ALGORITHM:

Creating an AI-based diabetes prediction system using the provided dataset involves several steps, including data preprocessing, model building, and evaluation. Here's a Python-based algorithm using a Random Forest classifier for this purpose:

```
# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score

# Step 1: Data Loading
data = pd.read_csv('diabetes_data.csv')  # Load the dataset from the provided link

# Step 2: Data Preprocessing
# Separate features (X) and target variable (y)
X = data.drop('Outcome', axis=1)
y = data['Outcome']
```

```
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Standardize/normalize the features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Step 3: Model Building (Random Forest Classifier)
clf = RandomForestClassifier(random_state=42)
clf.fit(X_train, y_train)

# Step 4: Model Evaluation
# Make predictions on the test set
y_pred = clf.predict(X_test)

# Calculate evaluation metrics
accuracy = accuracy_score(y_test, y_pred)
```

```
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
roc_auc = roc_auc_score(y_test, clf.predict_proba(X_test)[:, 1])

# Step 5: Display Results
print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1 Score:", f1)
print("ROC AUC Score:", roc_auc)

# Optionally, save the trained model for future use
import joblib
joblib.dump(clf, 'diabetes_prediction_model.pkl')
```