Sound Realty ML Serivce

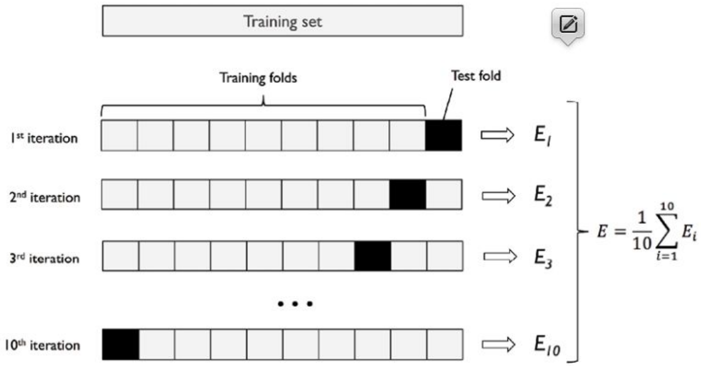
Non-technical presentation

Agenda

1. Model Validation
2. Model Improvements
3. Solution Architecture
4. Future Improvements

1. Model Validation

To validate the model, a 10-fold cross validation was conducted using the Sales Data and the model configuration from Sound Realty's proof of concept.



1. Model Validation

The KNN using the features selected at the proof of concept showed good generalization capabilities, showing an **average R2 metric of 0.752**.

- Average MSE: 33,226,673,237.08
- Average RMSE: $ 181,808.10
- Average MAE: $ 95,383.69
- **Average R2: 0.752**

2. Model Improvement

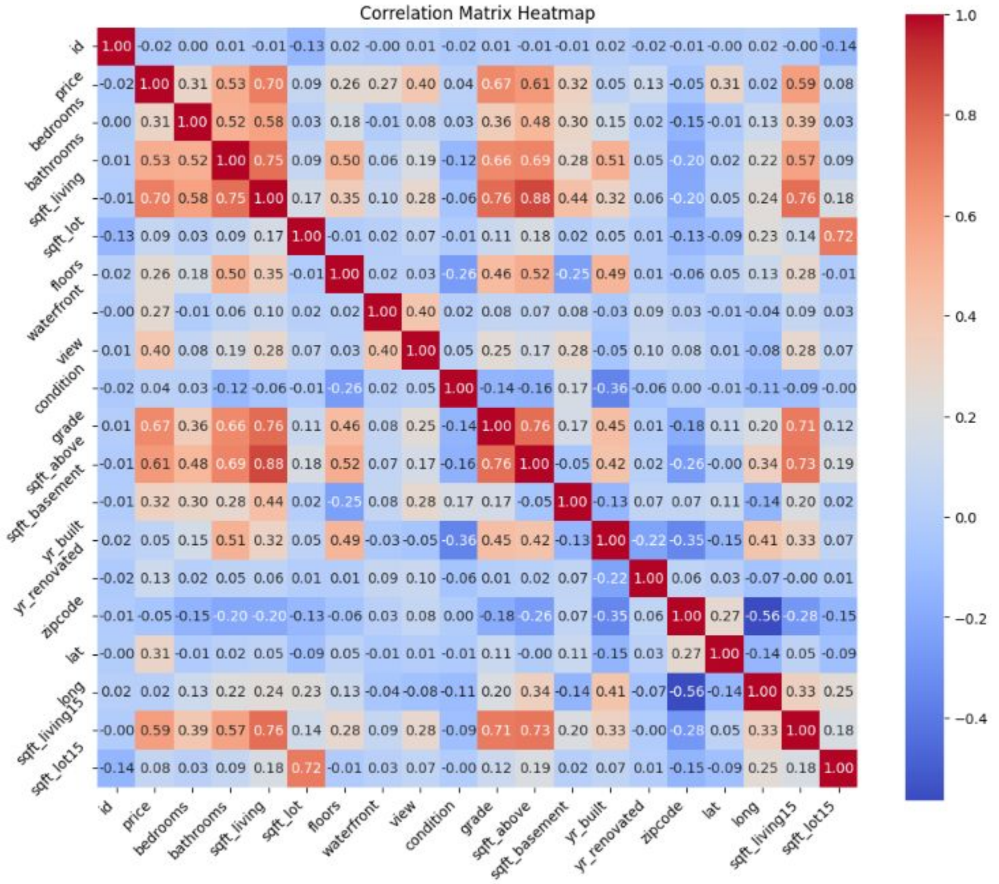
An Exploratory Data Analysis (EDA) was conducted on the available datasets to assess how the original model could be improved.

A Correlation Analysis showed that some features from the Sales Data with high correlation with the target price weren't being used, such as **grade**, **waterfront**, **view** and **lat**. On the other hand, features with low correlation with price were being used, such as **sqft_lot**.

Therefore, a selection of the features from the Sales Data with correlation with price higher than 0.2 were selected:

- Features **waterfront**, **view**, **grade**, **lat** and **sqft_living15** were added
- Feature **sqft_lot** was removed

2. Model Improvement



2. Model Improvement

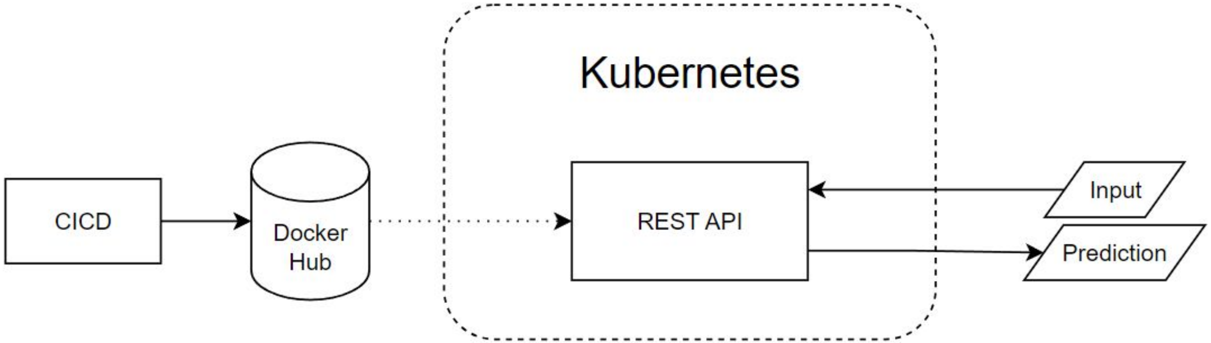
The new feature selection enabled the KNN to increase its R2 metric, reaching a value of **0.831** (~ 10% increase).

- Average MSE: 22,819,963,434.86
- Average RMSE: $ 150,608.94
- Average MAE: $ 80,811.54
- **Average R2: 0.831**

3. Solution Architecture

The solution was built to be as simple as possible:

1. Models and demographic data embedded in the Docker Image
2. Docker image deployed to a Kubernetes cluster using a CICD pipeline



3. Solution Architecture

Nonetheless a simple solution, it enables:

- Auto scaling
- Deployment of new models without stopping the service
- CICD automation

3. Future Improvements

Improvements for the solution architecture are already planned:

1. Create a proper training pipeline, in a separate repository and infrastructure
2. Implement Experiment Tracking and a Model Registry with MLFlow
3. Read Demographic Data from a Redis instance
4. Further improve auto scaling by using Keda and Prometheus

Those improvements will implement basic MLOps and enhance the API robustness.

Thank you!

Sound Realty ML Serivce

Technical presentation

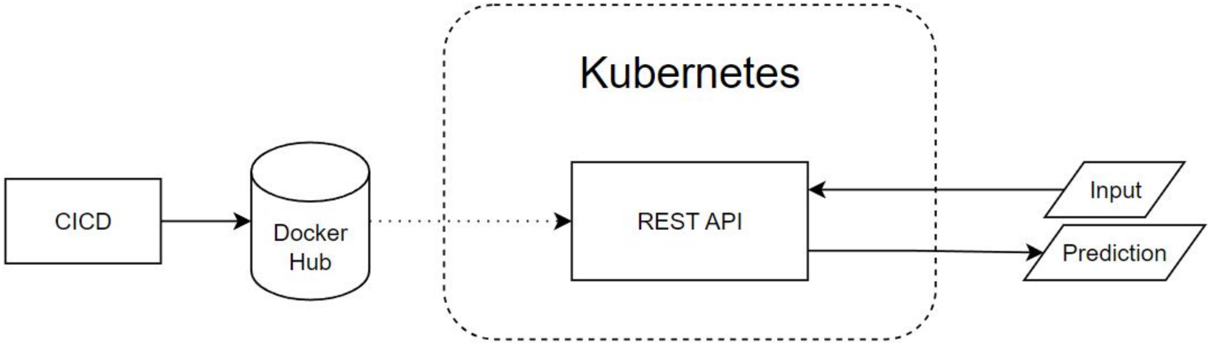
Agenda

1. Solution Architecture
2. Future Improvements

1. Solution Architecture

The solution was built to be as simple as possible:

1. Models and demographic data embedded in the Docker Image
2. Docker image deployed to a Kubernetes cluster using a CICD pipeline



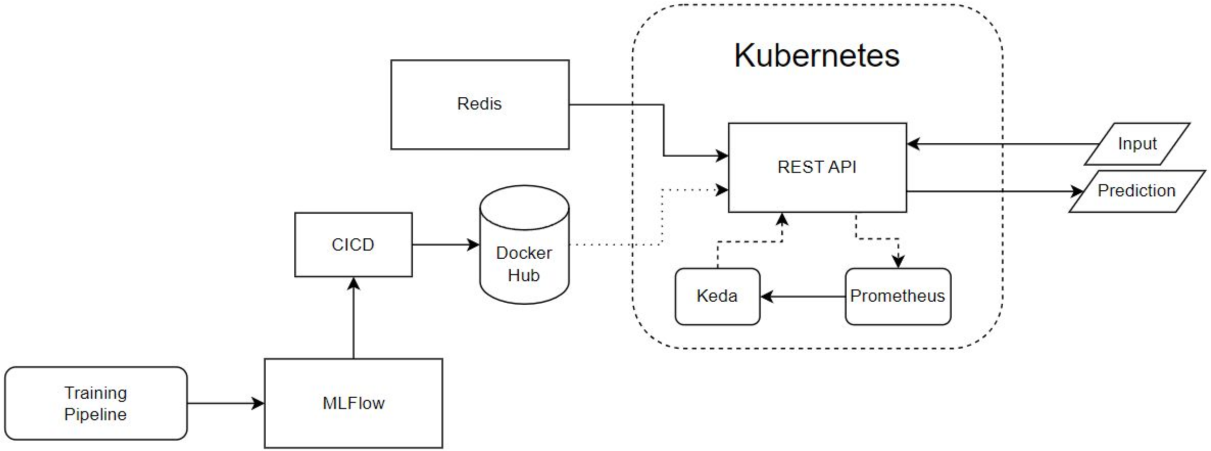
1. Solution Architecture

Nonetheless a simple solution, it enables:

- Auto scaling
- Deployment of new models without stopping the service
- CICD automation

1. Solution Architecture

However, the ideal architecture planned for the Sound Realty ML Service is presented below:



2. Future Improvements

The API robustness is the main goal at first, nonetheless there are still enough space for some MLOps Improvement.

1. Implement Data Quality monitoring during training
2. Configure Data Drift monitoring for the inputs sent to the API
3. Use an ML framework for the REST API, such as NVidia Triton