**Food prediction and classification**

**1. Introduction:**

* our project aims to perform image classification for three different food categories: pizza, falafel, and ice cream.
* The dataset used is located in the '/content/food-101/food-101/food-101/images' directory.

**2. Preprocessing:**

* Images are resized to (64, 64, 3) to standardize their dimensions.
* The HOG (Histogram of Oriented Gradients) features are extracted for each image to capture the local shape and structure information.

**3. Logistic Regression:**

* The Logistic Regression model is employed for image classification.
* The dataset is split into training and testing sets.
* GridSearchCV is used to find the best hyperparameters for the Logistic Regression model.
* The model is trained on the training set, and predictions are made on the testing set.
* Accuracy is a potential evaluation metric for the logistic regression model.

**4. K-Means Clustering:**

* K-Means clustering is applied to the HOG features after PCA (Principal Component Analysis) dimensionality reduction.
* PCA is used to reduce the dimensionality of the feature space for better visualization and potential improvement of clustering results.
* The number of clusters is set to 3, corresponding to the three food categories.
* Silhouette Score is calculated to evaluate the quality of the clustering.

**5. Visualization:**

* Visualization of the original images, HOG features, and clustered results provides insight into the model's performance.
* PCA is used for visualization, reducing the feature space to two dimensions for easy plotting.

**6. Conclusion:**

* A comparison of the results from Logistic Regression and K-Means Clustering is provided.
* Discuss the strengths and weaknesses of each approach and how well they perform in classifying food images.
* Consider potential improvements or future work, such as fine-tuning hyperparameters, exploring different algorithms, or increasing the dataset size.

**6. result:**

* Loss curve and accuracy curve
* Roc curve

A graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of

Description automatically generated

A graph of a line graph

Description automatically generated with medium confidence

**Introduction**:

The aim of this machine learning project is to predict house prices in India using a dataset containing various features related to the houses. We will employ two different regression algorithms, Linear Regression and K-Nearest Neighbors (KNN), to build predictive models. The project involves data preprocessing, feature engineering, model training, and evaluation.

**Dataset**:

The dataset, loaded from a CSV file, includes information about various houses in India. Features in the dataset may include aspects such as the number of bedrooms, area, location, and other relevant details. The target variable is the house price.

**Data Preprocessing:**

**Handling Categorical Variables:**

Identifying and encoding categorical variables using one-hot encoding.

Ensuring that all non-numeric columns are appropriately transformed for model compatibility.

**Handling Missing Values:**

Using the SimpleImputer from scikit-learn to fill missing values, typically with the mean, for numerical features.

**Scaling Features:**

Standardizing numerical features using StandardScaler to ensure that all features have the same scale, which is crucial for certain algorithms like Linear Regression.

**Linear Regression:**

**Data Splitting:**

Splitting the dataset into training and testing sets (e.g., 80% training, 20% testing).

**Feature Scaling:**

Scaling the features using StandardScaler to standardize the numerical features.

**Model Training:**

Creating a Linear Regression model and training it on the training data.

Model Evaluation:

Evaluating the model using metrics such as Mean Squared Error (MSE) and R-squared on the test set.

**Visualization:**

Creating scatter plots to visualize the predicted vs. actual house prices.

Generating histograms for each feature to understand their distributions.

**K-Nearest Neighbors (KNN) Regression:**

**Data Preparation:**

Dropping unnecessary columns (e.g., 'id', 'Date').

Converting the 'number of bedrooms' to integers.

Handling any remaining missing values.

**Data Splitting and Feature Scaling:**

Splitting the data into training and testing sets.

Standardizing the features using StandardScaler.

Model Training and Evaluation:

Creating a KNN Regressor model with a specified number of neighbors.

Training the model on the training set.

Making predictions on the test set.

Evaluating the model using MSE and R-squared.

**Project Implementation:**

**Loading Data and Preprocessing:**

Reading the dataset from a CSV file.

Handling categorical variables, missing values, and scaling features.

**Linear Regression:**

Splitting the data, scaling features, training the Linear Regression model, and evaluating its performance.

**KNN Regression:**

Dropping unnecessary columns, converting data types, handling missing values, splitting data, scaling features, training the KNN model, and evaluating its performance.

**Visualizations**:

Creating scatter plots and histograms to gain insights into the data distribution and model predictions.

**Conclusion:**

Summarizing the findings, discussing the strengths and weaknesses of each model, and providing insights into the factors that significantly impact house prices in the Indian real estate market. Additionally, recommendations for improving

model performance and potential areas for further exploration can be discussed.

A graph of a line graph

Description automatically generated with medium confidence