House Rent Price Prediction Using IBM Watson Studio Machine Learning

1. INTRODUCTION

1.1. Overview

Rental house analysis has turned out to be critical figurecurrent society subsequently the need a rental house administration system.This section will give a short comprehension about foundation of study, meaning of the venture issue explanation, its destinations, scopes, extend support, dangers, extend deliverables and venture spending plan and calendar

1.2. Purpose

In this project, we present a house rent prediction technique that utilizes hi storical data to trainsimple machine learning models which are more accur ate and can help us predict the rent ofthe house. The evaluation results sho w that the accuracy of the models is good enough to be used alongside the current state-of-the-art techniques.

2. LITERATURE SURVEY

2.1. Existing problem

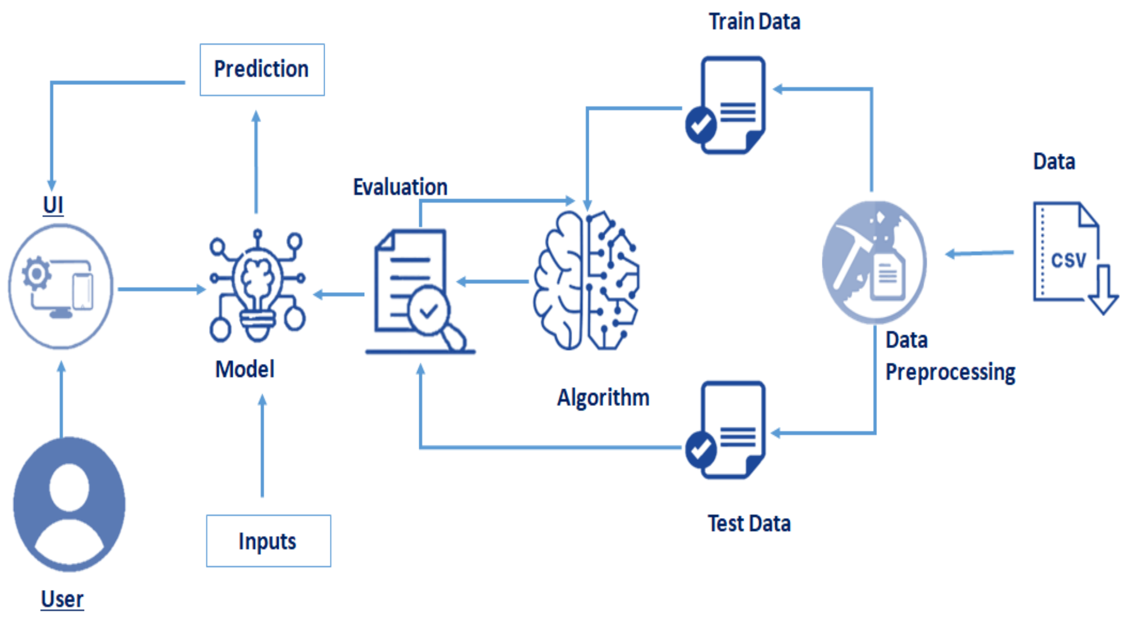
Firstly, with using online algorithm, we design an optimal (deterministic) online strategy. This strategy is to rent the house until the difference aggregated between rent fee and depreciation and opportunity loss sum up to the transaction cost caused by buying and selling house, and then to buy house. Subsequently, we design an online risk strategy in the light of forecasting, and obtain the set of online risk strategy that the manager can accept risk tolerance level by comparing the unsuccessful online risk strategy with optimal (deterministic) online strategy. The risk strategy will lead to reward if forecasting is successful, the risk is sustainable if unsuccessful. The decision-makers can design online strategy to improve outcome in the light of his risk tolerance level.

2.2. Proposed solution

Firstly, with using online algorithm, we design an optimal (deterministic) online strategy. This strategy is to rent the house until the difference aggregated between rent fee and depreciation and opportunity loss sum up to the transaction cost caused by buying and selling house, and then to buy house. Subsequently, we design an online risk strategy in the light of forecasting, and obtain the set of online risk strategy that the manager can accept risk tolerance level by comparing the unsuccessful online risk strategy with optimal (deterministic) online strategy. The risk strategy will lead to reward if forecasting is successful, the risk is sustainable if unsuccessful. The decision-makers can design online strategy to improve outcome in the light of his risk tolerance level.

3. THEORITICAL ANALYSIS

3.1. Block diagram



3.2. Hardware/software design

Software

- Anaconda Navigator : Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning-related applications. It can be installed on Windows, Linux, and macOS.Conda is an open-source, cross-platform, package management system. Anaconda comes with so very nice tools like JupyterLab, Jupyter Notebook, QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code. For this project, we will be using a Jupyter notebook and Spyder.

Python packages:

- NumPy : NumPy is a Python package that stands for 'Numerical Python. It is the core library for scientific computing, which contains a powerful n-dimensional array of objects.
- Pandas : pandas is a fast, powerful, flexible, and easy-to-use open-source data analysis and manipulation tool, built on top of the Python programming language.
- Matplotlib: It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits
- Scikit-learn is an open source data analysis library, and the gold standard for Machine Learning (ML) in the Python ecosystem. Key concepts and features include: Algorithmic decision-making methods, including: Classification: identifying and categorizing data based on patterns.
- Flask: Web framework used for building Web applications

Hardware

Device name : LAPTOP-D94535RL

Processor : Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11 GHz

System type : 64-bit operating system, x64-based processor

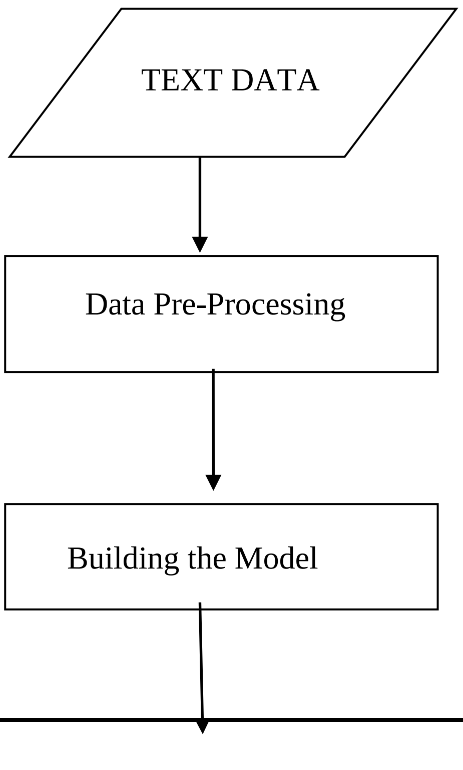
4. EXPERIMENTAL INVESTIGATIONS

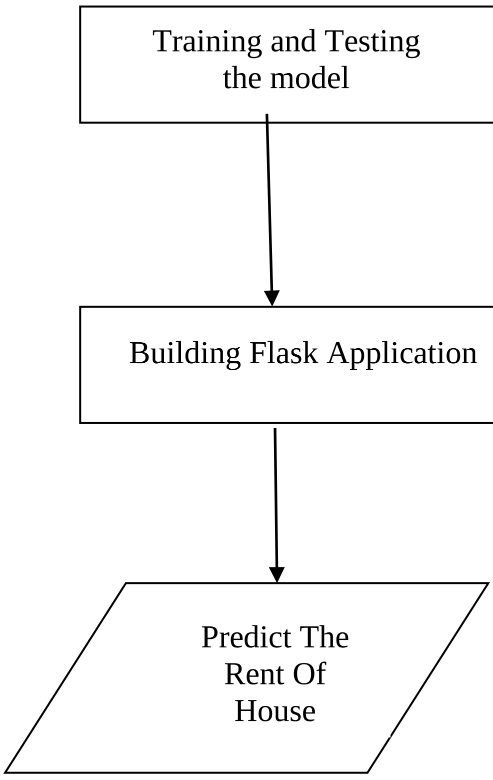
The user’s necessities record was examined for better comprehension of what was required of the framework. Methods for actualizing these necessities were broke down. Physical modules of the framework were planned and recognizing of the working condition in which they were to chip away at. The frameworkwas a visual essential framework/application. The database was refreshed each time the director; include,erases or erases information the framework.

It's just the overseer who has entryto the framework to view or roll out improvements when vital. The framework was intended to permit the head to see, alter, erase and add information to the databaseEach time a client comes, he/she is enrolled in the occupant enlistment table of the database withother pertinent insightsabout the inhabitant.

Framework configuration included changing the product prerequisites into an engineering that portrayed its top-level structure and distinguished the product parts and built up a nitty grittyplan for every product segments. For every prerequisite, an arrangement of at least one outline components was created.

5. FLOWCHART




Training and Testing the model
Building Flask Application
Predict The Rent Of House

6. RESULT

The output of this project is if we are giving the inputs According to the House then it will predict The Rent of House.

7. ADVANTAGES AND DISADVANTAGES

Advantages

● This model predicts the house rent based on the location.

● The house rent can be found remotely and we can plan our Budget.

Disadvantages

● Although the model predicts the house rent ,the probability of matching with exact prices is low.

8. APPLICATIONS

● Use to predict house rent.

● Useful in real state fields.

9. CONCLUSION

All in all, the product can be utilized as a stock framework to give a casing work that empowers the troughs to make sensible exchanges set aside a few minutes outline. Every exchange made on the framework run as an inseparable unit with the information being refreshed in the database for our situation it is Microsoft Access 2007 which is the back end. To wrap things up it is not the work that played the approaches to achievement howeverALMIGHTY GOD

10. FUTURE SCOPE

In future our project is meant to satisfy the needs of rental house owners. Several user friendly interfaces have also been adopted. This package shall prove to be a powerful in satisfying all the requirements of the users. It is with utmost faith that I present this software to you hoping that it will solve your problems and encourage you to continue appreciating technology because it is meant to change and ease all our work that seems to be very difficult. I don't mean that my project is the best or that I have used the best technology available it just a simple and a humble venture that is easy to understand. In extent we can add GPS system in build and can give live chat online option to users. This project can also be extended to IOS Platform and several state Database can be included. Could also allow local business to push deals/coupons within a certain geographic area

11. BIBILOGRAPHY

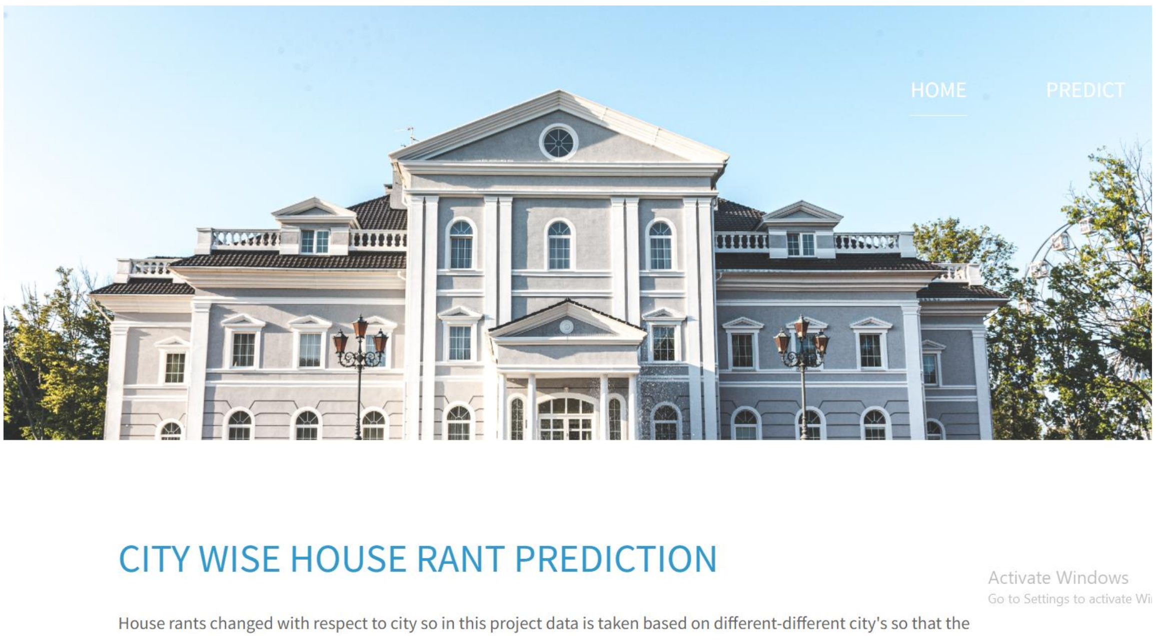
- https://www.mdpi.com/2078-2489/12/8/302/htm
- http://cs229.stanford.edu/proj2017/final-reports/5244159.pdf

12. APPENDIX

```
import numpy as np
import pickle
from flask import Flask,request, render_template

app=Flask(__name__,template_folder="templates")
model = pickle.load(open('rf_rand_model.pkl', 'rb'))

@app.route('/', methods=['GET'])
def index():
    return render_template('home.html')
@app.route('/home', methods=['GET'])
def about():
    return render_template('home.html')
@app.route('/pred',methods=['GET'])
def page():
    return render_template('upload.html')

@app.route('/predict', methods=['GET', 'POST'])
def predict():
    input_features = [float(x) for x in request.form.values()]
    features_value = [np.array(input_features)]
    print(features_value)

    #features_name                                                   =
['city','BHKS','sqft_per_inch','build_up_area','Type_of_property','deposit']
    prediction = model.predict(features_value)
    output=prediction[0]    #np.exp(predictions)
    output = np.exp(output)
    output = np.round(output)
    print(output)
    return render_template('upload.html', prediction_text= 'House Rent is {} '.format((output)))


if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8000, debug=False)
```

<u>SCREENSHOTS</u>

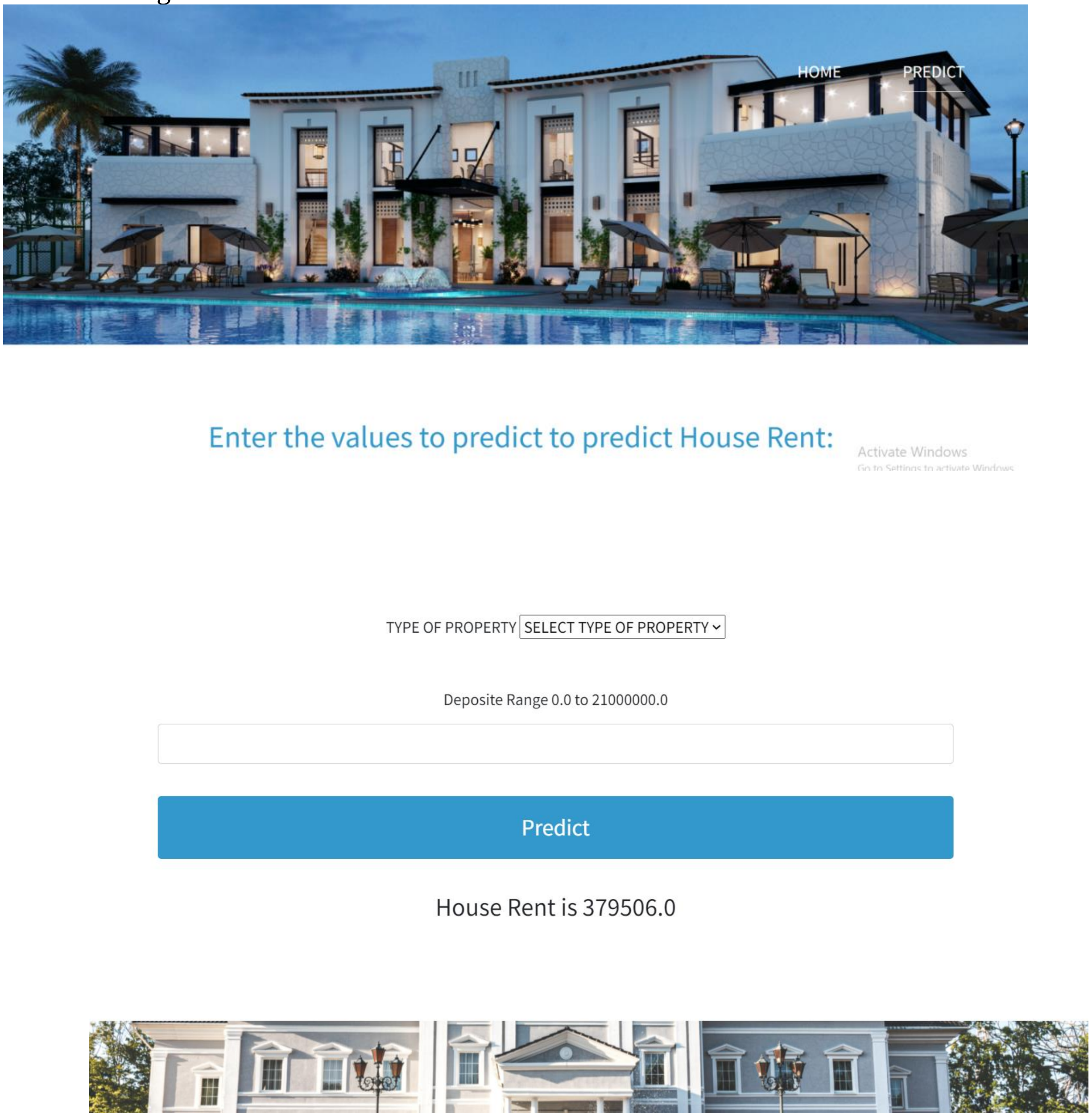
Home page



CITY WISE HOUSE RANT PREDICTION

House rants changed with respect to city so in this project data is taken based on different-different city's so that the

Prediction Page



Enter the values to predict to predict House Rent:

TYPE OF PROPERTY SELECT TYPE OF PROPERTY

Deposite Range 0.0 to 21000000.0

Predict

House Rent is 379506.0