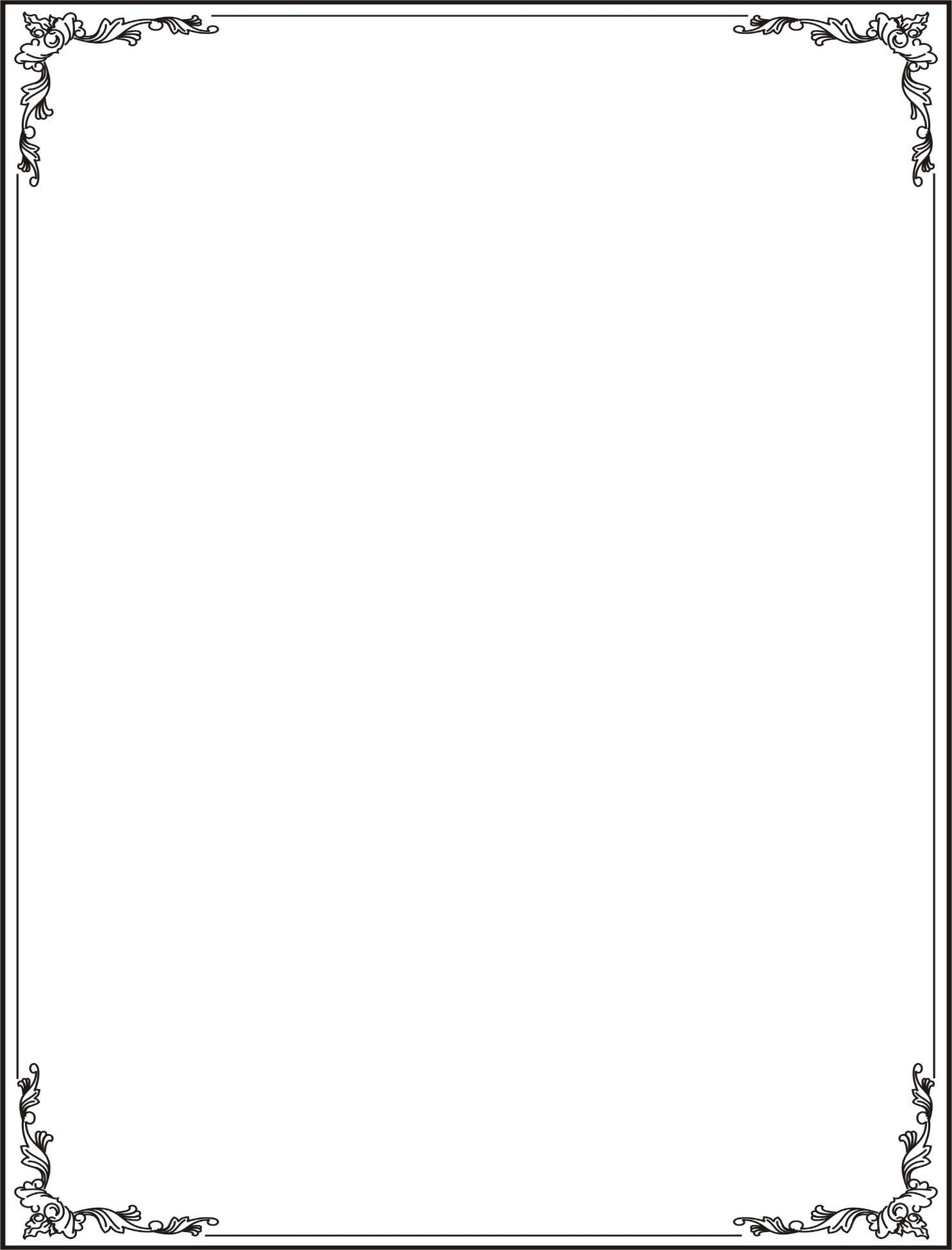
**VIETNAM NATIONAL UNIVERSITY - HO CHI MINH CITY**

**UNIVERSITY OF INFORMATION TECHNOLOGY**

-----🙞🙜🕮🙞🙜-----



**FINAL PROJECT**

**Topic: Building an application to   
forecast house prices**

**Subject: Business Data Analytics**

**Theory Lecturer: *PhD*. *Nguyễn Đình Thuân***

**Practical instructor: *Nguyễn Minh Nhựt***

**Class: *IS403.M21.HTCL***

**Students perform:**

***1. Hoàng Nhật Trung*** ***– 19522421***

***2. Phan Thành Bảo Trọng – 19522411***

***3.*** ***Nguyễn Thanh Sang – 19522123***

TEACHER'S COMMENTS

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

Table of Contents

[A. Member Introduction 1](#_Toc106174876)

[1. All members of the group 1](#_Toc106174877)

[2. Team Member Evaluation 1](#_Toc106174878)

[B. Introduction to the topic 3](#_Toc106174879)

[1. The reason for choosing the topic 3](#_Toc106174880)

[2. Data overview 3](#_Toc106174881)

[3. Attribute Description 4](#_Toc106174882)

[4. Describe the research problems 4](#_Toc106174883)

[5. Tools and Libraries 5](#_Toc106174884)

[C. Research Methods 6](#_Toc106174885)

[1. Applied Algorithms 6](#_Toc106174886)

[**1.1 Support Vector Regression** 6](#_Toc106174887)

[**1.2 Decision Tree Regression** 7](#_Toc106174888)

[**1.3 Ridge regression** 8](#_Toc106174889)

[**1.4 LassoLars regression** 10](#_Toc106174890)

[D. Results and analysis 13](#_Toc106174891)

[1. Data analysis 13](#_Toc106174892)

[2. Data preprocessing 19](#_Toc106174893)

[3. Algorithm execution 23](#_Toc106174894)

[4. Comparison 32](#_Toc106174895)

[5. Application Demo 33](#_Toc106174896)

[E. Summary 36](#_Toc106174897)

[1. General assessment 36](#_Toc106174898)

[2. Advantages 37](#_Toc106174899)

[3. Disadvantages 37](#_Toc106174900)

[4. Development 37](#_Toc106174901)

[F. References 38](#_Toc106174902)

LIST OF FIGURES

[Figure 1. 1: Import libraries 13](#_Toc106177387)

[Figure 1. 2: Read csv file 13](#_Toc106177388)

[Figure 1. 3: Remove unnecessary columns 13](#_Toc106177389)

[Figure 1. 4: Check data information 14](#_Toc106177390)

[Figure 1. 5: View data information 14](#_Toc106177391)

[Figure 1. 6: Remove m2 and convert to float 15](#_Toc106177392)

[Figure 1. 7: Replace the value in the column Giá 15](#_Toc106177393)

[Figure 1. 8: Split The price column string and replace the noise value 15](#_Toc106177394)

[Figure 1. 9: Convert to float 15](#_Toc106177395)

[Figure 1. 10: Replace Giá column 15](#_Toc106177396)

[Figure 1. 11: Value Separation 16](#_Toc106177397)

[Figure 1. 12: Assign value to column Day, Month, Year 16](#_Toc106177398)

[Figure 1. 13: Make a copy 16](#_Toc106177399)

[Figure 1. 14: Convert string to number 16](#_Toc106177400)

[Figure 1. 15: Khảo sát độ tương đồng giữa các cột 17](#_Toc106177401)

[Figure 1. 16: Visual similarity 18](#_Toc106177402)

[Figure 1. 17: Remove the Ngày column 18](#_Toc106177403)

[Figure 1. 18: Export csv file 18](#_Toc106177404)

[Figure 2. 1: Import necessary libraries 19](#_Toc106177442)

[Figure 2. 2: Import csv file 19](#_Toc106177443)

[Figure 2. 3: Check for null percent 20](#_Toc106177444)

[Figure 2. 4: Remove direction column 20](#_Toc106177445)

[Figure 2. 5: Remove the N/A value 20](#_Toc106177446)

[Figure 2. 6: Check for null percent 21](#_Toc106177447)

[Figure 2. 7: Test data 21](#_Toc106177448)

[Figure 2. 8: Check value groups 22](#_Toc106177449)

[Figure 2. 9: Remove lines with a Giấy tờ pháp lý column equal to 1 22](#_Toc106177450)

[Figure 2. 10: Remove values < 60,000,000 and convert to billion 22](#_Toc106177451)

[Figure 2. 11: Eliminate rows with Diện tích và Số tầng 22](#_Toc106177452)

[Figure 2. 12: Export csv file 22](#_Toc106177453)

[Figure 3. 1: Proceed to import the necessary libraries 23](#_Toc106177477)

[Figure 3. 2: Proceed to read the data 23](#_Toc106177478)

[Figure 3. 3: Check data information 24](#_Toc106177479)

[Figure 3. 4: Create a mapping dictionary 25](#_Toc106177480)

[Figure 3. 5: Split column, split test and train 25](#_Toc106177481)

[Figure 3. 6: Run the SVR algorithm 26](#_Toc106177482)

[Figure 3. 7: Predict and save results 26](#_Toc106177483)

[Figure 3. 8: Calculate the RMSE of the SVR algorithm 26](#_Toc106177484)

[Figure 3. 9: Calculate R square of SVR algorithm 26](#_Toc106177485)

[Figure 3. 10: Run the DTR algorithm 26](#_Toc106177486)

[Figure 3. 11: Make predictions 27](#_Toc106177487)

[Figure 3. 12: Calculating the RMSE of the DTR algorithm 27](#_Toc106177488)

[Figure 3. 13: Calculate R square of DTR algorithm 27](#_Toc106177489)

[Figure 3. 14: Run the LCV algorithm 27](#_Toc106177490)

[Figure 3. 15: Make predictions 27](#_Toc106177491)

[Figure 3. 16: Calculating the RMSE of the LCV algorithm 27](#_Toc106177492)

[Figure 3. 17: Calculate R square of LCV algorithm 28](#_Toc106177493)

[Figure 3. 18: Run Ridge algorithm 28](#_Toc106177494)

[Figure 3. 19: Make predictions 28](#_Toc106177495)

[Figure 3. 20: Calculating the RMSE of the Ridge algorithm 28](#_Toc106177496)

[Figure 3. 21: Calculate the R square of the Ridge algorithm 29](#_Toc106177497)

[Figure 3. 22: Compare RMSE 29](#_Toc106177498)

[Figure 3. 23: Compare R square 30](#_Toc106177499)

[Figure 4. 1: Compare RMSE 31](#_Toc106177500)

# **A. Member Introduction**

## **1. All members of the group**

**Group 15**

|  |  |  |  |
| --- | --- | --- | --- |
| Ordinal Numbers | First and Last name | Student Id | Position |
| 1 | Hoàng Nhật Trung | 19522421 | Leader |
| 2 | Phan Thành Bảo Trọng | 19522411 | Member |
| 3 | Nguyễn Thanh Sang | 19522123 | Member |

## **2. Team Member Evaluation**

|  |  |  |  |
| --- | --- | --- | --- |
| Members  Task | Hoàng Nhật Trung | Phan Thành Bảo Trọng | Nguyễn Thanh Sang |
| Create and write part A | **✓** | **✓** |  |
| Create and write part B |  | **✓** | **✓** |
| Create and write part C | **✓** | **✓** |  |
| Create and write part D | **✓** | **✓** |  |
| Create and write part E | **✓** | **✓** |  |
| Create and write part F | **✓** | **✓** |  |
| Find references | **✓** | **✓** | **✓** |
| Find documents to analyze | **✓** | **✓** | **✓** |
| Assign work | **✓** |  |  |
| Edit fonts and layouts |  | **✓** |  |
| Build apps | **✓** | **✓** |  |
| Preprocessing, analysis, running algorithms on python | **✓** |  |  |

# **B. Introduction to the topic**

## **1. The reason for choosing the topic**

Today, the real estate market plays an extremely important role in the development of modern society. They spread to more than 40 other important sectors of the economy and became a bridge for other markets. The buying and selling of land or houses between people has become more common and happens more often. Our grandfather once had the sentence: "Settle down with a house and a job". Having a house to live in will help us focus more on developing our career and family. However, the value of the house that people always want to buy will never stay at a number that will fluctuate wildly. That's why the house price prediction application was born to help users do those things, so the question is how can we assess the value of a house and by what factors to be able to do that? give the most accurate figure of the value of that house. To answer the above questions, my group decided to choose the topic "Building an application to forecast house prices" to research and learn through learned regression models and several scientific articles. that the group discovered.

## **2. Data overview**

House price prediction data in Hanoi is crawled and pre-processed by our team.

**Data sources:** <http://www.laydulieu.com/nha-dat/?model=0&type=0&province=01&district=&ward=&address=&fprice=&tprice=&fdate=&tdate=&field%5Bdien_tich%5D=&field%5Bhuong%5D=&field%5Btang%5D=&field%5Bphong%5D=&field%5Bnha_ve_sinh%5D=&field%5Bgiay_to_phap_ly%5D=&phone=>

## **3. Attribute Description**

Data includes 14 columns and 37500 rows

* Quận/Huyện: Which district is the house in?
* Phường/Xã: Which ward or commune does the house belong to?
* Đường, khu vực: The house is located on which street, which area?
* Giá: House price
* Ngày đăng: Date the house was posted for sale
* Diện tích: Area of ​​the house
* Hướng: Direction of the house
* Số tầng: Number of floors of the house
* Số phòng: Number of rooms in the house
* Nhà vệ sinh: Number of toilets of the house
* Giấy tờ pháp lý: Legal documents of the house
* Nhà: Type is house
* Cần Bán: House for sale
* Hà Nội: The location of the house

## **4. Describe the research problems**

Based on the properties available in the dataset, proceed to build an application to forecast house prices through a number of learned regression models, applying some algorithms:

* Support Vector Regression
* Decision Tree Regression
* LassoLars regression
* Ridge regression

## **5. Tools and Libraries**

Programming language: Python.

Libraries:

%matplotlib inline

import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

import seaborn as sns

from sklearn.svm import SVR

from sklearn.tree import DecisionTreeRegressor

from sklearn.linear\_model import LassoLarsCV

from sklearn.linear\_model import Ridge

from sklearn.model\_selection import train\_test\_split

from sklearn import linear\_model

from sklearn.metrics import mean\_squared\_error

from math import sqrt

from sklearn.metrics import r2\_score

from cProfile import label

from hashlib import new

import tkinter as tk

from tkinter import \*

from tkcalendar import DateEntry

# **C. Research Methods**

## **1. Applied Algorithms**

### **1.1 Support Vector Regression**

* Support Vector Machine[1] is a supervised learning algorithm that can solve both regression and classification problems.
* Support Vector Regression uses the principles of SVM in the classification problem, but margin plays a different role. The goal of the algorithm is to find the Hyperplane[2] containing the maximum training set inside margin.
* The steps of the algorithm:
* Step 1: Choose kernel function [3]
* Step 2: Create Correlation Matrix based on training data



* epsilon(ε): is the regularizer
* δi,j: is the kroeniker delta function
* Step 3: Find vector **α**

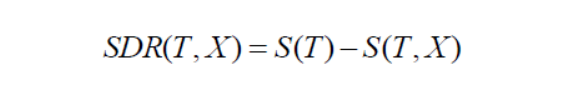


* y: vector values that correspond to the training set.
* K: correlation matrix.
* α: Unknown set that needs to be solved.
* Step 4: Use model to predict

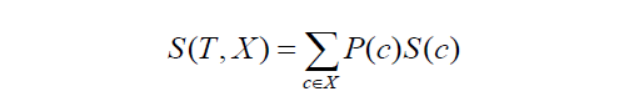


### **1.2 Decision Tree Regression**

* Decision Tree is a supervised learning algorithm that can solve both regression and classification problems.
* Decision Tree algorithm consists of 2 steps: Training and prediction, the result of training is that the algorithm will build a decision tree with decision nodes and leaf nodes.
* ID3 Algorithm[5] is one of the algorithms to build Decision Tree. For regression problem, Decision Tree has leaf nodes of numeric type, so when using algorithm ID3 we have to replace **Information Gain** or **Entropy** with **Standard Deviation Reduction**.
* Formula to calculate Standard Deviation Reduction:



* T: Prediction attribute
* X: Decisive Attribute
* S(X): Standard deviation of the attribute X
* S(T, X): Standard deviation of the X and T attributes
* To calculate the standard deviation for two attributes we use the formula



* P(c): Probability c occurs on the entire data
* S(c): Standard deviation of c
* c: Value group of attribute X
* The steps of the algorithm:
* Step 1: Calculate the standard deviation of the predictor attribute.
* Step 2: Calculate the Standard Deviation Reduction of each attribute.
* Step 3: Select the attribute with the largest Standard Deviation Reduction value as the decision node.
* Step 4: Repeat step 1 until all branches reach the leaf node.

### **1.3 Ridge regression**

* Linear regression model estimates beta by minimizing:

A picture containing calendar

Description automatically generated

* Ridge regression is a variant of linear regression to overcome overfitting. Minimal Ridge Regression:

A white paper with black writing

Description automatically generated with low confidence

* With λ (λ>0) called the tuning parameter of the model
* Like linear regression, Ridge regression tries to fit the data by minimizing RSS
* However, by adding the 2nd expression:



has large value when β value is large, Ridge regression will favor the model with smaller β value. Therefore, variables that are not significant will have the coefficient β decrease to 0.

* The parameter λ represents the priority between two quantities:
* λ large: Most beta coefficients decrease to 0 => underfit
* λ small: Same as normal linear regression => overfit
* The selection of the parameter λ is extremely important, selected by the method of cross-validation.

### **1.4 LassoLars regression**

* Ridge regression keeps all the variables, only reducing the coefficient estimate to 0.
* Another model will help select significant variables, other variables are set to zero: lasso
* Lasso regression will minimize the expression

Diagram

Description automatically generated

* Lasso uses l1-penalty instead of l2-penalty like Ridge regression
* With the formula:

Diagram

Description automatically generated

* Lasso model will be:

A picture containing diagram

Description automatically generated

* With the formula:

Diagram

Description automatically generated

* Lasso model will be:

Diagram

Description automatically generated

# **D. Results and analysis**

## **1. Data analysis**

Step 1: Import the necessary libraries for analysis

Graphical user interface, text, application, chat or text message

Description automatically generated

Figure 1. 1: Import libraries

Step 2: Read the csv file to be analyzed

A picture containing text

Description automatically generated

Figure 1. 2: Read csv file

Step 3: Proceed to remove the columns: Nhà, Cần bán, Hà Nội because it makes no sense in data analysis

Graphical user interface, text, application

Description automatically generated

Figure 1. 3: Remove unnecessary columns

Step 4: Check data set information

Graphical user interface, text

Description automatically generated

Figure 1. 4: Check data information

Step 5: Proceed to view the current dataset

Graphical user interface, text, application

Description automatically generated

Figure 1. 5: View data information

Step 6: Remove m2 in the Diện tích column and then convert the Diện tích column to float

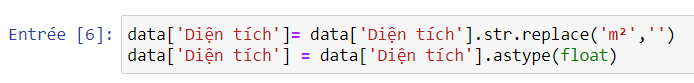


Figure 1. 6: Remove m2 and convert to float

Step 7: In the Giá column with a comma value, we replace it with a dot, billion becomes 1000000000 and million becomes 1000000

A picture containing chart

Description automatically generated

Figure 1. 7: Replace the value in the column Giá

Step 8: Perform string splitting from Giá column and replace noisy values

Text

Description automatically generated

Figure 1. 8: Split The price column string and replace the noise value

Step 9: Convert the cut string from string to float

Graphical user interface

Description automatically generated with low confidence

Figure 1. 9: Convert to float

Step 10: Replace Giá column with numeric value



Figure 1. 10: Replace Giá column

Step 11: Split the value in the Date column into a sub-array at the positions where the "/" sign appears



Figure 1. 11: Value Separation

Step 12: Based on the above split array, assign the positions in the array to the column Day, Month, Year, respectively



Figure 1. 12: Assign value to column Day, Month, Year

Step 13: Make a copy of the dataframe after processing

Text

Description automatically generated with medium confidence

Figure 1. 13: Make a copy

Step 14: Convert string columns to numbers to calculate similarity

A picture containing text

Description automatically generated

Figure 1. 14: Convert string to number

Step 15: Survey the similarity between columns with each other using Pearson's formula.

Incorporating seaborn graph representation library to visualize results using heatmap graphs. The variable annot=True helps to display similarity values in the resulting graph.

Based on the similarity matrix, the similarity between the Giá column and the other columns is non-zero, but the Ngày column has a rather low similarity of -0.0033, so we will remove this column.

Chart, treemap chart

Description automatically generated

Figure 1. 15: Khảo sát độ tương đồng giữa các cột

Step 16: Use the matplot library to visualize the similarity of columns and Giá columns

Chart, histogram

Description automatically generated

Figure 1. 16: Visual similarity

Step 17: Remove the Ngày column



Figure 1. 17: Remove the Ngày column

Step 18: Export csv file from dataframe copy



Figure 1. 18: Export csv file

## **2. Data preprocessing**

Step 1: Proceed to import the necessary libraries

Graphical user interface, text, application

Description automatically generated

Figure 2. 1: Import necessary libraries

Step 2: Import a csv file that needs preprocessing with 37500 lines and 13 columns

Graphical user interface, text, application

Description automatically generated

Figure 2. 2: Import csv file

Step 3: Check for null percent of each column

Graphical user interface, text

Description automatically generated

Figure 2. 3: Check for null percent

Step 4: The Direction column is removed because the percentage of nulls in this column is too high

A picture containing graphical user interface

Description automatically generated

Figure 2. 4: Remove direction column

Step 5: Remove the N/A value

Text

Description automatically generated with low confidence

Figure 2. 5: Remove the N/A value

Step 6: Perform null percent check in columns again after removing Direction column and N/A values

Graphical user interface, text

Description automatically generated

Figure 2. 6: Check for null percent

Step 7: Checking the data again after removing the Hướng column and the N/A values, we see that the data is only 15042 rows and 12 columns.

Graphical user interface, table

Description automatically generated with medium confidence

Figure 2. 7: Test data

Step 8: Checking the value groups of Giấy tờ pháp lý column, we see that the value 1 is an abnormal value

Graphical user interface, application, Word

Description automatically generated

Figure 2. 8: Check value groups

Step 9: Remove lines with a Giấy tờ pháp lý column equal to 1A picture containing graphical user interface

Description automatically generated

Figure 2. 9: Remove lines with a Giấy tờ pháp lý column equal to 1

Step 10: Based on [6], the house price in Hanoi is usually between $3,000 and $3,000,000, so we will remove values < 60,000,000 and convert the Giá column to billion units.

Text

Description automatically generated

Figure 2. 10: Remove values < 60,000,000 and convert to billion

Step 11: Eliminate rows with Diện tích less than 30 and Số tầng from 869 or more

Graphical user interface, text, application, chat or text message

Description automatically generated

Figure 2. 11: Eliminate rows with Diện tích và Số tầng

Step 12: Proceed to export the csv file after completing the data preprocessing

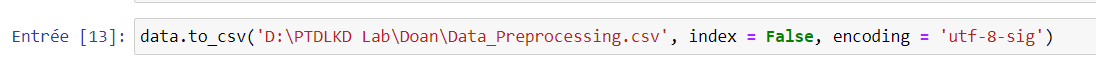


Figure 2. 12: Export csv file

## **3. Algorithm execution**

Step 1: Proceed to import the necessary libraries

Text

Description automatically generated

Figure 3. 1: Proceed to import the necessary libraries

Step 2: Proceed to read the data

Text

Description automatically generated with medium confidence

Figure 3. 2: Proceed to read the data

Step 3: Check data information

Graphical user interface, text

Description automatically generated

Figure 3. 3: Check data information

Step 4: Because the algorithm inputs are numeric values, we will create a mapping dictionary for string columns

Text, letter

Description automatically generated

Figure 3. 4: Create a mapping dictionary

Step 5: Split the columns into 2 categories: Column x contains attributes for decision making, column y contains actual values then split 30% for test data and 70% for training data. Set random\_state = 42 so that every rerun of the algorithm will give the same result

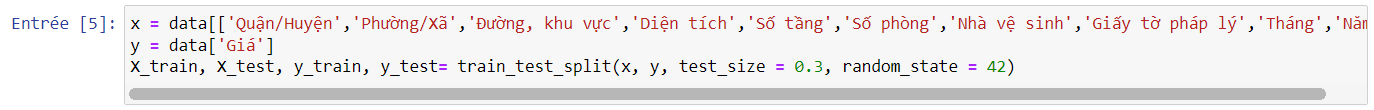


Figure 3. 5: Split column, split test and train

Step 6: Run SVR algorithm with kernel='rbf' to run algorithm with Radial basis function kernel

Graphical user interface, text, application

Description automatically generated

Figure 3. 6: Run the SVR algorithm

Step 7: Make a prediction and save the prediction result in the variable y\_pred

Shape

Description automatically generated

Figure 3. 7: Predict and save results

Step 8: Calculate the RMSE of the SVR algorithm

Graphical user interface, application

Description automatically generated

Figure 3. 8: Calculate the RMSE of the SVR algorithm

Step 9: Calculate R square of SVR algorithmGraphical user interface, text, application

Description automatically generated

Figure 3. 9: Calculate R square of SVR algorithm

Step 10: Run the DTR algorithm and set random\_state = 0 so that every rerun of the algorithm will give the same result

Graphical user interface, text, application

Description automatically generated

Figure 3. 10: Run the DTR algorithm

Step 11: Make predictions



Figure 3. 11: Make predictions

Step 12: Calculating the RMSE of the DTR algorithm

Graphical user interface, application, Word

Description automatically generated

Figure 3. 12: Calculating the RMSE of the DTR algorithm

Step 13: Calculate R square of DTR algorithm

Graphical user interface, text, application

Description automatically generated

Figure 3. 13: Calculate R square of DTR algorithm

Step 14: Run the LCV algorithm

Text

Description automatically generated with low confidence

Figure 3. 14: Run the LCV algorithm

Step 15: Make predictions

A picture containing icon

Description automatically generated

Figure 3. 15: Make predictions

Step 16: Calculating the RMSE of the LCV algorithm

A picture containing graphical user interface

Description automatically generated

Figure 3. 16: Calculating the RMSE of the LCV algorithm

Step 17: Calculate R square of LCV algorithm

Graphical user interface, text

Description automatically generated

Figure 3. 17: Calculate R square of LCV algorithm

Step 18: Run Ridge algorithm with alpha = 1.0

Graphical user interface, text, application

Description automatically generated

Figure 3. 18: Run Ridge algorithm

Step 19: Make predictions



Figure 3. 19: Make predictions

Step 20: Calculating the RMSE of the Ridge algorithm

Graphical user interface, text, application

Description automatically generated

Figure 3. 20: Calculating the RMSE of the Ridge algorithm

Step 21: Calculate the R square of the Ridge algorithm

Graphical user interface, text, application

Description automatically generated

Figure 3. 21: Calculate the R square of the Ridge algorithm

Step 22: Compare RMSE between algorithms

Chart, bar chart

Description automatically generated

Figure 3. 22: Compare RMSE

Step 23: Compare R square between algorithms

Chart, waterfall chart

Description automatically generated

Figure 3. 23: Compare R square

## **4. Comparison**

Chart, bar chart

Description automatically generated

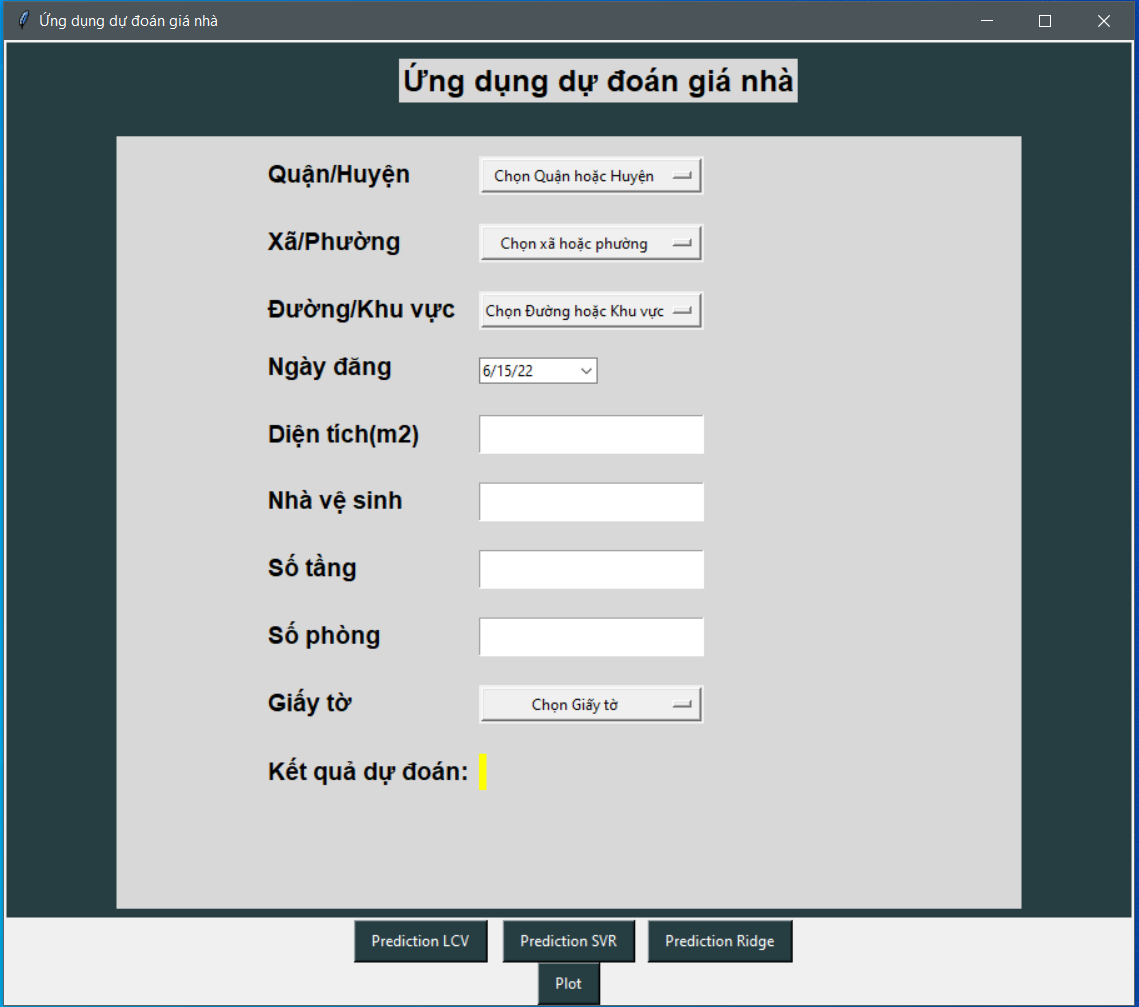
Figure 4. 1: Compare RMSE

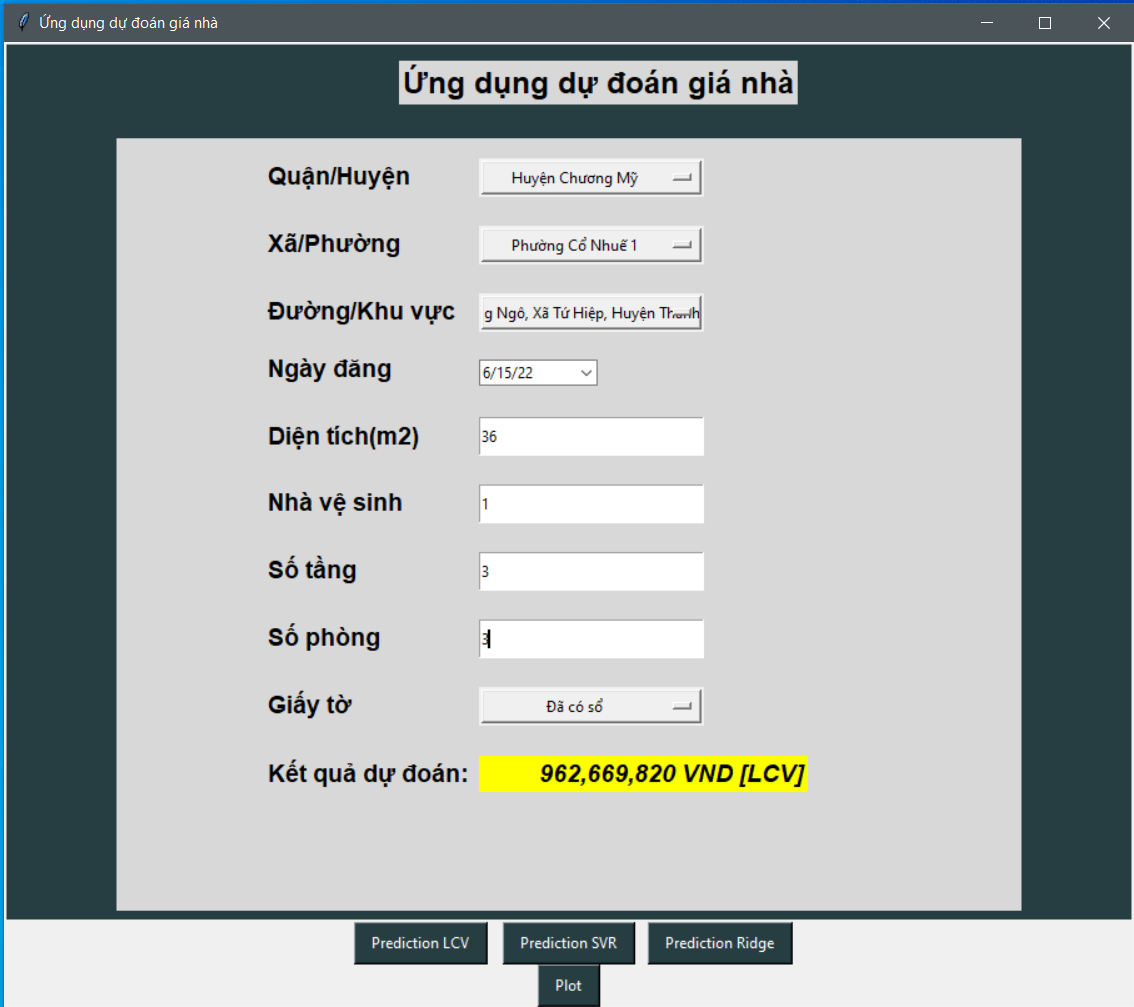
Based on the RMSE stats of each algorithm, we can evaluate as follows:

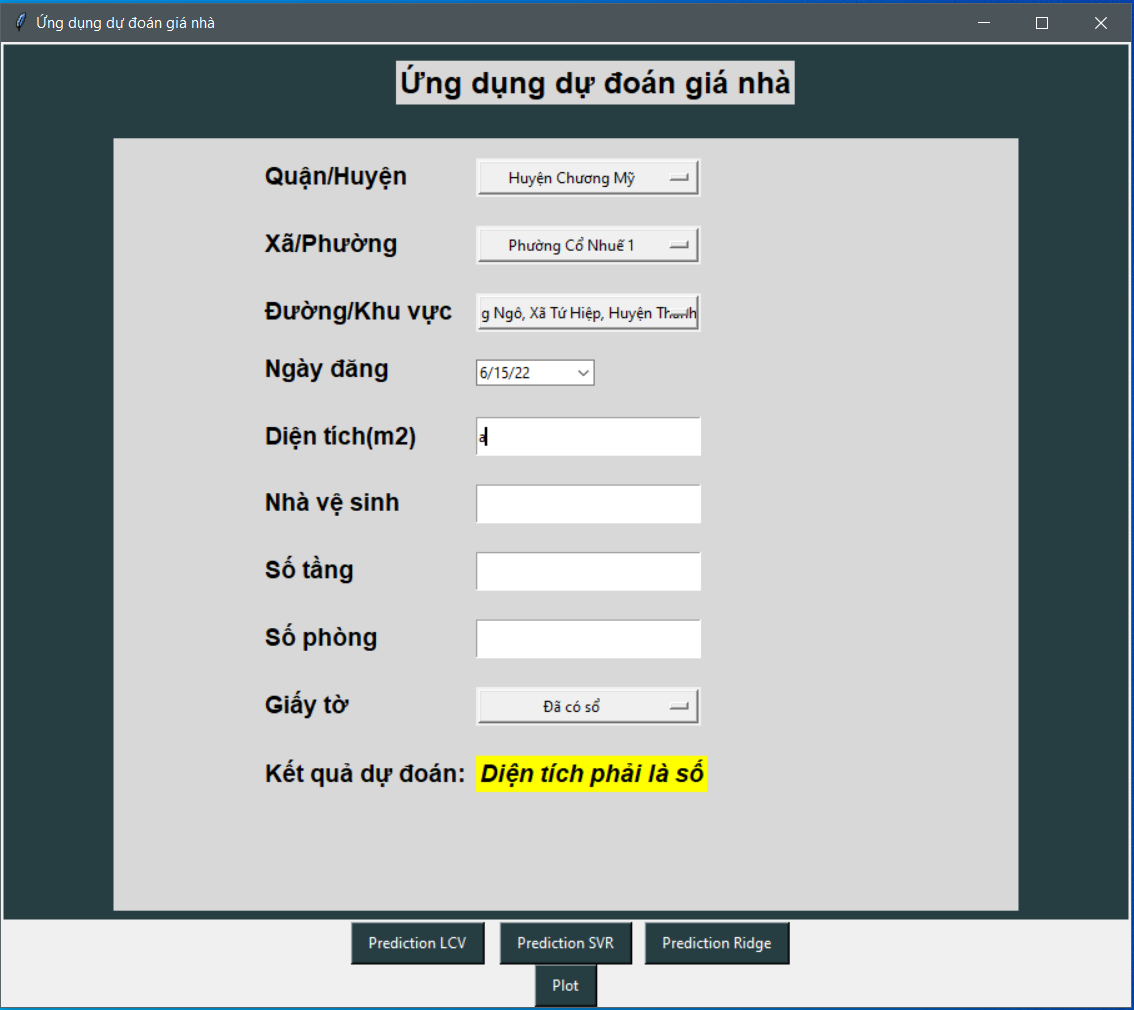
* LCV and Ridge models are the best of the 3 models
* Bad model is: DRT.

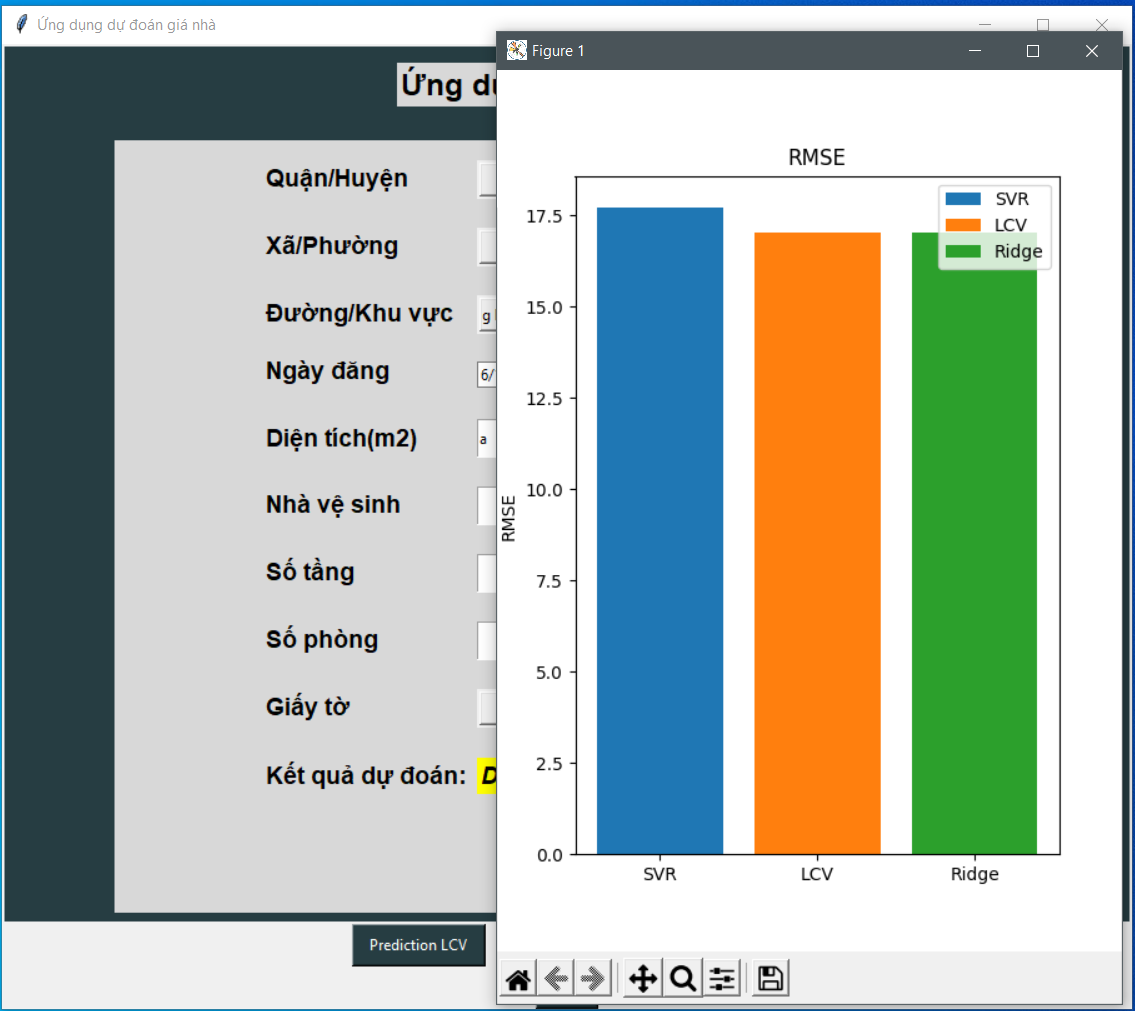
## **5. Application Demo**

Program setting: <https://drive.google.com/drive/u/0/folders/1nhX-ycoCqwXtZG3XAgpfzK3b9rxaH2YY>









# **E. Summary**

## **1. General assessment**

Based on the above analysis and results, we can conclude that the LassoLars and Ridge models are the most suitable among the four models above.

Through this report, we can learn more deeply and better about 4 models: Support Vector Regression, Decision Tree Regression, Ridge regression, LassoLars regression.

By combining the knowledge learned in class and the constant efforts of the whole team, we have created an application to

forecast house prices applies the regression models that the team has studied.

The application to forecast house prices not only gives prediction results on house prices, but also has detailed and specific visualization graphs to support users to the maximum.

## **2. Advantages**

* Learn a lot of new knowledge
* Cultivate the ability to self-study and work in groups
* Learn a new programming language
* Read many good reports
* Brief and complete summary of professional knowledge in the report

## **3. Disadvantages**

* Spend a lot of time researching and learning
* Difficulty in research and discovery process
* Difficulty in arranging project time among members

## **4. Development**

* Find more models that are better at predicting house prices
* Scaling the application
* Try to develop more features, and improve the application both in function and interface in the future

# **F. References**

<http://www.laydulieu.com/nha-dat/?model=0&type=0&province=01&district=&ward=&address=&fprice=&tprice=&fdate=&tdate=&field%5Bdien_tich%5D=&field%5Bhuong%5D=&field%5Btang%5D=&field%5Bphong%5D=&field%5Bnha_ve_sinh%5D=&field%5Bgiay_to_phap_ly%5D=&phone=>

[1] <https://en.wikipedia.org/wiki/Support-vector_machine>

[2] <https://www.educba.com/support-vector-regression/>

[3] <https://data-flair.training/blogs/svm-kernel-functions/>

[4] https://www.educba.com/support-vector-regression/

[5] <https://en.wikipedia.org/wiki/ID3_algorithm>

[6] https://retirepedia.com/5-cities-with-the-cheapest-property-in-vietnam.html