MACHINE LEARNING PROJECT

LIFE CYCLE

The main steps in every machine learning project:

1. ***Look at the big picture:***

   In this step you have to

   - Frame the problem:

     To determine how to design your system you need to find answers for these questions:

     - What is the business objective?
     - What the current solution looks like?

   - Designing your system:

     Now you need to answer the following questions:

     - What kind of algorithms you will use based on the supervision type.
     - Is it regression or classification problem.
     - Should you use batch or online learning.

   - Select a performance measure.

   - Check the assumptions.

2. ***Get the data:***
    - Download the data.
    - Take a quick look at the data structures.
    - Create a train and test set.

3. ***Explore and visualize the data to gain insights:***
    - Visualizing geographical data.
    - Look for correlations.
    - Experiment with attribute combinations.

4. ***Prepare the data for machine learning algorithms:***
    - Clean the data.
    - Handling text and categorical data.
    - Feature scaling and transformation.
    - Build custom transformers.
    - Build transformation pipeline.

5. ***Select a model and train a model:***
    - Train and evaluate the model on the training set.
    - Get better evaluation using cross validation.

6. ***Fine tune your model:***
    - After you select promising models, you need to fine tune them, there are a few ways to do that:
        - grid search.
        - Randomized search.
        - Ensemble search.
    - Analyzing the best models and their errors.
    - Evaluate your system on the test set.

7. ***Present your solution.***

8. ***Launch, monitor, and maintain your system.***

Apply the previous steps on a real data:

1. ***Look at the big picture:***
   We will use California housing price dataset to building a model to predict housing prices in the state.
   - Frame the problem:
     The goal of building the model is to predict house prices based on house features.
     The output of the model will be presented to real estate investors to know the estimated value of the house prices.
     The current solution is to hire real estate experts to estimate the house prices.

   - Designing the system:
     We will use supervised algorithms because we have a labeled data.
     Our problem is regression problem (house prices).
     We will use batch learning because the data is small and doesn't change quickly over time.

- Select a performance measure:
  Because we may have a large variance in house prices, and many outliers in house prices that cannot be dispensed with, so that the best performance measure in this situation is MAE (mean absolute error).

- Check the assumptions:
  The model must predict prices close to real houses values.

2. ***Get the data:***
   - Download the dataset to the local machine.

   - Take a quick look at the data structure:

To discover what are the features and what is the data shape.

3. ***Explore and visualize the data to gain insight:***

   - Review the first rows to see what the data looks like, and using functions like (head,

info, describe and value counts) to know the basic information about the data.
Through the previous processes we found that:

- the dataset contains 20640 samples.
- Each sample has 10 features as follows:
    1) Longitude: house coordinates in longitude.
    2) Latitude: house coordinates in latitude.
    3) Housing median age: age of the house
    4) Total rooms: number of all rooms in the house.
    5) Total bedrooms: number of bedrooms in the house.
    6) Population: total number of people in the house.
    7) Households: total number of families in the house.
    8) Median income: median income for families within a house measured in tens of thousands of USD.

9) Ocean proximity: degree of proximity of the house to the see.

10) Median house value: the price of the house (the label for our algorithm).

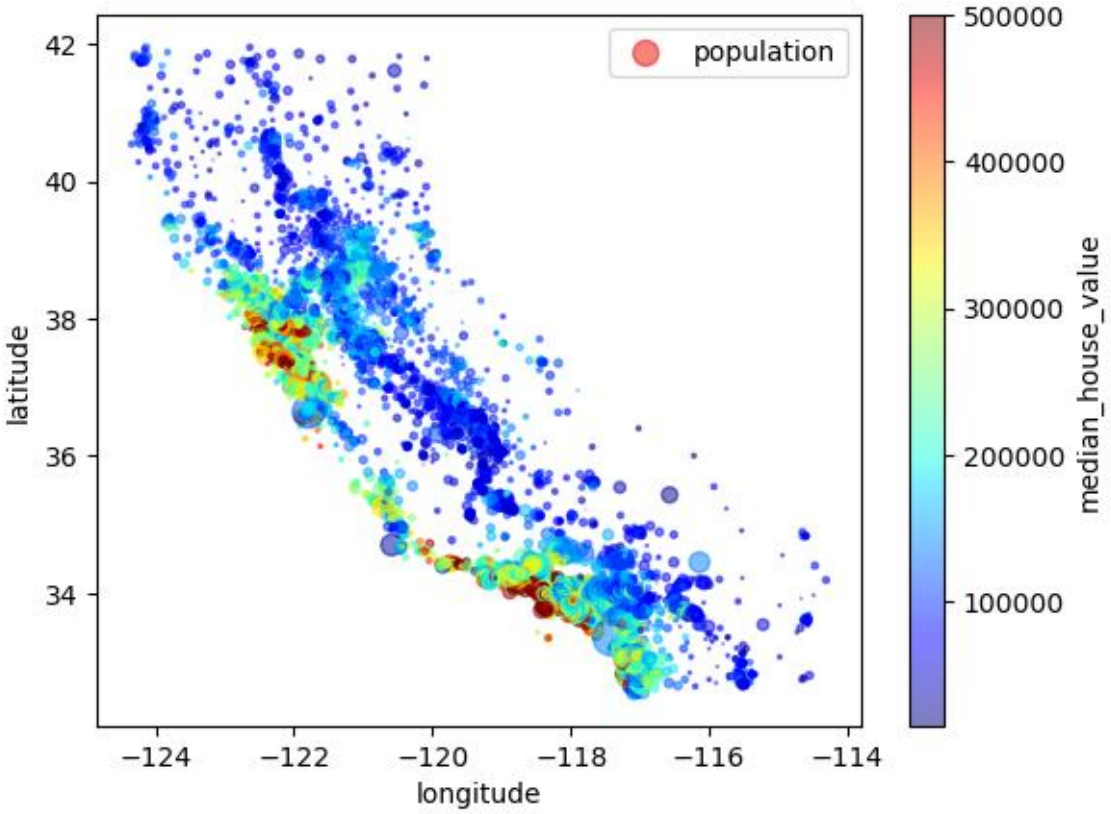
- All the features are continuous numerical values except ocean proximity is categorical value, and it has five different values.
- The following table show the summary of the numerical attributes.

	longitude	latitude	Housing median age	total rooms	total bedrooms	population	households	median income	median_house_value
count	20640.000000	20640.000000	20640.000000	20640.000000	20433.000000	20640.000000	20640.000000	20640.000000	20640.000000
mean	-119.569704	35.631861	28.639486	2635.763081	537.870553	1425.476744	499.539680	3.870671	206855.816909
std	2.003532	2.135952	12.585558	2181.615252	421.385070	1132.462122	382.329753	1.899822	115395.615874
min	-124.350000	32.540000	1.000000	2.000000	1.000000	3.000000	1.000000	0.499900	14999.000000
25%	-121.800000	33.930000	18.000000	1447.750000	296.000000	787.000000	280.000000	2.563400	119600.000000
50%	-118.490000	34.260000	29.000000	2127.000000	435.000000	1166.000000	409.000000	3.534800	179700.000000
75%	-118.010000	37.710000	37.000000	3148.000000	647.000000	1725.000000	605.000000	4.743250	264725.000000

	longitude	latitude	Housing median age	total rooms	total bedrooms	population	households	median income	median_house_value
max	-114.310000	41.950000	52.000000	39320.000000	6445.000000	35682.000000	6082.000000	15.000100	500001.000000

- Form the previous table we can notice a lot of things like the wide variation in house prices, and median income.

- The following image show the houses locations and prices.



- From the image before we notice there are two main areas in which prices are high, and these areas are near to the see.

- The following image showing the correlations between the numerical features.

- The most influential factor in price is the median income, followed by location, age and total rooms, they have almost the same effect.

4. Prepare the data:

➢ Clean the data:

There are some missing values in the total bedroom column, we fill it with the median value of this column.

Then we combine some columns to create more useful features:

- Rooms per households: represent the number of rooms for each family in the house.
- Bedrooms per rooms: represent the percentage of the bedrooms out of the total rooms.
- Population per household: represent average number of family members in the house.

➢ Handling text and categorical data:
We have one categorical feature (ocean proximity), handle it using one hot encoder.

➢ Feature scaling:
Scale the numerical features using StandardScaler.

➢ Build custom transformers:

To apply the previous operations, we need to build transformers:

- First, we need to build custom transformer to create the new features that we mentioned above, and we named it CombinedAttributesAdder.
- After that we need to build pipeline to apply transformations in the numerical features.
- And another pipeline for categorical features.
- Finally, we combine them together in one pipeline, named full pipeline.

Lastly, we divide the data into training and testing sets.