Project Report On

Identification of methodology used in real estate property valuation

INTRODUCTION

1.1 Overview

Real estate property valuation is a critical process that involves determining the market value of a property. The value of a property is determined by various factors such as location, size, amenities, and condition. Real estate valuation is crucial for various purposes such as property sales, mortgages, insurance, taxation, and financial reporting. The process of property valuation involves the use of various methodologies, and the choice of methodology depends on the type of property, the purpose of valuation, and the availability of data. This project report will focus on identifying the methodologies used in real estate property valuation and will provide an overview of each methodology. The report will also highlight the advantages and disadvantages of each methodology and discuss the factors that affect the choice of methodology.

1.2 Purpose

The purpose of this project report is to provide an overview of the methodologies used in real estate property valuation. The report aims to identify and explain the different valuation methods, their advantages and disadvantages, and the factors that affect the choice of methodology. The report will help readers understand the valuation process and the importance of selecting an appropriate methodology for arriving at an accurate and reliable valuation. The report will also provide insights into the practical application of each methodology in the real estate industry. Overall, the report aims to enhance the reader's knowledge and understanding of real estate property valuation and its methodologies.

2.LITERATURE SURVEY

2.1 Existing Problem

The existing problem with real estate property valuation projects is that they can be subjective and depend on the methodology used, the quality of the data analyzed, and the expertise of the appraiser. This can result in inconsistencies in property valuations and potential disputes between buyers, sellers, lenders, and other parties involved in a real estate transaction.

To address this problem, a proposed solution would be to adopt standardized appraisal practices and guidelines that ensure consistent and objective property valuations. For example, the Uniform Standards of Professional Appraisal Practice (USPAP) is a set of guidelines that establish ethical and competency standards for appraisers, including requirements for data analysis and reporting.

Another proposed solution would be to incorporate technology and data analysis tools that can help appraisers gather and analyze property and market data more efficiently and accurately. For example, automated valuation models (AVMs) can use algorithms and data analysis to estimate a property's value based on recent sales data and property characteristics.

By adopting standardized practices and incorporating technology, real estate property valuations can be made more consistent, objective, and accurate, which can reduce disputes and promote greater confidence in the valuation process.

2.2 Proposed Solution

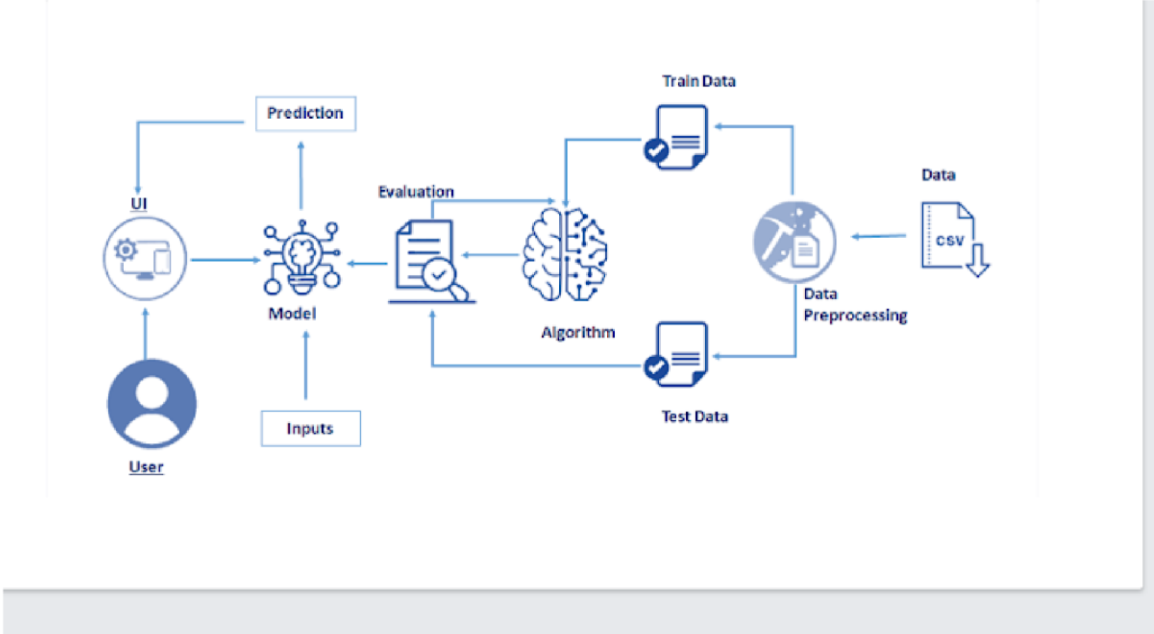
The methodology used in a real estate property valuation project report typically follows established valuation principles, but the specific solution proposed will depend on the nature and purpose of the project.

For example, a proposed solution for a property appraisal for lending purposes may involve applying the cost approach, income approach, and sales comparison approach to determine a market value for the property, while also considering any relevant market data and physical property characteristics.

On the other hand, a proposed solution for a property appraisal for property tax purposes may involve using the cost approach to determine the property's current value, while also considering any relevant local and state tax laws and regulations.

In both cases, the proposed solution would involve a thorough analysis of the property and market data, as well as careful consideration of the specific valuation purpose and context.

3. THEORETICAL ANALYSIS



3.2 Hardware/Software Designing

1.Software Requirements

Pandas: It is a python library mainly used for data manipulation.

NumPy: This python library is used for numerical analysis.

Counter: Python Counter is a container that will hold the count of each of the elements present in the container.

Matplotlib and Seaborn: Both are the data visualization library used for plotting graphs which will help us for understanding the data.

Accuracy score: used in classification type problem and for finding accuracy it is used.

R2 Score: Coefficient of Determination or R^2 is another metric used for evaluating the performance of a regression model. The metric helps us to compare our current model with a constant baseline and tells us how much our model is better.

Literal_eval: we can use ast.literal_eval() to evaluate the string as a python expression.

Pickle: to serialize your machine learning algorithms and save the serialized format to a file.
Word cloud: to create visualizations with text data

4.EXPERIMENTAL INVESTIGATION

Experimental investigation is not typically used in real estate property valuation projects, as the value of a property is typically determined by established valuation methods, such as the cost approach, income approach, and sales comparison approach. These methods rely on analyzing market data and property characteristics rather than conducting experiments.

However, experimental investigation could potentially be used in certain cases to test the impact of specific property features or characteristics on property value. For example, an experimental investigation could be conducted to test the impact of home staging on the sale price of a property. In this case, the experimental design could involve staging some properties and leaving others unstaged, and then comparing the sale prices to determine whether staging had a significant impact on property value.

However, it is important to note that such experimental investigations would need to be carefully designed and controlled to ensure that the data collected are reliable and valid, and that any confounding variables are controlled for. Furthermore, the results of such investigations may not necessarily be applicable to all properties or markets, and would need to be interpreted with caution.

5.FLOWCHART

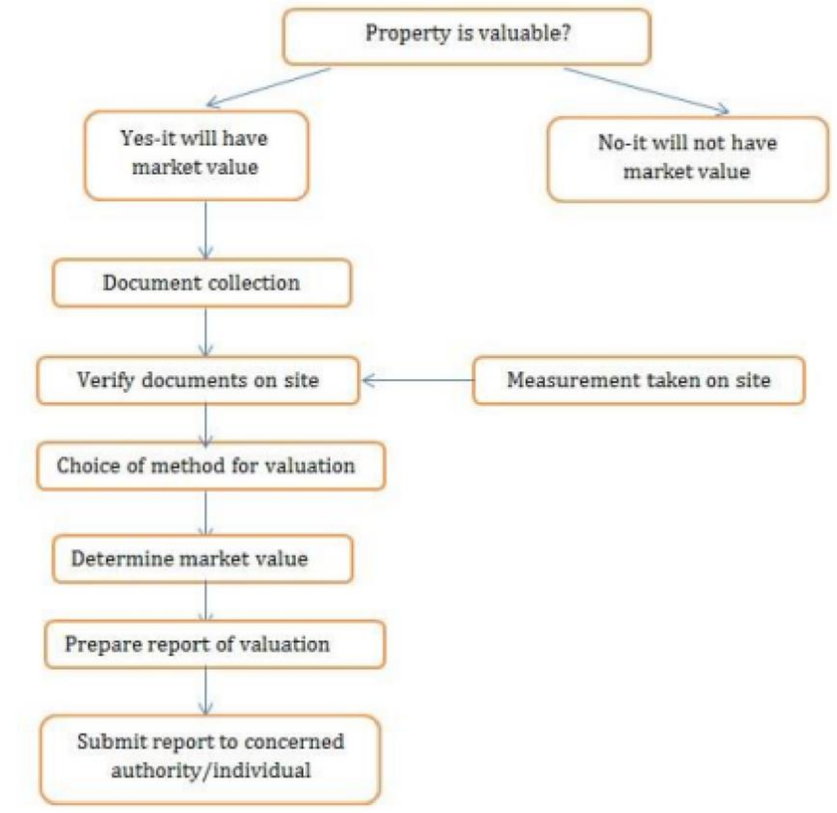

Property is valuable?
Yes-it will have market value
No-it will not have market value
Document collection
Verify documents on site
Measurement taken on site
Choice of method for valuation
Determine market value
Prepare report of valuation
Submit report to concerned authority/individual

Fig-1: Methodology used in valuation

6.RESULT

Fig:1 output of the dataset:

Fig 2: Web Application view :

Fig 3: Predicting the GDP:

7.ADVANTAGES AND DISADVANTAGES

The methodology used in real estate property valuation projects, such as the cost approach, income approach, and sales comparison approach, has several advantages and disadvantages:

Advantages:

1. Established methods: These valuation methods have been used for many years and are well-established in the industry.
2. Objective: These methods rely on data and objective analysis, which can help ensure that valuations are fair and accurate.
3. Comprehensive: The use of multiple valuation methods, such as the cost, income, and sales comparison approaches, can provide a comprehensive understanding of the property's value.

Disadvantages:

1. Time-consuming: Conducting a thorough property inspection, gathering and analyzing data, and applying multiple valuation methods can be time-consuming and require specialized expertise.
2. Subjectivity: The interpretation of data and the selection of comparable properties for the sales comparison approach can introduce some subjectivity into the valuation process.
3. Uncertainty: Real estate property valuations are estimates, and there is always some uncertainty in the final value due to changes in the market, property condition, and other factors.

Overall, the methodology used in real estate property valuation projects has several advantages and disadvantages, and the specific approach taken will depend on the property being valued and the purpose of the valuation. It is important to consider these factors when interpreting valuation results and making decisions based on property value

8.APPLICATIONS

The methodology used in real estate property valuation projects has several applications, including:

1. Real estate transactions: Real estate property valuations are commonly used in real estate transactions to help determine a fair market price for a property.
2. Property taxes: Property valuations are often used by local governments to determine property taxes.
3. Estate planning: Property valuations can be useful for estate planning purposes, such as determining the value of a property for inheritance or tax purposes.
4. Insurance: Property valuations can help insurance companies determine appropriate coverage and premiums for properties.
5. Investment decisions: Property valuations can be useful for making investment decisions, such as determining the potential return on investment for a particular property.
6. Legal disputes: Property valuations may be used in legal disputes, such as property tax appeals, eminent domain cases, or divorce settlements.

Overall, the methodology used in real estate property valuation projects has a wide range of applications and is an important tool for making informed decisions related to real estate. It is important to use a reliable and accurate valuation method and to consider the specific purpose of the valuation when interpreting results.

9.CONCLUSION

In conclusion, the methodology used in real estate property valuation projects is a critical tool for determining the fair market value of a property. The cost approach, income approach, and sales comparison approach are established valuation methods that rely on objective data analysis and provide a comprehensive understanding of the property's value. While these methods have some limitations, such as subjectivity and uncertainty, they are widely used in the industry and have a range of important applications.

10. FUTURE SCOPE

1. In terms of future scope, advances in technology, data analysis, and machine learning may provide opportunities to further refine and improve real estate property valuation methods. For example, using big data analytics to track real estate market trends, property characteristics, and comparable sales data could provide more accurate and comprehensive valuations. Additionally, machine learning algorithms could potentially be used to automatically analyze data and identify patterns that may impact property value. It is likely that advances in technology and data analysis will continue to shape the future of real estate property valuation, providing new opportunities to improve accuracy and efficiency.

11.BIBLIOGRAPHY

1. https://www.kaggle.com

2. https://github.com

APPENDIX:

Source code:

```
import numpy as np

from flask import Flask, request, jsonify, render_template

import pickle

import sklearn.externals as extjoblib

import joblib

app = Flask(__name__)

model = joblib.load('Random.pkl')

model1 = joblib.load('Linear.pkl')
```

```
@app.route('/')

def home():

    return render_template('index.html')


@app.route('/predict',methods=['POST'])

def predict():

    '''

    For rendering results on HTML GUI

    '''


    int_features = [float(x) for x in request.form.values()]

    final_features = [np.array(int_features)]


    prediction = model.predict(final_features)

    prediction1 = model1.predict(final_features)
    output = round(prediction[0], 2)

    output1 = round(prediction1[0], 2)

    return render_template('index.html', prediction_text='Random Forest: GDP of the country is  $
 {}'.format(output), prediction_text2='Linear Regression: GDP of the country is  $
{}'.format(output1)
)

@app.route('/country',methods=['POST'])
 def
country():
    '''
    For rendering results on HTML GUI
    '''

    text=request.form.get("country")

    return render_template('index.html', prediction_text3=text)
```

```
@app.route('/predict_api',methods=['POST'])

def predict_api():

    '''

    For direct API calls trought request

    '''

    data = request.get_json(force=True)

    prediction = model.predict([np.array(list(data.values()))])



    output = prediction[0]

    return jsonify(output)



if __name__ == "__main__":

    app.run(debug=True)

# -*- coding: utf-8 -*-

"""GDP.ipynb
```

```
Automatically generated by Colaboratory.

Original file is located at

    https://colab.research.google.com/drive/1VKGQ2gzGRmH0VyIv-N-QSs7cDFGfAdsA

"""

import numpy as np
```

```
import pandas as pd

import seaborn as sns

from matplotlib import pyplot as plt

from sklearn.preprocessing import LabelEncoder

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

from sklearn.ensemble import RandomForestRegressor

import pickle

world=pd.read_csv("countries of the world.csv",decimal=',')

world.info()

world.head()

world.value_counts('Region')

#world.value_counts('Region_label')

world.describe()
world.isnull().sum()
```

```
for col in world.columns.values:
    if world[col].isnull().sum() == 0:
        continue
    if col == 'Climate':
        guess_values = world.groupby('Region')['Climate'].apply(lambda x: x.mode().max())
    else:
        guess_values = world.groupby('Region')[col].median()
    for region in world['Region'].unique():
        world[col].loc[(world[col].isnull())&(world['Region']==region)] = guess_values[region]


LE = LabelEncoder()
world['Region_label'] = LE.fit_transform(world['Region'])
world['Climate_label'] = LE.fit_transform(world['Climate'])
world.head()


world['Region_label'].unique()


world['Region_label']


train, test = train_test_split(world, test_size=0.3, shuffle=True)
training_features = ['Population', 'Area (sq. mi.)',
       'Pop. Density (per sq. mi.)', 'Coastline (coast/area ratio)',
       'Net migration', 'Infant mortality (per 1000 births)',
       'Literacy (%)', 'Phones (per 1000)',
       'Arable (%)', 'Crops (%)', 'Other (%)', 'Birthrate',
       'Deathrate','Region_label' ,

       'Climate_label']

   target = 'GDP ($ per capita)'

   train_X = train[training_features]

   train_Y = train[target]

   test_X = test[training_features]

   test_Y = test[target]



   train, test = train_test_split(world, test_size=0.3, shuffle=True)

   training_features = ['Population', 'Area (sq. mi.)',

       'Pop. Density (per sq. mi.)', 'Coastline (coast/area ratio)',

       'Net migration', 'Infant mortality (per 1000 births)',

       'Literacy (%)', 'Phones (per 1000)',

       'Arable (%)', 'Crops (%)', 'Other (%)', 'Birthrate',

       'Deathrate', 'Region_label',
          'Climate_label']

    target = 'GDP ($ per capita)'

    train_X = train[training_features]

    train_Y = train[target]

    test_X = test[training_features]

    test_Y = test[target]


   print(train_X.shape)

   print(train_Y.shape)
```

```
 print(test_X.shape)

print(test_Y.shape)


 model1 = LinearRegression()
```

```
model1.fit(train_X, train_Y)

train_pred_Y = model1.predict(train_X)

test_pred_Y = model1.predict(test_X)


print('Training Score : ',model1.score(train_X,train_Y))

#print(f'Test score : ',r2_score(test_pred_Y,test_Y))


model = RandomForestRegressor(n_estimators = 100,

                              max_depth = 6,

                              min_weight_fraction_leaf = 0.05,

                              max_features = 0.8,

                              random_state = 42)

model.fit(train_X, train_Y)

train_pred_Y = model.predict(train_X)
```

```
test_pred_Y = model.predict(test_X)

from sklearn.metrics import r2_score

print('Training Score :
',model1.score(train_X,train_Y))

print(f'Test score :
',r2_score(test_pred_Y,test_Y))
```

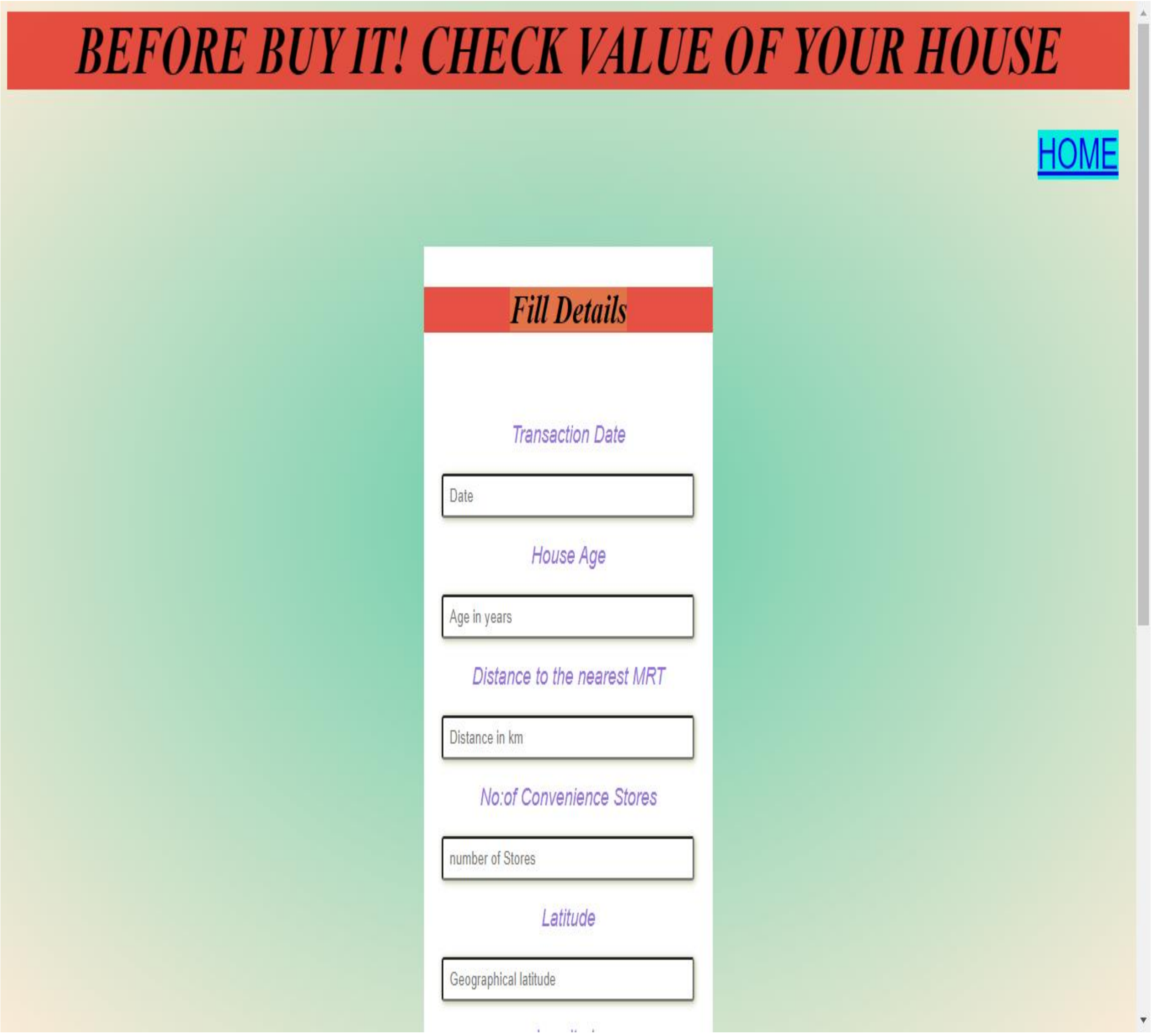
```
df = pd.DataFrame(columns = training_features)

df1=[[31056997.0,647500.0,48.0,0,23.06,163.07,36.0,3.2,12.13,0.22,87.65,46.6,20.34,0,0]]

model.predict(df1)

df=[[3581655,28748, 124.6,  1.26    ,4.93   ,21.52,86.5 ,71.2   ,21.09  ,4.42   ,74.49

,15.11,5.22,3,4]]
```

```
model.predict(df)

 dfk=[[500000000.0,  3287263.0   ,152.0, 2.00,   0.00,   5.00,   99.00,  1000.00,    60.0,   10.0,
 30.00,  10.00,  5.00,   0.0,    0.0,]]

model.predict(df)

 from joblib import Parallel, delayed

 import
joblib

 # Save the model as a pickle in a file

 joblib.dump(model1,
'Linear.pkl')

 # Load the model from the file

 knn_from_joblib                                   =
joblib.load('Linear.pkl')
```

```
# Use the loaded model to make predictions
```

```
knn_from_joblib.predict(dfk)
```

OUTPUT



BEFORE BUY IT! CHECK VALUE OF YOUR HOUSE

HOME

Fill Details

Transaction Date

Date

House Age

Age in years

Distance to the nearest MRT

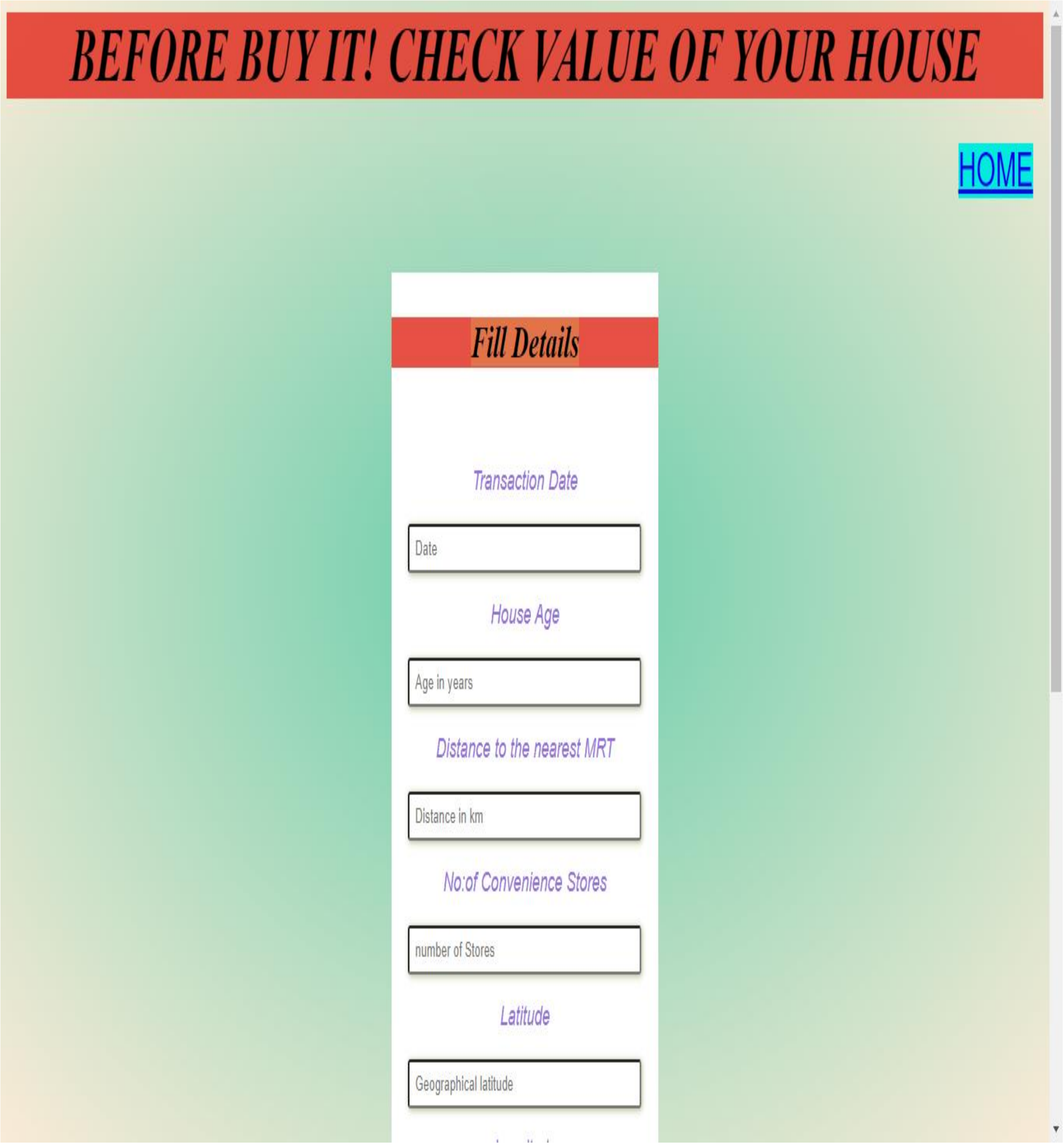
Distance in km

No:of Convenience Stores

number of Stores

Latitude

Geographical latitude



BEFORE BUY IT! CHECK VALUE OF YOUR HOUSE

HOME

Fill Details

Transaction Date

Date

House Age

Age in years

Distance to the nearest MRT

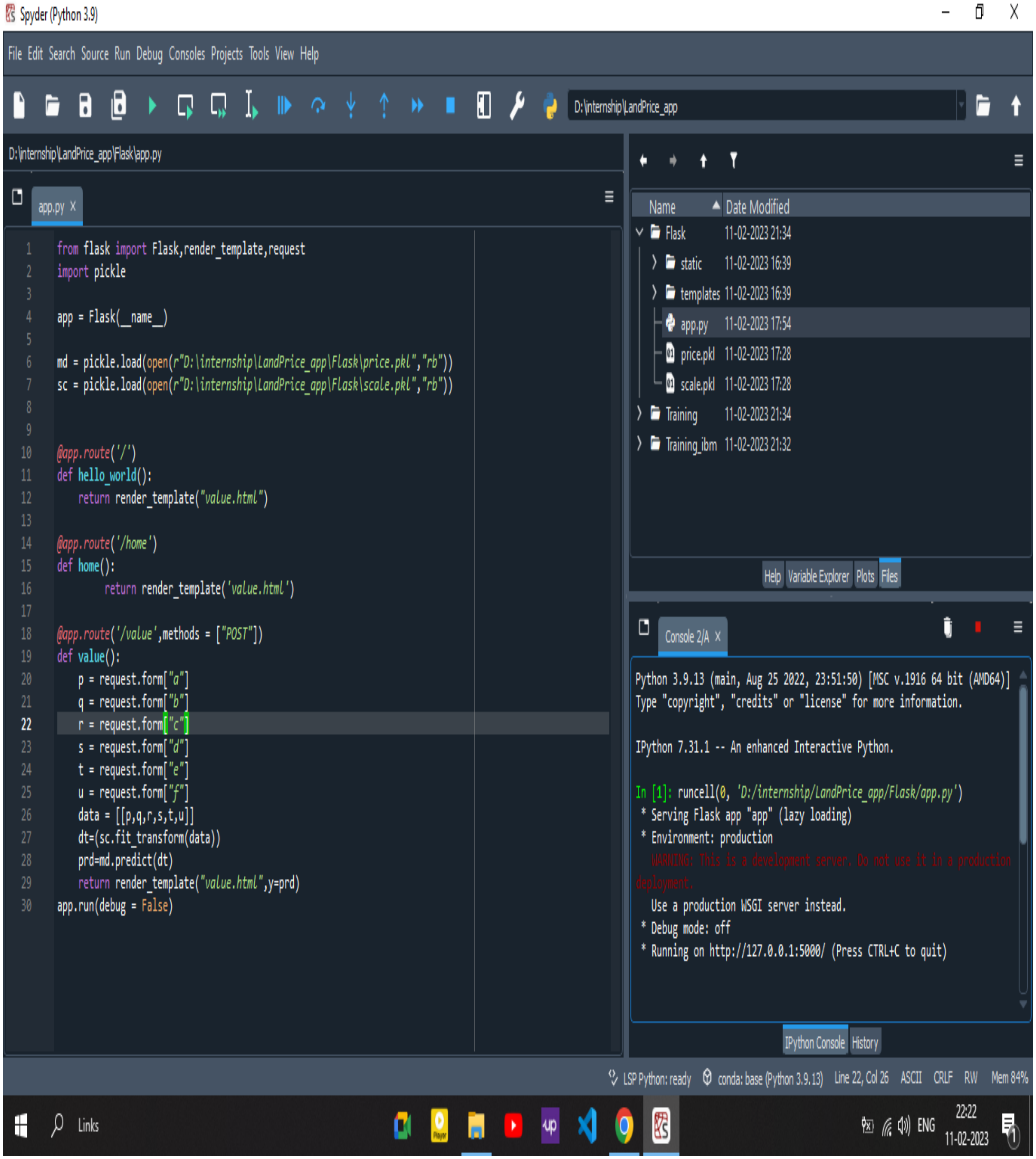
Distance in km

No:of Convenience Stores

number of Stores

Latitude

Geographical latitude



```python
from flask import Flask,render_template,request
import pickle

app = Flask(__name__)

md = pickle.load(open(r"D:\internship\LandPrice_app\Flask\price.pkl","rb"))
sc = pickle.load(open(r"D:\internship\LandPrice_app\Flask\scale.pkl","rb"))


@app.route('/')
def hello_world():
    return render_template("value.html")

@app.route('/home')
def home():
        return render_template('value.html')

@app.route('/value',methods = ["POST"])
def value():
    p = request.form["a"]
    q = request.form["b"]
    r = request.form["c"]
    s = request.form["d"]
    t = request.form["e"]
    u = request.form["f"]
    data = [[p,q,r,s,t,u]]
    dt=(sc.fit_transform(data))
    prd=md.predict(dt)
    return render_template("value.html",y=prd)
app.run(debug = False)
```

Name	Date Modified
Flask	11-02-2023 21:34
static	11-02-2023 16:39
templates	11-02-2023 16:39
app.py	11-02-2023 17:54
price.pkl	11-02-2023 17:28
scale.pkl	11-02-2023 17:28
Training	11-02-2023 21:34
Training_ibm	11-02-2023 21:32

```
Python 3.9.13 (main, Aug 25 2022, 23:51:50) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.31.1 -- An enhanced Interactive Python.

In [1]: runcell(0, 'D:/internship/LandPrice_app/Flask/app.py')
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production
deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```