Supervised Learning - Regression

Data Science

California Housing Dataset

Written by:
Shubh Desai

July 12, 2024

Supervised Learning – Regression

Business Understanding

- **Problem Statement**

The problem we are addressing is predicting target variable based on various attributes in the California dataset. This is a supervised learning problem where the goal is to accurately predict the target variable.

- **Importance of the Problem**

Predicting this target variable is important because it helps in understanding the factors that influence the outcome. Accurate predictions can lead to better decision-making and strategic planning.

- **Data Source**

The dataset was downloaded from Kaggle. The specific dataset used is the "California dataset". The dataset is named california.csv and contains the following columns:

Data Collection

This dataset offers details on a range of Californian housing units.
Variables:

- MedInc: median income in block group
- HouseAge: median house age in block group
- AveRooms: average number of rooms per household
- AveBedrms: average number of bedrooms per household
- Population: block group population
- AveOccup: average number of household members
- Latitude: block group latitude
- Longitude: block group longitude

Row #	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	house_price
1	8.3252	41	6.9841269841	1.0238095238	322	2.5555555556	37.88	-122.23	4.526
2	8.3014	21	6.2381370826	0.9718804921	2401	2.1098418278	37.86	-122.22	3.585
3	7.2574	52	8.2881355932	1.0734463277	496	2.802259887	37.85	-122.24	3.521
4	5.6431	52	5.8173515982	1.0730593607	558	2.5479452055	37.85	-122.25	3.413
5	3.8462	52	6.2818532819	1.0810810811	565	2.1814671815	37.85	-122.25	3.422
6	4.0368	52	4.7616580311	1.103626943	413	2.1398963731	37.85	-122.25	2.697
7	3.6591	52	4.9319066148	0.9513618677	1094	2.1284046693	37.84	-122.25	2.992
8	3.12	52	4.7975270479	1.0618238022	1157	1.7882534776	37.84	-122.25	2.414
9	2.0804	42	4.2941176471	1.1176470588	1206	2.0268907563	37.84	-122.26	2.267
10	3.6912	52	4.9705882353	0.9901960784	1551	2.1722689076	37.84	-122.25	2.611
11	3.2031	52	5.4776119403	1.07960199	910	2.263681592	37.85	-122.26	2.815
12	3.2705	52	4.772479564	1.0245231608	1504	2.0490463215	37.85	-122.26	2.418
13	3.075	52	5.3226495726	1.0128205128	1098	2.3461538462	37.85	-122.26	2.135
14	2.6736	52	4	1.0977011494	345	1.9827586207	37.84	-122.26	1.913
15	1.9167	52	4.2629032258	1.0096774194	1212	1.9548387097	37.85	-122.26	1.592
16	2.125	50	4.2424242424	1.071969697	697	2.6401515152	37.85	-122.26	1.4
17	2.775	52	5.9395770393	1.0483383686	793	2.3957703927	37.85	-122.27	1.525
18	2.1202	52	4.0528052805	0.9669966997	648	2.1386138614	37.85	-122.27	1.555
19	1.9911	50	5.3436754177	1.0859188544	990	2.3627684964	37.84	-122.26	1.587
20	2.6033	52	5.4654545455	1.0836363636	690	2.5090909091	37.84	-122.27	1.629

Data Understanding

- **Exploratory Data Analysis (EDA)**

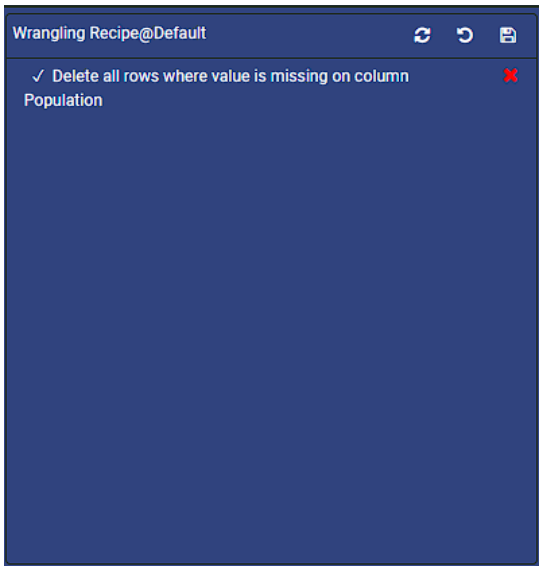
Initial data exploration reveals the following characteristics about the dataset: The dataset comprises 20,640 entries, each with 9 features.

The target variable in this context can be considered as house_price, which we aim to understand and predict based on other housing and demographic features.

Data Preparation

Handling Missing Values

Missing values in the dataset were handled by deleting.



Data Splitting

The dataset was split into training and testing sets: 80% of the data was used for training and 20% was reserved for testing.

- 80% of the data was used for training.
- 20% of the data was reserved for testing.

Methodology

Model Selection

For this task, we used the following algorithms:

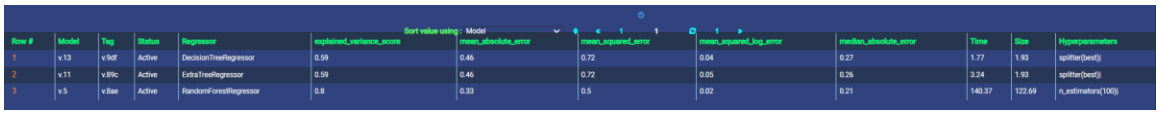
- **RandomForestRegressor**
  A random forest is a meta estimator that fits a number of decision tree regressors on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.
- **ExtraTreeRegressor**
  Extra-trees differ from classic decision trees in the way they are built. When looking for the best split to separate the samples of a node into two groups, random splits are drawn for each of the max_features randomly selected features and the best split among those is chosen.
- **DecisionTreeRegressor**
  Decision Tree Regressor tries to predict a continuous target variable by cutting the feature variables into small zones, and each zone will have one prediction.

Version-Tag	Dataset	Algorithm	Rank	Error	Doc.	Publish	Delete
v.5-v.8ae	california_dataset-california	RandomForestRegressor	1	0.33			
v.13-v.9df	california_dataset-california	DecisionTreeRegressor	2	0.46			
v.11-v.89c	california_dataset-california	ExtraTreeRegressor	2	0.46			

Model Evaluation

Evaluation Metrics

The models were evaluated on the test dataset using the following metrics:

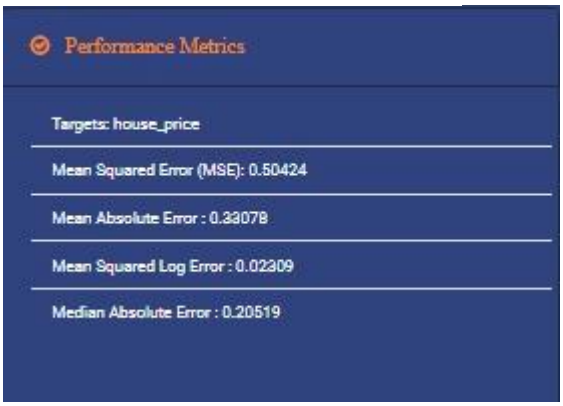


Row #	Model	Tag	Status	Regressor	explained_variance_score	mean_absolute_error	mean_squared_error	mean_squared_log_error	median_absolute_error	Time	Size	Hyperparameters
1	v.13	v.9df	Active	DecisionTreeRegressor	0.59	0.46	0.72	0.04	0.27	1.77	1.93	splitter(best)|
2	v.11	v.89c	Active	ExtraTreeRegressor	0.59	0.46	0.72	0.05	0.26	3.24	1.93	splitter(best)|
3	v.5	v.8ae	Active	RandomForestRegressor	0.8	0.33	0.5	0.02	0.21	140.37	122.69	n_estimators(100)|

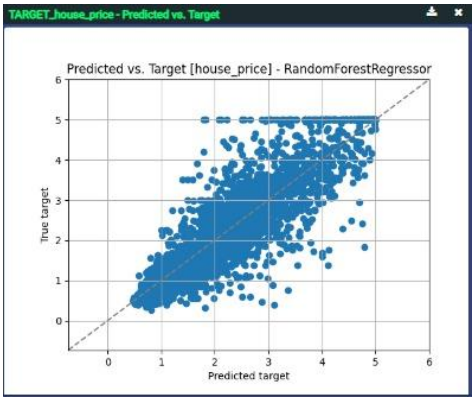
Best Model

The best-performing model is RandomForestRegressor with 0.33 error rate.

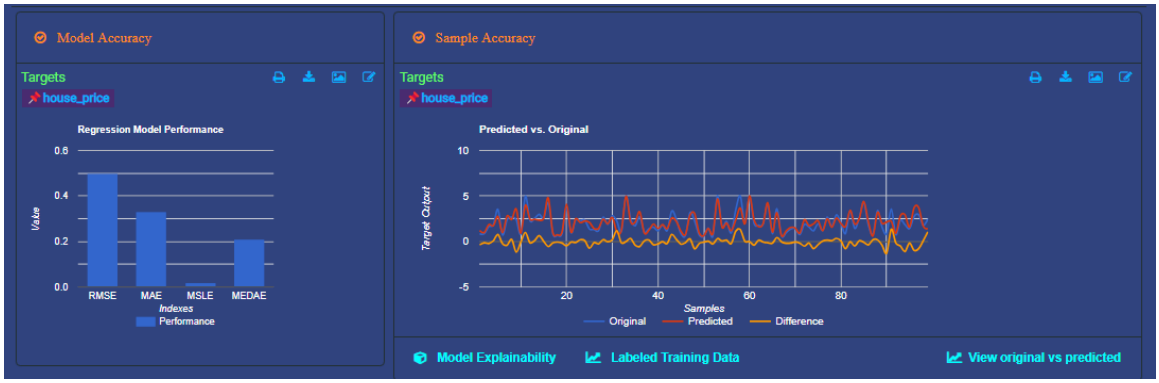
Performance Metrics



Predicted vs. Target



Model Accuracy and Sample Accuracy



Conclusions

Improvements

Future improvements could include adding more relevant features and applying advanced techniques like ensemble learning to boost model performance.

Key Learnings

This project highlights the importance of data preprocessing, feature engineering, and the value of model evaluation metrics in selecting the best model for deployment.