Date 10/3/25
Page

Lab - 2.

Demonstrate the steps to build a machine learning model that predicts the median housing price using the California housing price dataset.

1. Perform describe & info steps

```
data = pd.read_csv("Housing.csv")
data.describe()
data.info()
```

2. Plot the histogram of each feature (Indicate what does histogram indicate on median-income & house-median-age)

```
data.hist(bins = 50, figsize=(20,15))
plt.show()
```

3. Demonstrate the process of creating a test set (Write the difference b/w random & stratified test set).

Random split = Splits data randomly without considering feature distribution. This can lead to skewed test sets if the dataset is imbalanced (e.g: if median-income has uneven representation).

Stratified split: Ensure the test set reflects the distribution of a key feature (median-income). This is crucial when a feature ~~strongly~~ correlates with the target (house value) reducing sampling bias

Date 10/3/25
Page

4. What does geographical features graph indicate w.r.t housing price & location

The graph shows California's outline, with higher house values (red/yellow) concentrated near the coast. Inland areas (green/blue) have lower values. This indicates a strong location-based influence on housing-price, with proximity to the coast correlating with higher values.

5. Plot graph to show feature correlation with housing price, which feature correlates to maximum. Analyze what graph indicates

Maximum correlation : median-income

The scatter plot of median-income vs median-house value shows a clear positive trend - higher incomes correspond to higher house values.

6. List the features that could be combined to improve correlation.

Features to combine

1. total-rooms & households → room-per-household
2. total-bedrooms & total-rooms → bedrooms-per-room
3. population & household → population-per-household

Observation: room-per-household improves correlation slightly over total-rooms. The plot shows positive trend, but with more scatter, indicating less predictive alone than median-income

Date 10/3/25
Page

7. List the features that needs to cleaned.

total-bedrooms has missing values (207 missing)

median = data["total-bedrooms"].median()
data["total-bedrooms"].fillna(median, inplace=true)

8. Is there any categorical data that needs to be converted to numerical? If so explain the method used to convert.

Yes, there is a categorical data ocean-proximity. Use one-hot-encoding (since its nominal, not ordinal)
Converts category Nearbay, Ocean into binary columns.

encoder = OneHot Encoder (sparse = false)
ocean-encoded = encoder.fit-transform(data[["ocean proximity"]])

9. Discuss the importance of feature scaling.

* Many ML algorithms are sensitive to features like total-rooms & median-income.
* Scaling ensures all features contribute equally improving convergence & accuracy.

Methods: Standardization, Normalization

10. Design a pipeline including (missing transform, feature scaling & encoding).

Date 10/3/25
Page

1. Numeric pipeline:
- fills missing values (total_bedrooms) with median
- Adds derived features (rooms_per_household, bedrooms_per_room, population_per_household).
- Scaler: standardizes numeric features to zero mean & unit variance.

2. categorical pipeline:

OneHotEncoder: converts ocean_proximity into binary columns.

10/3