_data.sample(9)
diabetes_data.tail(9)
```

This code displays random samples and the last 9 rows of your diabetes dataset. It helps you quickly view the structure of your data and understand its content.

Checking for Missing Values

Code:

```
diabetes_data.isnull()

diabetes_data.isnull().sum()
```

Data Information

Code:

```
diabetes_data.info()

diabetes_data.dtypes

diabetes_data.columns
```

These code snippets provide an overview of your diabetes dataset. diabetes_data.info() gives information about data types, non-null values, and memory usage. diabetes_data.dtypes shows the data types of each column and diabetes_data. columns lists the column names.

Duplicates Removal

Code:

```
diabetes_data.duplicated().sum()

diabetes_data = diabetes_data.drop_duplicates()

diabetes_data.duplicated().sum()
```

The code first checks for and reports the number of duplicate rows in the diabetes dataset. Then, it removes the duplicates using drop_duplicates(). This ensures your data contains unique records, avoiding data inconsistencies.

Scatter Plot

Code:

```
plt.scatter(diabetes_data['Glucose'], diabetes_data['BloodPressure'])

plt.xlabel('Glucose')

plt.ylabel('Blood Pressure')

plt.title('Scatter Plot: Glucose vs. Blood Pressure')

plt.show()
```

This code creates a scatter plot between "Glucose" and "BloodPressure". It helps visualize the relationship between glucose levels and blood pressure.

Box Plot

Code:

```
plt.boxplot(diabetes_data['Age'], vert=False)

plt.xlabel('Age')

plt.title('Box Plot: Age Distribution')

plt.show()
```

This box plot visualizes the distribution of ages in the diabetes dataset. It shows the median, quartiles, and potential outliers.

Histograms

Code

```
plt.hist(diabetes_data['glucose_level'], bins=20, edgecolor='k')

plt.hist(diabetes_data['insulin_level'], bins=20, edgecolor='k')
```

These snippets generate histograms to visualize glucose and insulin levels' distribution. Histograms provide insights into data distribution.

Bar Charts

Code:

```
outcome_counts = diabetes_data['Outcome'].value_counts()

plt.bar(outcome_counts.index, outcome_counts.values)

plt.xlabel('Outcome')

plt.ylabel('Count')

plt.title('Bar Chart: Outcome Distribution')

plt.show()
```

This code creates a bar chart to display the distribution of the 'Outcome' variable. It helps understand the distribution of outcomes in the dataset.

Tweet Length Distribution

Code:

```
plt.hist(diabetes_data['tweet_length'], bins=20)

plt.xlabel('Tweet Length')

plt.ylabel('Frequency')

plt.title('Tweet Length Distribution')

plt.show()
```

This code creates a histogram to visualize the distribution of tweet lengths in your dataset. This might be applicable if there is text data associated with the diabetes records.

Top Hashtags and Mentions

Code:

```
top_hashtags = diabetes_data['hashtags'].value_counts().head(10)

top_mentions = diabetes_data['mentions'].value_counts().head(10)

plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)

top_hashtags.plot(kind='bar', title='Top Hashtags')

plt.subplot(1, 2, 2)
```

```
top_mentions.plot(kind='bar', title='Top Mentions')

plt.tight_layout()

plt.show()
```

These snippets display bar charts of the top hashtags and mentions in the tweets, allowing you to identify popular topics and user mentions.