AFRICAN INSTITUTE FOR MATHEMATICAL SCIENCES

(AIMS RWANDA, KIGALI)

Name: Lucas Mirija RAZAFIMANANTSOA
Course: Statistical Machine Learning

Assignment Number: 1
Date: December 7, 2025

Exercise 1

Problem statement : *House pricing*

We will build a model to predict the average house price in a small area of California. We will use informations about the area to do that, using the built in dataset from .

Wheel 1 : Clear problem formulation

In this task, we do a regression to predict $y \in \mathbb{R}$, given a vector of features $x \in \mathbb{R}^p$.
In fact, this is a regression since our goal is to predict a price, which is a continuous quantity.
Our main goal here is to find a function $\hat{f}$ such that $\forall x_i \in \mathbb{R}^p$,

$$\hat{f}(x_i) \approx y_i$$

with minimum error.

Hence, the main metric we will use is *RMSE*.

Wheel 2 : The data

The dataset used in this study is the California Housing dataset from *Scikit-learn*. It contains data collected from the 1990 California census, where each row represents informations about a **block group**, which is a small geographic area containing several hundred to a few thousand people.
The dataset includes the features (x):

- **MedInc:** Median income in the block group.
- **HouseAge:** Median age of the houses.
- **AveRooms:** Average number of rooms per house.
- **AveBedrms:** Average number of bedrooms per house.

- **Population:** Number of people living in the block group.
- **AveOccup:** Average number of people per household.
- **Latitude, Longitude:** Geographical location of the block group.

The target variable (y) is **MedHouseVal**, which represents the median house value in the block group.

Formally, the data can be written as :

$$\mathcal{D}_n = \left\{(x_i, y_i) \overset{\text{iid}}{\sim} p(x, y), \quad x_i \in \mathcal{X},\ y_i \in \mathcal{Y}, \quad i = 1, \ldots, n\right\}$$

In our case

- $n = 20640$, (already without missing values, obtained using `df.count()`).
- $\mathcal{X} \subset \mathbb{R}^p$ where $p = 8$, (using `X.shape`).
- $\mathcal{Y} \subset \mathbb{R}$
- The population is $\mathcal{X} \times \mathcal{Y} \supset \mathcal{D}_n$.
- $p(x, y)$ is the joint distribution which is unknown.

Exporatory Data Analysis

- **Data types**:
  Every components of $x \in \mathcal{X}$ is a real number. The target value y is a real number. (`df.info()`)
- **Missing values** :
  By observing the information about the data, we observe that each column contains 20640 non null values, that means that the dataset doesnt contain any missing value. (`df.info()`)
- **Getting insight on the features and target**:
  - **Correlation**:

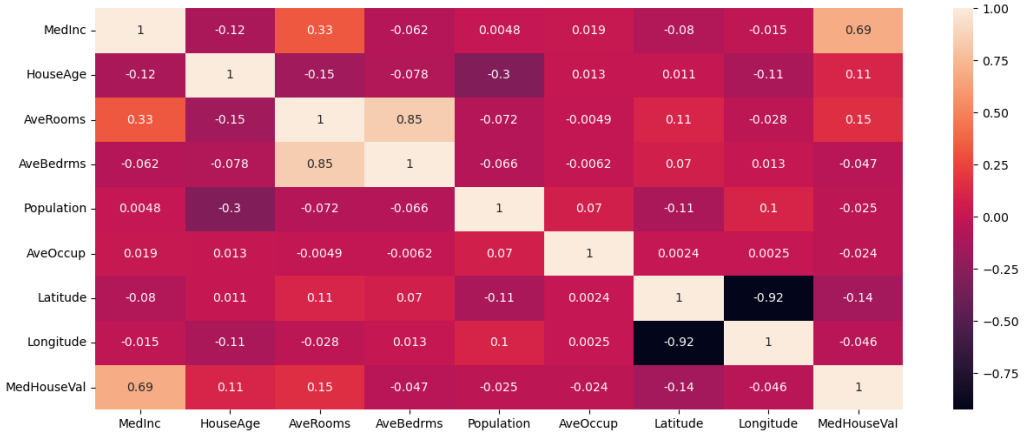


Figure 1: Correlation matrix heatmap

Since we are interested in the house price, we only need to look at the last row or last column of that correlation matrix. We now have to keep in mind that the **Median Income** correlates highly with the target variable.

– **Geographic features**

We can consider the plot below as a map of California. Notice that the south-west part have higher house prices (near the coastline which is represented by the red line) and more central areas have lower prices. That means that Latitude **AND** Longitude are good predictors of the house price in California.

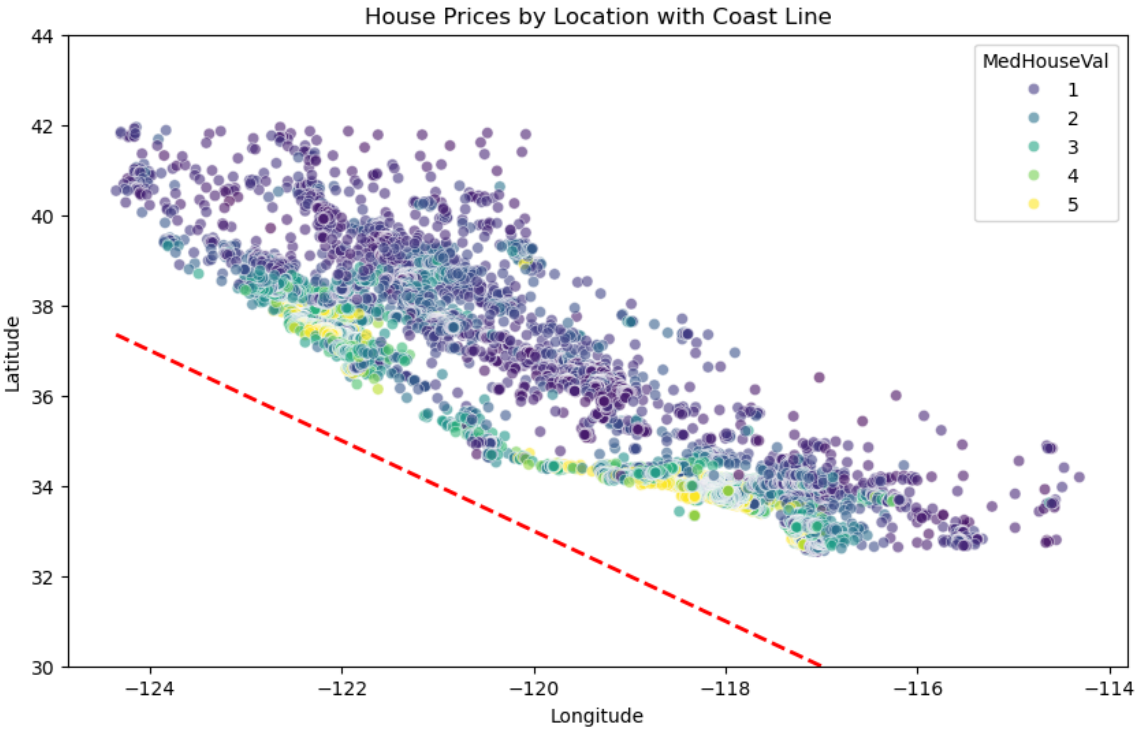
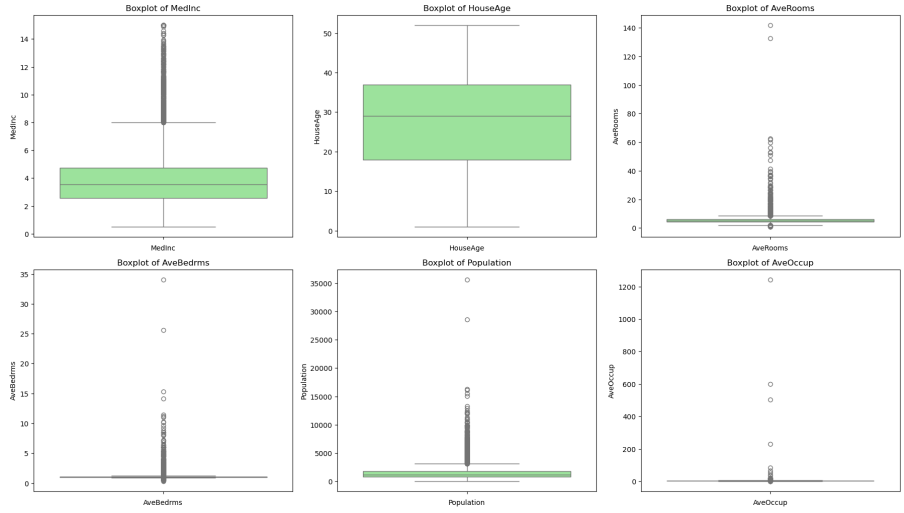


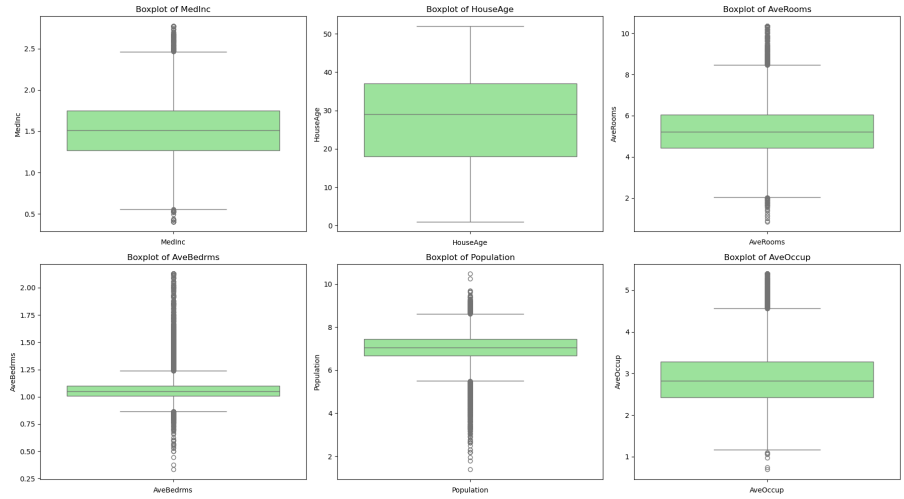
Figure 2: Geographic map

Intuitively, the other unmentionned features also play crutial roles in predicting the price of a house but we can't see that 'relation' for now. So let's manipulate those features and the target to make them make sense.

- **Features engeneering**:
  - **Visualization**:



(a) Before



(b) After

Figure 3: Comparison of feature distributions before and after engineering

  - **MedInc** and **Population** were right-skewed so we take the log to remove the skew.
  - **HouseAge** is already nicely distributed.
  - For the other variables, there are some unexpected value (for example **AveRoom**>20). By reflecting on the meaning of those features, we can deduce that maybe they are not actually coming from the houses, but from some blocks that contain mainly big buildings but small number of houses. To deal with those values, we cap them to the 99% percentile.
  - For the geographical features, we transformed them into the distance to the coastline

of equation $Ax + By + C = 0$. Approximated by:

$$\text{DistToCoast} = \frac{|A \cdot \text{Longitude} + B \cdot \text{Latitude} + C|}{\sqrt{A^2 + B^2}}$$

Notice that the line only has to be somehow parallel to the beach because we are interested in the proximity. We now have $p = 7$

Wheel 3 : Function Space and Model Specification

We saw that we have two *good* linear predictors of y in the features, i.e **MedInc** and **Distance to Coast**. It is then reasonable to consider a linear model with parameters that are to be optimized later on:

$$\mathcal{H}_{lin} := \left\{ f : \mathcal{X} \to \mathcal{Y} \,\middle|\, f(\mathbf{x}) = w^T \mathbf{x} + b,\ w \in \mathbb{R}^p,\ b \in \mathbb{R} \right\}$$

However, as mentioned previously, some other features also influence house prices but do not show a clear linear correlation with the target. This may be because their effect is *hidden* within non-linear transformations or interactions between variables (for example, combining longitude and latitude into distance to the coast).
Therefore, it is reasonable to consider non-linear models. Candidate hypothesis spaces include **Decision Trees** and **Random Forests**, which can capture non-linear relationships and feature interactions.

$$\mathcal{H}_{tree} := \left\{ f : \mathcal{X} \to \mathcal{Y} \,\middle|\, f(\mathbf{x}) = \sum_{m=1}^{M} c_m \mathbf{1}_{\{\mathbf{x} \in R_m\}},\ R_m \text{ disjoint regions},\ c_m \in \mathbb{R} \right\}$$

And

$$\mathcal{H}_{for} := \left\{ f : \mathcal{X} \to \mathcal{Y} \,\middle|\, f(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t f_t(\mathbf{x}),\ f_t \in \mathcal{H}_{tree},\ \alpha_t \in \mathbb{R} \right\}$$

Wheel 4 : Risk Minimization

In this regression task, we will estimate the risk using the **RMSE** over the training set, defined as:

$$\hat{R}_{train}(f) := \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left(f(\mathbf{x}_i) - y_i \right)^2}$$

However, to account for generalization and avoid overfitting, we consider the **cross-validation RMSE** instead:

$$\hat{R}_{CV}(f) := \sqrt{\frac{1}{K} \sum_{k=1}^{K} \frac{1}{|V_k|} \sum_{i \in V_k} \left(f^{(-k)}(\mathbf{x}_i) - y_i \right)^2}$$

where V_k is the validation fold k, and $f^{(-k)}$ is the model trained without fold k.
We will do this procedure for each candidate model (resampling several times) and select the one with the smallest cross-validation RMSE.

Wheel 5 and Wheel 6

- We used cross-validation with repeated resampling to evaluate each model's performance, including regularization for the linear models (see notebook).
- Random Forest achieved the lowest RMSE and performed the best overall.
- Hyperparameters were tuned using a grid search combined with cross-validation.
- The minimum RMSE obtained is $\hat{R}_{CV}(\hat{f}) \simeq 0.54$. Considering our target represents house values in hundreds of thousands of dollars, this corresponds to a typical prediction error of approximately \$54,000, which is reasonable for estimating house prices.

Wheel 7

- We evaluated the best-performing Random Forest on the test set.
- The test RMSE is $\hat{R}_{test}(\hat{f}) \simeq 0.55$, which is very close to the cross-validation RMSE ($\simeq 0.54$).
- This indicates that the model generalizes well and does not overfit the training data (whose RMSE is $\simeq 0.2$).
- In practical terms, this corresponds to a typical prediction error of approximately \$55,000 on house values, which is reasonable for our dataset.

Wheel 8 : Statistical Inference and Theoretical Justification

- The squared loss used in RMSE minimization is theoretically justified because its minimizer is the conditional expectation $\mathbb{E}[Y|X]$, which is the target in regression tasks.
- Linear models capture additive trends, while trees and forests can approximate complex non-linear relationships and interactions between features.
- Cross-validation provides an approximate estimator of the out-of-sample prediction error, ensuring that the selected model generalizes beyond the training data.
- we choose the CV-risk as benchmark because it "mimics" the global out-of-sample risk:

$$\widehat{R}_{\mathrm{CV}}(f) \;\approx\; R(f) \quad \text{where} \quad R(f) = \mathbb{E}\big[(y - f(x))^2\big] \,.$$

Wheel 9 : Deployment

The model is saved using Joblib and compressed in a `tar.xz` format.
The project is available on this link
`https://github.com/lucas-razafimanantsoa/California-house-pricing.git`