# KNN Classifier with Distance Metrics Evaluation

## Overview

This Jupyter Notebook evaluates the performance of a K-Nearest Neighbors (KNN) classifier using three different distance metrics:

- Euclidean Distance
- Cosine Distance
- Manhattan Distance

The goal is to determine which distance metric is the most effective in terms of accuracy and F1 score for the given dataset.

## Dataset

The dataset used in this notebook contains features and labels for classification. The exact features and labels are explored during the analysis, with a focus on how the relationships between data points influence the model's performance.

## Distance Metrics

### 1. **Euclidean Distance:**

- Measures the straight-line distance between two points in space.
- Achieved the highest accuracy and F1 score, making it the best metric for this dataset.

### 2. **Cosine Distance:**

- Evaluates the cosine of the angle between two vectors.
- Achieved second-best performance, close to Euclidean, indicating that the direction of data points has some significance.

### 3. **Manhattan Distance:**

- Measures the distance between two points by summing the absolute differences of their coordinates.
- Showed the lowest performance in both accuracy and F1 score.

# Notebook Structure

## 1. Data Preprocessing

The dataset is loaded and preprocessed to ensure it is in a format suitable for KNN classification.

## 2. KNN Model Training and Evaluation

The KNN classifier is trained using each of the three distance metrics (Euclidean, Cosine, and Manhattan). The model's accuracy and F1 score are calculated and compared.

## 3. Results and Conclusions

The performance of each distance metric is evaluated based on the model's accuracy and F1 score. The results show that Euclidean distance is the most effective metric for this dataset, followed closely by Cosine distance, while Manhattan distance shows limited effectiveness.

# Conclusions

1. **Euclidean Distance**: Achieved the highest accuracy and F1 score, making it the most effective metric for this dataset.
2. **Cosine Distance**: Performance was slightly lower than Euclidean but still close, indicating that the orientation of data points matters.
3. **Manhattan Distance**: Recorded the lowest F1 score, suggesting it is less effective in capturing the relationships between data points for this particular dataset.

# Files Included

- `48622613_Portfolio3.ipynb` - The Jupyter Notebook with all the code and analysis.
- `data/` - A folder containing the dataset loan_approval.cvs

# Usage

To run the notebook, simply open it in Jupyter Notebook or any compatible environment and execute the cells in order. The results of the analysis will show how each distance metric performs with the KNN classifier.

# Dependencies

- Python 3.x
- Pandas

- Scikit-learn
- Jupyter Notebook

Install the necessary dependencies by running:

```
pip install pandas scikit-learn jupyter
```

In [ ]: