Project Report

I. Introduction

• **Project Goal:-**

The primary goal of this project is to build and evaluate three different machine learning models—Regression, Classification, and Clustering—using the California Housing Dataset. The regression model aims to predict house prices, the classification model aims to identify whether a house is expensive or not, and the clustering model aims to group similar houses into clusters without using labels.

• **Data Description:-**

The dataset used in this project is the

California Housing Dataset (for Regression & Clustering)

Dataset Size:

- Total rows: 20,640
- Total columns: 10

Features Included:

- longitude
- latitude
- housing_median_age
- total_rooms
- total_bedrooms
- population
- households
- median_income
- median_house_value (target for regression)
- ocean_proximity (categorical feature)

Heart Disease Dataset (for Classification)

Target: `Heart Disease` (binary)

Original values: `'Absence'` / `'Presence'`

Converted to numeric: 0 → Absence, 1 → Presence

II. Data Preprocessing & EDA

1. Handling Missing Values

A full inspection of missing values was performed using: `df.isnull().sum()`
The dataset contained no missing values in most columns, except for:

- total_bedrooms → 207 missing entries

To address this issue, the Median Imputation strategy was applied.
The median is preferred over the mean because it is more robust to outliers and preserves the distribution of housing-related numerical features.

2. Feature Encoding

The dataset included one categorical feature:

- ocean_proximity (textual values)

Since machine learning models require numerical inputs, this feature was converted using:
✔ One-Hot Encoding
Applied via: `pd.get_dummies(df, columns=["ocean_proximity"])`
This transformation:

- Increased the total number of feature columns from 10 to 12

3. Outlier Detection

Outliers were examined using box plots for numerical features such as:

- median_income
- total_rooms
- housing_median_age

Outliers were **not removed** as they represent real variations in housing dat

4. Feature Scaling

StandardScaler applied to numeric features for models sensitive to scale (Logistic Regression, K-Means)
Mean = 0, Std Dev = 1

```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

This ensures fair contribution of all numeric features during model training.

5. Train–Test Split

To properly evaluate model performance, the dataset was split into training and testing subsets using:
`train_test_split(test_size=0.2, random_state=42)`
This produced:

- Training set shape: (16512, 12)
- Testing set shape: (4128, 12)

Heart Disease Dataset:

- Training: 80% of samples
- Testing: 20% of sample

An 80/20 split ensures enough data for training while keeping an adequate sample for evaluation.
The random_state=42 ensures reproducibility.

6. Target Variable Definition

Regression: `median_house_value` (continuous)

Classification: `Heart Disease` (0 → Absence, 1 → Presence)

Clustering: All features, no target

III. Modeling and Results

Regression Model Results — Linear Regression

The Linear Regression model was trained to predict the continuous variable median_house_value.
The model's performance on the test set was as follows:

- MSE: 4,908,476,721.1566
- RMSE: 70,060.52
- R^2 Score: 0.6254

Interpretation:
The RMSE indicates that the model's average prediction error is around 70,060 USD, which is reasonable given the price range of the dataset.
An R^2 score of 0.625 means that the model explains 62.5% of the variance in house prices. This is a moderate performance, suggesting that the model captures meaningful patterns, but additional features or more advanced models could improve accuracy.

Classification Model Results —Logistic Regression

Target: `Heart Disease`

Results:

Accuracy: 0.9074 (≈ 91%)

Confusion Matrix:

[[31 2]

[3 18]]

Precision: 0.900

Recall: 0.857

Interpretation:

Model is highly accurate and balanced

Precision = 90% → when predicting disease, correct 90% of the time

Recall = 85.7% → detects most patients with heart disease

Classification Model Results — Decision Tree Classifier

Target: `Heart Disease`

Results:

Accuracy: 0.685 (≈ 69%)

Confusion Matrix =

[[22 11]
[6 15]]

Precision: 0.577 // Recall: 0.714

Interpretation:

Lower accuracy than Logistic Regression

Precision = 57.7% → fewer correct positive predictions

Recall = 71.4% → detects most positive cases but with more false positives

Conclusion: Logistic Regression outperforms Decision Tree for this dataset

Clustering Results – K-Means Clustering Results

To identify natural groupings in the data, K-Means clustering was applied to the scaled feature set.
The Elbow Method was used to determine the optimal number of clusters.

Optimal K Selection (Elbow Method)
The inertia dropped sharply between K = 1 and K = 3, after which the rate of improvement slowed noticeably.

This indicates that the “elbow” occurs around:

👉 K = 3 clusters
This value represents the point where adding more clusters yields diminishing returns.
Silhouette Score
The clustering quality was evaluated using the Silhouette Score:

- Silhouette Score: 0.2829

Interpretation:
A silhouette score around 0.28 indicates moderate separation between clusters. The clusters are somewhat distinct but still show overlap, which is expected given the complexity of the dataset and high-dimensional features.

Cluster Interpretation
After applying K-Means with K = 3, the dataset was divided into three groups that represent different patterns in housing attributes (e.g., location, income level, population density).
Although clustering is unsupervised, these clusters help reveal underlying structure in the data.

IV. Conclusion

Regression: Linear Regression (moderate performance)
Classification: Logistic Regression (high accuracy and balanced precision/recall)
Clustering: K-Means (3 clusters, moderate silhouette)

Challenges:

Regression: High RMSE due to outliers and wide price range
Classification: Decision Tree less effective on this dataset

Clustering: Moderate separation due to complex, high-dimensional features

Potential Next Steps:

Experiment with advanced regression models (e.g., Gradient Boosting, Random Forest)
Feature engineering to improve classification performance

Dimensionality reduction (e.g., PCA) to improve clustering visualization

V. Resources

Colab Notebook Link:
https://colab.research.google.com/drive/1dOXY8GfKGnSBgOXNcp2EYDkg2gfOsN9K?usp=sharing