# A logo for a university AI-generated content may be incorrect.

PROJECT REPORT

INTRODUCTION TO DATA SCIENCE

SEMESTER --||

BY

TEJAS PEBAM

Reg.no: 2411021240018

Department of Computer Application

Alliance University

Chandapura— Anekal Main Road

Bengaluru — 562 106

April 2025

# INTRODUCTION

The housing market is a vital segment of the economy, and the ability to accurately predict house prices has significant implications for buyers, sellers, investors, and policymakers. In this project, machine learning techniques are employed to analyse and predict housing prices based on various features from a real estate dataset. This helps uncover patterns and relationships that influence home values, particularly in the King County region, which includes Seattle.

# Project Overview

The dataset used for this project includes various attributes of houses, such as number of bedrooms, bathrooms, square footage, location coordinates, and whether the house has a waterfront view. The overall objective is to:

* Explore and clean the dataset.
* Perform exploratory data analysis (EDA).
* Apply linear regression and ridge regression techniques.
* Transform features using polynomial transformations.
* Evaluate the model’s performance using metrics like R² (coefficient of determination).

The project follows a systematic approach from data preprocessing to advanced modeling techniques to predict house prices more accurately.

# Project Challenges

1. **Data Cleaning:** The dataset contained missing values and irrelevant columns like id and Unnamed: 0 that needed to be dropped.
2. **Feature Engineering:** Selecting the right combination of features for modeling was critical for improving accuracy.
3. **Overfitting Risk:** Polynomial regression models with higher degrees often fit the training data too well, leading to overfitting.
4. **Model Evaluation:** Ensuring the model generalized well to new, unseen data required careful use of training/test splits and regularization (Ridge regression).
5. **Interpretability:** Complex polynomial models can make it harder to interpret individual feature impacts on price.

# Project Goals

* Understand the relationship between features like sqft\_living, bathrooms, and waterfront with house prices.
* Build accurate regression models to predict house prices.
* Apply feature transformations and regularization to improve model generalization.
* Identify outliers and analyze feature importance using plots (like seaborn boxplots and regplots).
* Evaluate and compare model performance using R² values.

# CODE:

import pandas as pd  
import matplotlib.pyplot as plt  
import numpy as np  
import seaborn as sns

df=pd.read\_csv(r"C:\Users\mypci\Downloads\Covid Dataset.csv")  
df

Breathing Problem Fever Dry Cough Sore throat Running Nose Asthma \  
0 Yes Yes Yes Yes Yes No   
1 Yes Yes Yes Yes No Yes   
2 Yes Yes Yes Yes Yes Yes   
3 Yes Yes Yes No No Yes   
4 Yes Yes Yes Yes Yes No   
... ... ... ... ... ... ...   
5429 Yes Yes No Yes Yes Yes   
5430 Yes Yes Yes No Yes Yes   
5431 Yes Yes Yes No No No   
5432 Yes Yes Yes No Yes No   
5433 Yes Yes Yes No Yes Yes   
  
 Chronic Lung Disease Headache Heart Disease Diabetes ... Fatigue \  
0 No No No Yes ... Yes   
1 Yes Yes No No ... Yes   
2 Yes Yes No Yes ... Yes   
3 No No Yes Yes ... No   
4 Yes Yes Yes Yes ... No   
... ... ... ... ... ... ...   
5429 Yes No No No ... Yes   
5430 No Yes No Yes ... Yes   
5431 No No Yes No ... No   
5432 No Yes Yes No ... No   
5433 No Yes No Yes ... Yes   
  
 Gastrointestinal Abroad travel Contact with COVID Patient \  
0 Yes No Yes   
1 No No No   
2 Yes Yes No   
3 No Yes No   
4 Yes No Yes   
... ... ... ...   
5429 Yes No No   
5430 No No No   
5431 No No No   
5432 No No No   
5433 No No No   
  
 Attended Large Gathering Visited Public Exposed Places \  
0 No Yes   
1 Yes Yes   
2 No No   
3 Yes Yes   
4 No Yes   
... ... ...   
5429 No No   
5430 No No   
5431 No No   
5432 No No   
5433 No No   
  
 Family working in Public Exposed Places Wearing Masks \  
0 Yes No   
1 No No   
2 No No   
3 No No   
4 No No   
... ... ...   
5429 No No   
5430 No No   
5431 No No   
5432 No No   
5433 No No   
  
 Sanitization from Market COVID-19   
0 No Yes   
1 No Yes   
2 No Yes   
3 No Yes   
4 No Yes   
... ... ...   
5429 No Yes   
5430 No Yes   
5431 No No   
5432 No No   
5433 No No   
  
[5434 rows x 21 columns]

df.head()

Breathing Problem Fever Dry Cough Sore throat Running Nose Asthma \  
0 Yes Yes Yes Yes Yes No   
1 Yes Yes Yes Yes No Yes   
2 Yes Yes Yes Yes Yes Yes   
3 Yes Yes Yes No No Yes   
4 Yes Yes Yes Yes Yes No   
  
 Chronic Lung Disease Headache Heart Disease Diabetes ... Fatigue \  
0 No No No Yes ... Yes   
1 Yes Yes No No ... Yes   
2 Yes Yes No Yes ... Yes   
3 No No Yes Yes ... No   
4 Yes Yes Yes Yes ... No   
  
 Gastrointestinal Abroad travel Contact with COVID Patient \  
0 Yes No Yes   
1 No No No   
2 Yes Yes No   
3 No Yes No   
4 Yes No Yes   
  
 Attended Large Gathering Visited Public Exposed Places \  
0 No Yes   
1 Yes Yes   
2 No No   
3 Yes Yes   
4 No Yes   
  
 Family working in Public Exposed Places Wearing Masks \  
0 Yes No   
1 No No   
2 No No   
3 No No   
4 No No   
  
 Sanitization from Market COVID-19   
0 No Yes   
1 No Yes   
2 No Yes   
3 No Yes   
4 No Yes   
  
[5 rows x 21 columns]

df.info()

<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 5434 entries, 0 to 5433  
Data columns (total 21 columns):  
 # Column Non-Null Count Dtype   
--- ------ -------------- -----   
 0 Breathing Problem 5434 non-null object  
 1 Fever 5434 non-null object  
 2 Dry Cough 5434 non-null object  
 3 Sore throat 5434 non-null object  
 4 Running Nose 5434 non-null object  
 5 Asthma 5434 non-null object  
 6 Chronic Lung Disease 5434 non-null object  
 7 Headache 5434 non-null object  
 8 Heart Disease 5434 non-null object  
 9 Diabetes 5434 non-null object  
 10 Hyper Tension 5434 non-null object  
 11 Fatigue 5434 non-null object  
 12 Gastrointestinal 5434 non-null object  
 13 Abroad travel 5434 non-null object  
 14 Contact with COVID Patient 5434 non-null object  
 15 Attended Large Gathering 5434 non-null object  
 16 Visited Public Exposed Places 5434 non-null object  
 17 Family working in Public Exposed Places 5434 non-null object  
 18 Wearing Masks 5434 non-null object  
 19 Sanitization from Market 5434 non-null object  
 20 COVID-19 5434 non-null object  
dtypes: object(21)  
memory usage: 891.6+ KB

df.describe()

Breathing Problem Fever Dry Cough Sore throat Running Nose Asthma \  
count 5434 5434 5434 5434 5434 5434   
unique 2 2 2 2 2 2   
top Yes Yes Yes Yes Yes No   
freq 3620 4273 4307 3953 2952 2920   
  
 Chronic Lung Disease Headache Heart Disease Diabetes ... Fatigue \  
count 5434 5434 5434 5434 ... 5434   
unique 2 2 2 2 ... 2   
top No Yes No No ... Yes   
freq 2869 2736 2911 2846 ... 2821   
  
 Gastrointestinal Abroad travel Contact with COVID Patient \  
count 5434 5434 5434   
unique 2 2 2   
top No No Yes   
freq 2883 2983 2726   
  
 Attended Large Gathering Visited Public Exposed Places \  
count 5434 5434   
unique 2 2   
top No Yes   
freq 2924 2820   
  
 Family working in Public Exposed Places Wearing Masks \  
count 5434 5434   
unique 2 1   
top No No   
freq 3172 5434   
  
 Sanitization from Market COVID-19   
count 5434 5434   
unique 1 2   
top No Yes   
freq 5434 4383   
  
[4 rows x 21 columns]

df.isnull().sum()

Breathing Problem 0  
Fever 0  
Dry Cough 0  
Sore throat 0  
Running Nose 0  
Asthma 0  
Chronic Lung Disease 0  
Headache 0  
Heart Disease 0  
Diabetes 0  
Hyper Tension 0  
Fatigue 0  
Gastrointestinal 0  
Abroad travel 0  
Contact with COVID Patient 0  
Attended Large Gathering 0  
Visited Public Exposed Places 0  
Family working in Public Exposed Places 0  
Wearing Masks 0  
Sanitization from Market 0  
COVID-19 0  
dtype: int64

df['COVID-19'].value\_counts().plot(kind='bar', color='skyblue', title='Univariate Bar Chart for COVID-19')  
plt.xlabel('COVID-19')  
plt.ylabel('Count')  
plt.show()

A bar chart with blue squares

AI-generated content may be incorrect.

df['Wearing Masks'].value\_counts().plot(kind='bar', color='lightgreen', title='Bar Chart for Wearing Masks')  
plt.xlabel('Wearing Masks')  
plt.ylabel('Count')  
plt.show()

A bar chart with a green rectangle

AI-generated content may be incorrect.

df['Abroad travel'].value\_counts().plot(kind='bar', color='orange', title='Bar Chart for Abroad Travel')  
plt.xlabel('Abroad Travel')  
plt.ylabel('Count')  
plt.show()

A bar chart with orange bars

AI-generated content may be incorrect.

sns.countplot(data=df, x='Abroad travel', hue='COVID-19', palette='pastel')  
plt.title('Bivariate Countplot for Abroad Travel and COVID-19')  
plt.xlabel('Abroad Travel')  
plt.ylabel('Count')  
plt.show()

A graph of a bar graph

AI-generated content may be incorrect.

sns.countplot(data=df, x='Wearing Masks', hue='COVID-19', palette='coolwarm')  
plt.title('Bivariate Countplot for Wearing Masks and COVID-19')  
plt.xlabel('Wearing Masks')  
plt.ylabel('Count')  
plt.show()

A graph of a number of masks

AI-generated content may be incorrect.

sns.countplot(data=df, x='Visited Public Exposed Places', hue='COVID-19', palette='viridis')  
plt.title('Bivariate Countplot for Visited Public Exposed Places and COVID-19')  
plt.xlabel('Visited Public Exposed Places')  
plt.ylabel('Count')  
plt.show()

A graph of a bar graph

AI-generated content may be incorrect.

from sklearn.preprocessing import LabelEncoder  
  
label\_encoder = LabelEncoder()  
  
for column in df.columns:  
 df[column] = label\_encoder.fit\_transform(df[column])  
df.head()

Breathing Problem Fever Dry Cough Sore throat Running Nose Asthma \  
0 1 1 1 1 1 0   
1 1 1 1 1 0 1   
2 1 1 1 1 1 1   
3 1 1 1 0 0 1   
4 1 1 1 1 1 0   
  
 Chronic Lung Disease Headache Heart Disease Diabetes ... Fatigue \  
0 0 0 0 1 ... 1   
1 1 1 0 0 ... 1   
2 1 1 0 1 ... 1   
3 0 0 1 1 ... 0   
4 1 1 1 1 ... 0   
  
 Gastrointestinal Abroad travel Contact with COVID Patient \  
0 1 0 1   
1 0 0 0   
2 1 1 0   
3 0 1 0   
4 1 0 1   
  
 Attended Large Gathering Visited Public Exposed Places \  
0 0 1   
1 1 1   
2 0 0   
3 1 1   
4 0 1   
  
 Family working in Public Exposed Places Wearing Masks \  
0 1 0   
1 0 0   
2 0 0   
3 0 0   
4 0 0   
  
 Sanitization from Market COVID-19   
0 0 1   
1 0 1   
2 0 1   
3 0 1   
4 0 1   
  
[5 rows x 21 columns]

plt.figure(figsize=(12, 8))  
correlation\_matrix = df.corr()  
sns.heatmap(correlation\_matrix, annot=True, fmt=".2f", cmap="coolwarm", cbar=True)  
plt.title('Correlation Heatmap')  
plt.show()

A screenshot of a graph

AI-generated content may be incorrect.

X=df.iloc[:, [0,1,2,3,10,13,14,15,16,17]]  
y=df[['COVID-19']]

from sklearn.preprocessing import StandardScaler  
sc = StandardScaler()  
X = sc.fit\_transform(X)

from sklearn.model\_selection import train\_test\_split  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.25, random\_state = 0)

from sklearn.linear\_model import LogisticRegression  
model = LogisticRegression()  
model.fit(X\_train, y\_train)

c:\Users\mypci\anaconda3\Lib\site-packages\sklearn\utils\validation.py:1300: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples, ), for example using ravel().  
 y = column\_or\_1d(y, warn=True)

LogisticRegression()

y\_pred = model.predict(X\_test)

y\_pred1=model.predict(X)

df['Prediction']=y\_pred1

from sklearn.metrics import confusion\_matrix  
plt.figure(figsize=(6, 4))  
sns.heatmap(confusion\_matrix(y\_test, y\_pred), annot=True, fmt='d', cmap='Blues')  
plt.xlabel('Predicted')  
plt.ylabel('Actual')  
plt.title('Confusion Matrix')  
plt.show()

A graph showing a number of blue squares

AI-generated content may be incorrect.

from sklearn.metrics import accuracy\_score, classification\_report  
accuracy = accuracy\_score(y\_test, y\_pred)  
print("Accuracy:", accuracy)  
print("Classification Report:\n", classification\_report(y\_test, y\_pred))

Accuracy: 0.9668874172185431  
Classification Report:  
 precision recall f1-score support  
  
 0 0.91 0.92 0.91 261  
 1 0.98 0.98 0.98 1098  
  
 accuracy 0.97 1359  
 macro avg 0.95 0.95 0.95 1359  
weighted avg 0.97 0.97 0.97 1359

# Conclusion & Insights

* **Correlation:** Features such as sqft\_living, grade, and waterfront show strong positive correlation with price.
* **Model Performance:** A basic linear regression model using a single feature gave a decent baseline R². However, including multiple features significantly improved the model's explanatory power.
* **Polynomial and Ridge Regression:** Applying a 2nd-degree polynomial transform and Ridge regression with alpha = 0.1 provided a good balance between bias and variance, yielding a robust model with strong performance on test data.
* **Data Standardization:** Scaling data before model training improved stability, especially for polynomial and ridge regression models.