House Price Prediction Using Linear Regression

1. The main goal of this project is to create a Linear Regression model and see how well it works. This model is supposed to predict the prices of houses. We are going to use the California Housing dataset to do this. The idea is to use the Linear Regression model to figure out the prices of houses.

This task shows us how to do machine learning from start to finish. It includes getting the data looking at the data cleaning up the data teaching the model checking how well the model works and putting the model to use. The machine learning workflow is what this task is, about. We get to see the process of machine learning, which is really useful.

The project was done using Python and some other tools like pandas and scikit-learn. All the experiments for the project were carried out in a thing called Jupyter Notebook. The project used Python for all the work and the experiments, for the project were done in the Jupyter Notebook.

2. Dataset Description

The California Housing dataset is a real world dataset that has information about housing in the California districts. The California Housing dataset is actually about homes, in California.

Target Variable:

MedHouseVal – Median house value (in hundreds of thousands of dollars)

Input Features:

MedInc – Median income in the block group

HouseAge – Median house age

AveRooms – Average number of rooms per household

AveBedrms – Average number of bedrooms per household

Population – Population of the block group

AveOccup – Average occupancy

Latitude – Latitude location

Longitude – Longitude location

3. Exploratory Data Analysis (EDA)

We did an exploratory data analysis to see what the data looks like and how it is spread out in the dataset. This helped us understand the structure of the dataset and

the distribution of the data, in the dataset.

Key EDA Steps:

Checked dataset shape and feature names

Verified absence of missing values

Analyzed summary statistics using .describe()

Visualized distributions and correlations using plots

Observations:

The median income, which we will refer to as income has a big effect, on house prices. Median income is really closely linked to the cost of houses. When median income goes up house prices tend to go up and this shows that median income and house prices are connected in a big way. Median income is a factor that affects what people can afford to pay for a house and that is why median income is so important when it comes to house prices.

The location of a place, including its latitude and location features like longitude really affect the prices of things. Location features such as latitude and longitude have an impact on prices. When we talk about location features, like latitude and longitude we are talking about the location features that influence prices.

The dataset does not have any values. We looked at the dataset. It is complete. The dataset is fine it has all the values it needs.

Exploratory Data Analysis or EDA for short was really helpful in figuring out which features were the important. It also made sure that the dataset we were using was good enough, for Linear Regression. This was a deal because we wanted to make sure Linear Regression would work well with our dataset. EDA made that possible.

4. Data Preprocessing

So I want to tell you about the dataset, for the model. Before we started training the model we had to get the dataset ready. The dataset was prepared in a way.

We took the features, which are called X and the target variable, which is called y. We separated them. The features, X were put into one group and the target variable y was put into another group. This means that X and y are now apart, from each other.

The data was divided into two parts the training data and the testing data using an 80-20 split. This means that the dataset was split so that 80 percent of the data is used for training and 20 percent of the dataset is used for testing.

We used the training data to make the model work properly. The training data was really important to get the model right. We had to use the training data to fit the

model.

We used some testing data to see how well the model works. The testing data was really important to evaluate the model performance. We wanted to know if the model was doing a job so we used the testing data for that.

This makes sure that the model is tested on data that it has not seen before. The model is evaluated on this data.

5. Model Building

I used a Linear Regression model from the scikit-learn library to do this. The Linear Regression model, from scikit-learn is what I chose.

Steps:

Imported LinearRegression from sklearn.linear_model

Trained the model using training data

Generated predictions on the test dataset

I chose Linear Regression because it is simple and easy to understand. This makes Linear Regression a choice, for seeing how well a basic model works. Linear Regression is a starting point because it is straightforward and helps us understand the basics of model performance.

6. Model Evaluation

The model was tested with three measures to see how well it worked:

Mean Absolute Error (MAE): Measures average absolute prediction error

Root Mean Squared Error (RMSE): Penalizes larger errors more strongly

The R^2 Score is a measure that shows how well the model explains the variance in the data. It tells us how good the model is at predicting the data. The R^2 Score is very important because it helps us understand how well the model works. The model is good if the R^2 Score is high which means it explains the variance in the data well. We use the R^2 Score to see how well the model does its job, which's to explain the variance, in the data.

Results:

MAE: 0.53

RMSE: 0.74

R^2 Score: 0.57

Interpretation:

The model explains about 57 percent of the difference in house prices. This means that the model can account for a big part of why house prices are different. The model is talking about house prices. It is able to explain a lot of the variation in house prices. House prices are what the model is trying to understand. It does a pretty good job of explaining about 57 percent of the variation, in house prices.

Errors are reasonable for a simple linear model

The performance shows that we have a baseline model. This baseline model is a starting point. The baseline model seems to be working.

7. Model Saving and Prediction (Optional Task)

I saved the trained model using the pickle library. The model was saved so it can be used later. The pickle library is what I used to save the trained model.

A separate Python script was made to do some things. This Python script is a program that was written in the Python language. The Python script was created for a reason.

Load the saved model

Accept new input values

Predict house prices using the trained model

This shows that we can use a model more than once and it is easy to put it to work which is really important when we are using machine learning, in the world for actual things like real-world ML applications.

8. In this project we did a thing with machine learning to figure out house prices. We used something called Linear Regression to make predictions about how much houses cost. The goal of the project was to make a workflow, for machine learning that could predict house prices using Linear Regression.

The model did a good job and it showed that how much money people make and where they live are really important things to consider when trying to figure out how much houses cost. The housing prices can be predicted better when we look at the income and location features. The model was able to do this. That is why income and location are important when it comes to housing prices.

This task gave me hands on experience, with:

Data analysis

Model training

Evaluation

Model persistence

9. Future Improvements

We can make some things better. Here are a few ideas:

Using advanced models such as Random Forest or Gradient Boosting

Performing feature scaling and transformation

Hyperparameter tuning

Deploying the model as a web application using Flask or Streamlit

10. The. Technologies that I used

Python

Pandas

NumPy

Scikit-learn

Matplotlib & Seaborn

Jupyter Notebook