T.C. ESKİŞEHİR TECHNICAL UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

BIM309 – Artificial Intelligence

HOMEWORK IV - Report

House Price Prediction Using Linear Regression in Python

Kaouther MOUHEB

99926527616

I. Introduction

This is a brief report describing the Linear Regression method used for making predictions for continuous variables and an example application of the method for predicting house prices based on the given dataset.

II. Linear Regression

• **The aim** is to analyze the specific relationships between two or more variables and thus gain information (predict) about one through knowing the values of the others.

• **Regression:** A statistical measure that attempts to determine the strength of the relationship between one dependent variable y (the label) and a series of independent variables {x_1, x_2, …, x_n} (the features).

• **Linear Regression:** Assumes a linear relationship between the dependent variable y and the set of independent variables {x_1, x_2, …, xn} written as:

$$y_p = f(x_i) = w_0 + \sum_{i=1}^{n} w_i x_i \quad \text{where i} \in \{1, 2, \ldots, n\} \qquad (1)$$

Where y_p is the predicted value and w_i's are called the coefficients of the model and n is the dimension (number of features). Learning the regression model is finding the coefficients that minimize a loss (or error) function.
In ordinary least squares linear regression, the error function is the residual sum of squares given as:

$$error = \frac{1}{2n} \sum_{i=1}^{n} (y_i - f(x_i))^2 \quad \text{where i} \in \{1, 2, \ldots, n\} \qquad (2)$$

Where

- y_i is the real value of the dependent variable for the i^{th} data entry.
- $f(x_i)$ is the value predicted by the linear regression model with the current coefficients.
- n the number of entries in the training set.
- x_i is the feature vector for the i^{th} data entry

The unique solution for w_i's that minimize the residual sum of squares error function is found to be:

$$W = (X^T X)^{-1} X^T y \qquad (3)$$

Where W is the vector of optimal coefficients, X is the matrix of features, y is the vector of expected labels.

Pros: Only needs a training dataset, no need to specify a convergence rate. No iteration needed

Cons: Only works if $X^T X$ is invertible. Slow $O(n^3)$ with n the number of features.

III. Evaluation Metrics:

There exist multiple evaluation metrics to measure the performance of Linear Regression. One of which is the Root Mean Squared Error (RMSE) given by the formula:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - f(x_i))^2}{n}} \quad (4)$$

Where y_i is the real value of the dependent variable for the i^{th} data entry and $f(x_i)$ is the value predicted by the linear regression model, and n the number of entries in the test set, and x_i is the feature vector for the i^{th} data entry

IV. Implementation:

In this work, we trained a linear regression model on a given dataset for house prices.

• First, we create a function to preprocess the data. This function takes the path to the dataset file as an input parameter and returns 4 variables:

- **train_features**: a matrix that contains the values of the independent variables for the training set namely, lot_area, living_area, num_floors, num_bedrooms, num_bathrooms, waterfront, year_built, and year_renovated.
- **train_labels**: a vector that contains the expected value of the dependent variable namely the 'price' for each training data entry.
- **test_features**: a matrix that contains the values of the independent variables for the test set namely, lot_area, living_area, num_floors, num_bedrooms, num_bathrooms, waterfront, year_built, and year_renovated.
- **test_labels**: a vector that contains the expected value of the dependent variable namely the 'price' for each test data entry.

To create the previously mentioned variables we used the **Pandas** library.

- First, we read the whole data set to a pandas DataFrame using pandas' **read_csv()** function. This function detects the data types automatically.
- Then, we divide the data into **features matrix** (lot_area, living_area, num_floors, num_bedrooms, num_bathrooms, waterfront, year_built, and year_renovated) by taking the last 8 columns of the DataFrame and **labels vector** (price) by taking the first column of the DataFrame. The first 5 rows for each variable are shown in Figure 1.
- After that, we find the split point to split the dataset to 80% training and 20% test sets. We calculate this using the following code:

  *split_point = (int(round(len(labels) * 0.8)))*
- Next, we divide the labels and features where the entries from 0 to split_point form the training data, and the rest forms the test data. (For our dataset we will have 17290 rows in the training set and 4323 rows in the test set).

• Next, we create a function to build, train, and test the linear regression model using the **LinearRegression** class of the **sklearn.linear_model** package, and **mean_squared_error** function of the **sklearn.metrics** package. We used **scikit-learn** version 0.23.2 for this task.

This function takes the training and test data as input and returns the regression model and the corresponding RMSE value.

- First, we create an instance of the **LinearRegression** class to initialize the model
- We train the model using the training features and labels as input parameters to the **fit()** function of the LinearRegression class.
- To test the model, we calculate predictions of the price for the test features using the **predict()** function of the LinearRegression class.
- Finally, we calculate the RMSE value using the **mean_squared_error()** function by using the predictions and test labels as input parameters and setting the **squared** parameter to **False**.

• Finally, we call the 2 previously created functions in the main function and print the results (the **model coefficients** and the **RMSE** value obtained). For robustness, we handle the **FileNotFoundError** and **KeyError** using **try-except blocks** to show error messages if the data file is not found or does not match the format given in the requirements (eg. The price column missing).

	lot_area	living_area	num_floors	num_bedrooms	num_bathrooms	waterfront	year_built	year_renovated
0	5650	1180	1.00000	3	1.00000	0	1955	0
1	7242	2570	2.00000	3	2.25000	0	1951	1991
2	10000	770	1.00000	2	1.00000	0	1933	0
3	5000	1960	1.00000	4	3.00000	0	1965	0
4	8080	1680	1.00000	3	2.00000	0	1987	0
5	101930	5420	1.00000	4	4.50000	0	2001	0

0	221900
1	538000
2	180000
3	604000
4	510000
5	1225000

Figure 1: Sample of the features matrix (left) and labels vector (right)

V. Results:

```
Model Coefficients:
 [-2.62033217e-01  2.96203179e+02  3.90060173e+04 -5.65288367e+04
  6.05965720e+04  7.34424607e+05 -3.31586028e+03  9.02662444e+00]
RMSE = 244520.35
```

Figure 2: Resulting model coefficients and Root Mean Squared Error value

References:

• Lecture notes
• https://www.slideshare.net/Tech_MX/linear-regression-14155467
• https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html
• https://sciencing.com/calculate-rmsd-5146965.html
• https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html