Regression and ANOVA

Final Project
MATH 533

Authors: *Ling Fei Zhang and Sevag Baghdassarian*

Instructor:	*Mehdi Dagdoug*
Department:	Mathematics
Date:	December 9, 2023

Contents

1 Introduction

Linear regression is a tool used in a wide array of statistical applications. In Python, there are many different packages providing implementations for the regression technique, and they may be overwhelming for some mathematicians coming from R into Python. In this project, we design a linear regression package bridging the gap between R and Python. Our package provides the basic functionalities of fitting a linear model and providing summaries in more or less the same way as done in R. In the following sections, you will learn how to use the package, including the different functionalities provided, as well as an example application on an existing dataset.

2 Functionality

The package offers the following primary functions:

- fit(X: NumPy array, y: NumPy array): internally computes and stores the linear regression weights for a model fitted on X and y (including the intercept)
- predict(X: NumPy array): internally stores the target predictions and residuals for the provided X data
- plotting functions: different functions to plot the regression lines and their confidence intervals
- metrics functions: functions that internally store various metrics of the model (update_metrics updates all of them automatically)
- summarize(): generates a summary of the model, much like in R

By default, the plots and metrics that are computed by the model are stored in the figure folder. Internally, the package uses NumPy matrix functions to compute the weights. The package assumes the intercept is not included in the covariates ands temporarily adds a column of ones to X in order to include the intercept in the weight computations.

3 Usage

In order to use the model, the covariate matrix and the target vector must be converted to NumPy arrays. If .csv files are being used for the datasets, it is recommended to first import the data into a Pandas dataframe using the read_csv function. Afterwards, the covariates and targets may be extracted and converted to NumPy arrays using the to_numpy function. The main.py file provides an example application of these steps.

4 Toy Dataset Overview

The dataset contains the following covariates: [id, date, price, bedrooms, bathrooms, sqft living, sqft lot, floors, waterfront, view, condition, grade, sqft above, sqft basement, yr built, yr renovated, zipcode, lat, long, sqft living15, and sqft lot15]. While each of these variables holds potential insights, certain attributes are deemed irrelevant for our analysis. Specifically, the variables id, data, waterfront, view, condition, zipcode, lat, long are excluded from our study. Additionally, to focus our analysis on a more representative range of house prices, we have excluded houses with prices exceeding $2 million.

5 Objective of the Analysis

Our primary objective is to understand the intricate relationship between the selected covariates and the housing prices. To achieve this, we employ three distinct Linear Regression models. The models include a univariate model, which explores the impact of `sqft_living` on house prices; a bivariate model, which explores the impact of `sqft_living, yr_built` on prices; and a multivariate model, which explores the effect of all interested covariates on prices.

6 Model Visualizations

6.1 Univariate Model

In the univariate case, we measure the relationship between the size of the living room versus the price of the house. Below show the the result of our model. In this plot, we've also added a confidence interval of 95%.

Figure 1: Univariate Linear Regression

6.2 Bivariate Model

In the bivariate case, we measure the relationship between the size of the living room and the built year versus the price of the house. Below we show the relationship of the covariates with the housing price via a 3D plot.

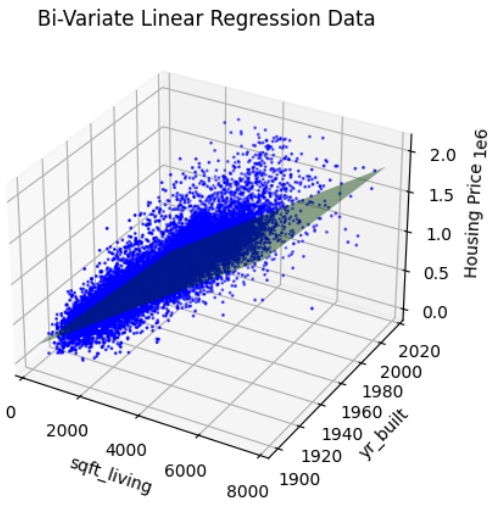

Figure 2: Bivariate Linear Regression

7 Model Summaries

We display our models summary as tables below, following the format seen in R.

7.1 Univariate Model

(a) Residuals

Min	1Q	Median	3Q	Max
-949152.5	-137658.7	-22279.9	100540.6	1354243.5

(b) Coefficients

Coef	Estimate	Std. error	t value	p value
Intercept	54408.71	3717.16	14.64	0.0000e+00
x1	226.57	1.67	135.73	0.0000e+00

Residual standard error: 211022.60 on 21413 degrees of freedom
R-squared: 0.46, Adjusted R-squared: 0.46
F-statistic: 18422.07 on 1 and 21413 DF, p-value: 1.1102230246251565e-16

(c) Other Metrics

Sigma naive	Sigma cor	MSE	MAE	RMSE
44526380006.3	44528459317.9	44526380006.3	154679.9	211012.7

Table 1: Univariate Linear Regression Summary

7.2 Bivariate Model

(a) Residuals

Min	1Q	Median	3Q	Max
-1096853.0	-128232.1	-20094.3	99775.8	1406782.3

(b) Coefficients

Coef	Estimate	Std. error	t value	p value
Intercept	3632942.1	98913.89	36.73	0.0000e+00
x1	247.69	1.72	143.81	0.0000e+00
x2	-1837.58	50.76	-36.2	0.0000e+00

Residual standard error: 204851.25 on 21412 degrees of freedom
R-squared: 0.49, Adjusted R-squared: 0.49
F-statistic: 10429.69 on 2 and 21412 DF, p-value: 1.1102230246251565e-16

(c) Other Metrics

Sigma naive	Sigma cor	MSE	MAE	RMSE
41958154700.2	41962073642.4	41958154700.2	149916.3	204836.9

Table 2: Bivariate Linear Regression Summary

7.3 Multi Variate Model

(a) Residuals

Min	1Q	Median	3Q	Max
-10467199.4	-49975.4	118974.6	418096.6	2028580.6

(b) Coefficients

Coef	Estimate	Std. error	t value	p value
Intercept	-2.0001e+05	2.6942e+05	-7.4237e-01	4.5787e-01
x1	9.8760e-01	1.6195e-02	6.0982e+01	0.0000e+00
x2	3.5252e+05	3.9604e+03	8.9012e+01	0.0000e+00
x3	-6.4459e+05	6.8439e+03	-9.4185e+01	0.0000e+00
x4	4.2554e+02	8.3863e+07	5.0742e-06	1.0000e+00
x5	6.0819e-02	9.9083e-02	6.1382e-01	5.3934e-01
x6	1.0715e+05	7.3842e+03	1.4511e+01	0.0000e+00
x7	-1.2752e+03	4.7668e+03	-2.6752e-01	7.8907e-01
x8	-3.9027e+02	8.3863e+07	-4.6537e-06	1.0000e+00
x9	-3.6567e+02	8.3863e+07	-4.3603e-06	1.0000e+00
x10	3.5100e-09	1.4305e+02	2.4537e-11	1.0000e+00
x11	4.5284e-11	7.5493e+00	5.9985e-12	1.0000e+00
x12	-8.8343e-12	7.1257e+00	-1.2398e-12	1.0000e+00
x13	-7.7981e-14	1.5133e-01	-5.1531e-13	1.0000e+00

Residual standard error: 416593.71 on 21401 degrees of freedom
R-squared: -1.09, Adjusted R-squared: -1.10
F-statistic: 1850.62 on 13 and 21401 DF, p-value: 1.1102230246251565e-16

(c) Other Metrics

Sigma naive	Sigma cor	MSE	MAE	RMSE
173436864560.7	173542213558.0	173436864560.7	318062.1	416457.5

Table 3: Multi Variate Linear Regression Summary

8 Hypothesis Testing

We also performed hypothesis testing on our models, which are summarized as tables below.

8.1 Univariate Model

Coefficient	t-value	p-value	Reject Null Hypothesis?
Intercept	14.64	0.0	True
x1	135.73	0.0	True

Table 4: Hypothesis Testing in Univariate Linear Regression

8.2 Bivariate Model

Coefficient	t-value	p-value	Reject Null Hypothesis?
Intercept	36.73	0.0	True
x1	143.81	0.0	True
x2	-36.2	0.0	True

Table 5: Hypothesis Testing in Bivariate Linear Regression

8.3 Multi Variate Model

Coefficient	t-value	p-value	Reject Null Hypothesis?
Intercept	-0.74	0.46	False
x1	60.98	0.0	True
x2	89.01	0.0	True
x3	-94.19	0.0	True
x4	0.0	1.0	False
x5	0.61	0.54	False
x6	14.51	0.0	True
x7	-0.27	0.79	False
x8	-0.0	1.0	False
x9	-0.0	1.0	False
x10	0.0	1.0	False
x11	0.0	1.0	False
x12	-0.0	1.0	False
x13	-0.0	1.0	False

Table 6: Hypothesis Testing in Multi Variate Linear Regression

9 Conclusion

In conclusion, the project offers a basic implementation of linear regression similar to the way it is presented in R. An example application on the toy dataset shows how to use the package on a dataset from a .csv file, which is the kind of data one works with in a lot of cases. One of the challenges in the project was keeping everything in order and organized, since many of the functions are interdependent (one erroneous function would make everything erroneous). In the future, it would be worth exploring and implementing additional kinds of regression into the model, and incorporating finer machine learning techniques.