“HOUSE RENT PREDICTION IN HYDERABAD”

A Course End Project Submitted in Partial Fulfillment of the Requirements for the Course of

A8704 – MACHINE LEARNING LABORATORY

In

Department of Artificial Intelligence and Machine Learning

Submitted By

Name of the Student	Roll Number
P.ABHINAY	23885A7306
N.SUSHANTH KUMAR	22881A7337
P.AKSHAYA REDDY	22881A7342
R.SRITHA NAYAK	22881A7347
SHIREEN SULTHANA	22881A7352
UDAY	22881A7357
V.HARSHITH	22881A7362

CASE STUDY REPORT

VARDHAMAN COLLEGE OF ENGINEERING

(AUTONOMOUS)

Affiliated to JNTUH, Approved by AICTE, Accredited by NAAC with A++ Grade, ISO 9001:2015 Certified

Kacharam, Shamshabad, Hyderabad – 501218, Telangana, India

2024

VARDHAMAN COLLEGE OF ENGINEERING

(AUTONOMOUS)

Affiliated to JNTUH, Approved by AICTE, Accredited by NAAC with A++ Grade, ISO 9001:2015 Certified
Kacharam, Shamshabad, Hyderabad – 501218, Telangana, India

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

CERTIFICATE

Certified that this is a bonafide record of the course end project work entitled, **"HOUSE RENT PREDICTION IN HYDERABAD"**, done by ,**P.ABHINAY (23885A7306), N.SUSHANTH KUMAR(22881A7337), P.AKSHAYA REDDY(22881A7342), R.SRITHA NAYAK(22881A7347), SHIREEN SULTHANA(22881A7352), UDAY (22881A7357), V.HARSHITH (22881A7362) submitted to the faculty of Artificial Intelligence and Machine Learning, in partial fulfillment of the requirements for the course of Machine Learning during the year 2024 (IV Semester).**

Semester End Examination held on ……………………………………………

Course Instructor **Course Lead**

ABSTRACT

This project aims to develop a predictive model for house rent prices in Hyderabad using machine learning techniques and deploy it as a web application. The model is built using a dataset containing various attributes of rental properties, such as locality, number of bathrooms, floor level, parking facilities, property size, type of BHK (bedroom, hall, kitchen), and maintenance costs. The project involves several key steps: data cleaning and preprocessing, feature engineering, model training, and deployment.

Data preprocessing includes removing outliers, handling missing values, and encoding categorical variables. The Regression model is trained to predict the rental prices based on the cleaned dataset. The model is evaluated and fine-tuned to ensure accurate predictions.The predictive model is integrated into a web application built with Flask, a Python-based web framework. The web interface lets users input property details and receive predicted rental prices. The front end is developed using HTML, CSS, and JavaScript, ensuring a user-friendly experience.

This project demonstrates the practical application of machine learning in real estate, providing a valuable tool for potential renters and real estate professionals to estimate rental prices accurately. It highlights the importance of data preprocessing, feature selection, and model evaluation in building effective predictive models. Integrating the model into a web application illustrates how machine learning solutions can be deployed to solve real-world problems, making advanced analytics accessible to a broader audience

List of Contents

Contents **Page No.**

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Introduction

The city of Hyderabad, located in the Indian state of Telangana, is a thriving metropolis with a growing population and economy. As a result, the demand for housing and rentals has increased significantly, leading to a rise in house rent prices. Predicting house rent in Hyderabad is crucial for both tenants and landlords, as it helps them make informed decisions about their living arrangements and investments.

To predict house rent in Hyderabad, it is essential to consider various factors that influence rent prices. These factors include the number of bedrooms (BHK), size of the house, area type, location, furnishing status, and tenant preferences. By analyzing these factors, we can develop a model that accurately predicts house rent in different parts of the city.

One of the primary factors affecting house rent in Hyderabad is the location. The city is divided into various localities, each with its unique characteristics and amenities. For instance, localities like Banjara Hills, Jubilee Hills, and Gachibowli are considered prime areas with high demand and higher rent prices. On the other hand, localities like Kukatpally and Miyapur are considered more affordable with lower rent prices.

Another significant factor influencing house rent in Hyderabad is the size and type of house. Larger houses with more bedrooms and amenities tend to command higher rent prices. Additionally, houses with modern amenities like air conditioning, gym, and swimming pool are in high demand and often come with higher rent prices.

Furnishing status is also an important factor in determining house rent in Hyderabad. Houses that are fully furnished with modern appliances and furniture tend to attract higher rent prices. Furthermore, houses that are partially furnished or require additional furniture and appliances may command lower rent prices.

1.2 Aim and Objective

Aim: Develop a predictive model and an interactive web application that accurately forecasts house rent using machine learning algorithms and real-time data inputs.

Objectives:

- **Data Collection and Preprocessing**: Acquire a diverse dataset related to house rent, including variables such as location , type of BHK, no of bedrooms ,floor. Preprocess the data by cleaning, handling missing values, and engineering relevant features to improve model accuracy.
- **Model Development**: Train various machine learning models, including Decision Trees, Random Forests, and XG Boost, to predict delivery times. Fine-tune model parameters and select the most accurate model based on evaluation metrics.
- **Evaluation**: Assess the performance of the trained models using metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2) score. Compare the results to determine the best-performing model for deployment.

- **Frontend Integration**: Design and implement a web-based user interface using HTML, CSS, and JavaScript. The interface will allow users to input parameters such as current weather, traffic conditions, and delivery location to obtain real-time predictions of delivery times.
- **Deployment**: Integrate the selected predictive model with the frontend interface to create a seamless and interactive user experience. Deploy the application for real-world use, enabling users to access house rent .

1.3 Problem Statement

In Hyderabad, a rapidly growing metropolitan city with a burgeoning real estate market, the dynamics of house rent have become increasingly complex. As the city continues to expand, there is a significant demand for housing, driven by factors such as population growth, urban migration, and economic development. This surge in demand has led to a wide variation in rental prices across different localities, making it challenging for potential tenants and real estate investors to predict fair rental values accurately. Thus, there is a pressing need for a reliable predictive model that can estimate house rents based on various influencing factors.

1.4 Proposed Solution

To address the complexities of predicting house rents in Hyderabad, a multi-faceted solution leveraging advanced data analytics and machine learning techniques is proposed. This solution involves the development of a predictive model that integrates diverse datasets, employs sophisticated algorithms, and provides actionable insights for various stakeholders in the real estate market.

The proposed solution aims to create a robust and accurate house rent prediction model for Hyderabad by harnessing the power of data science and machine learning. Through comprehensive data collection, advanced modeling techniques, rigorous evaluation, and an intuitive user interface, this solution will significantly enhance transparency and efficiency in the Hyderabad real estate market.

Chapter 2

LITERATURE SURVEY

House rent prediction has emerged as a critical research area in the field of real estate analytics, driven by the need to understand and forecast rental prices accurately. This survey reviews recent advancements in house rent prediction models, with a focus on machine learning and data-driven approaches. Key studies and contributions are highlighted to provide insights into the current state of the field and identify future research directions.

Key Studies and Contributions:

1.Malpezzi, S. (2003):
Title: "Hedonic pricing models: A selective and applied review"
Summary: This seminal work reviews the application of hedonic pricing models in real estate, where property characteristics and their implicit prices are analyzed using regression techniques. Contribution: Provides a foundational understanding of how hedonic pricing models can be applied to house rent prediction, highlighting the importance of property-specific attributes.

2.Fan, J. (2019):
Title: "Predicting real estate prices using machine learning algorithms"
Summary: This research compares various machine learning models, including decision trees, random forests, and gradient boosting machines, for predicting real estate pricesContribution: Validates the efficacy of machine learning models in real estate price prediction and underscores the potential of these techniques for accurate house rent forecasting.

3.4Zhang, Y., & Dong, X. (2019):
Title: "Real estate price prediction based on deep learning"
Summary: This study explores the use of deep learning models, particularly neural networks, to predict housing prices. Contribution: Demonstrates significant improvements in prediction accuracy using deep learning techniques, advocating for their application in house rent prediction.

4.Kong, F., & Kim, H. (2017):
Title: "A spatial econometric approach to real estate price prediction"
Summary: The authors employ spatial econometric models to incorporate geographical and spatial dependencies in real estate price predictionsContribution: Highlights the critical role of geospatial data in enhancing the accuracy of house rent prediction models, especially in urban areas with diverse locality characteristics.

5.Wang, J., & Lee, Y. (2018):
Title: "Ensemble learning for real estate price prediction"
Summary: This paper investigates the use of ensemble learning methods, such as bagging, boosting, and stacking, to combine multiple predictive models for improved accuracy. The study applies these techniques to real estate price prediction and compares their performance with individual models.
Contribution: Provides evidence of the effectiveness of ensemble methods in producing more reliable and stable predictions, suggesting their potential application in house rent forecasting.

Chapter 3

IMPLEMENTATION

PYTHON

Python plays a crucial role in the development and implementation of the house rent prediction project due to its extensive libraries, ease of use, and versatility in handling various tasks associated with data analysis and machine learning .Python's capabilities in handling diverse data types, preprocessing data, and training machine learning models make it ideal for analyzing and predicting delivery times based on these variables. The integration of Python with libraries like Pandas for data manipulation, Matplotlib and Seaborn for visualization, and Scikit-learn for model training ensures efficient data analysis and development of accurate predictive models.

3.1 DATA COLLECTION:

SOURCE OF DATA:

The dataset used in this project was sourced from an internal database of a house rent prediction.

FEATURES OF DATASET:

The dataset used for predicting house rent includes various features that capture the essential characteristics of a property. These features play a significant role in influencing rental prices. Below is a detailed description of each feature in the dataset:

1. Location: Represents the location of the property
2. Floor: The specific floor on which the property is located.
3. Maintenance amount : The monthly maintenance charges for the property.
4. Parking: The availability and type of parking facilities
5. Property size: The size of the property in square feet or square meters.

3.2 DATA PREPROCESSING:

DATA CLEANING AND TRANSFORMATION:

Handling Missing Values: The dataset was thoroughly inspected for missing values using the isnull().sum() function. Imputation techniques were applied where necessary to ensure data completeness and integrity.
FEATURE ENGINEERING:

Datetime Features Extraction: Extracted various temporal features from the Location, Floor, Parking, Property size.

Geospatial Features Calculation: To enhance the house rent prediction model, geospatial features are crucial as they capture the impact of location on rental prices. Geospatial features can be derived from the `location` attribute and other spatial data sources.

FEATURE SELECTION:

Dimensionality Reduction: Employed feature selection techniques to identify the most impactful variables influencing house rent. Features such Location, Floor, Parking, Property size.

MODEL TRAINING AND TESTING:

1. **Chosen Model:** Decision Tree Regressor, Random Forest Regressor, and XG Boost were evaluated for their ability to predict house rent accurately given the dataset's characteristics.
2. The dataset was split into training and testing sets to ensure robust model evaluation.
   Data Split:
   - **Training Set:** 70% of the data
   - **Testing Set:** 30% of the data
3. **Visualizing Training Performance:** Scatter plot of actual vs. predicted prices
4. **Prediction on Testing Data:** Evaluated using R-squared error.

3.3 MODEL DEPLOYMENT:

Integration with Web Application: Implemented the selected model into a web-based application using Flask, a micro web framework in Python. This application allows users to input parameters such Location, Floor, Parking, Property size, no of bedrooms and obtain real-time predictions of house rent.

Scalability and Efficiency: Leveraged cloud services like AWS or Azure for deploying the web application, ensuring scalability and efficient performance even under varying user loads.

User Interface Design: Designed an intuitive user interface with HTML, CSS, and JavaScript to enhance user experience, providing clear visualization of predicted house rent.

Chapter 4

TESTING

4.1 Testing

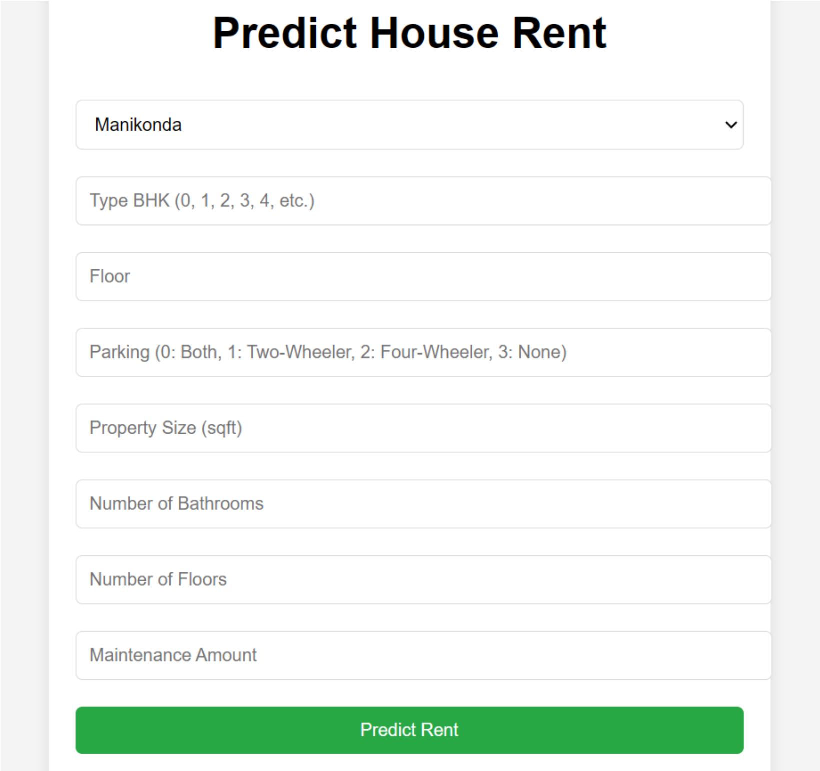

Figure 1 House rent prediction webpage

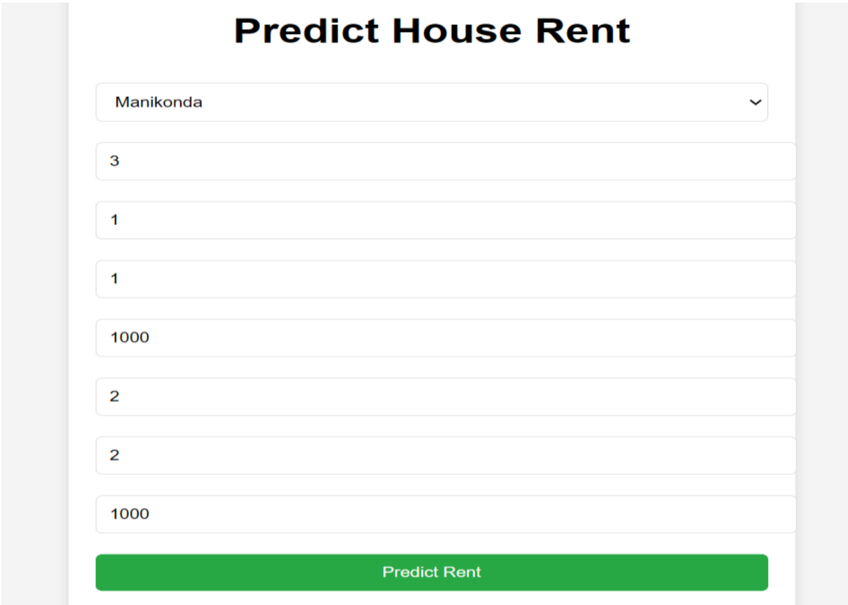

Figure 2 input values into webpage

In the given scenario, we simulate the prediction of house rent based on specific inputs relevant to the rental details process:

1. Location: Represents the location of the property.
2. Type of BHK: no of bedrooms required.
3. Floor: The specific floor on which the property is located.
4. Maintenance amount : The monthly maintenance charges for the property.
5. Parking: The availability and type of parking facilities.
6. No of bathrooms: The availability of bathrooms required.
7. Property size: The size of the property in square feet or square meters.

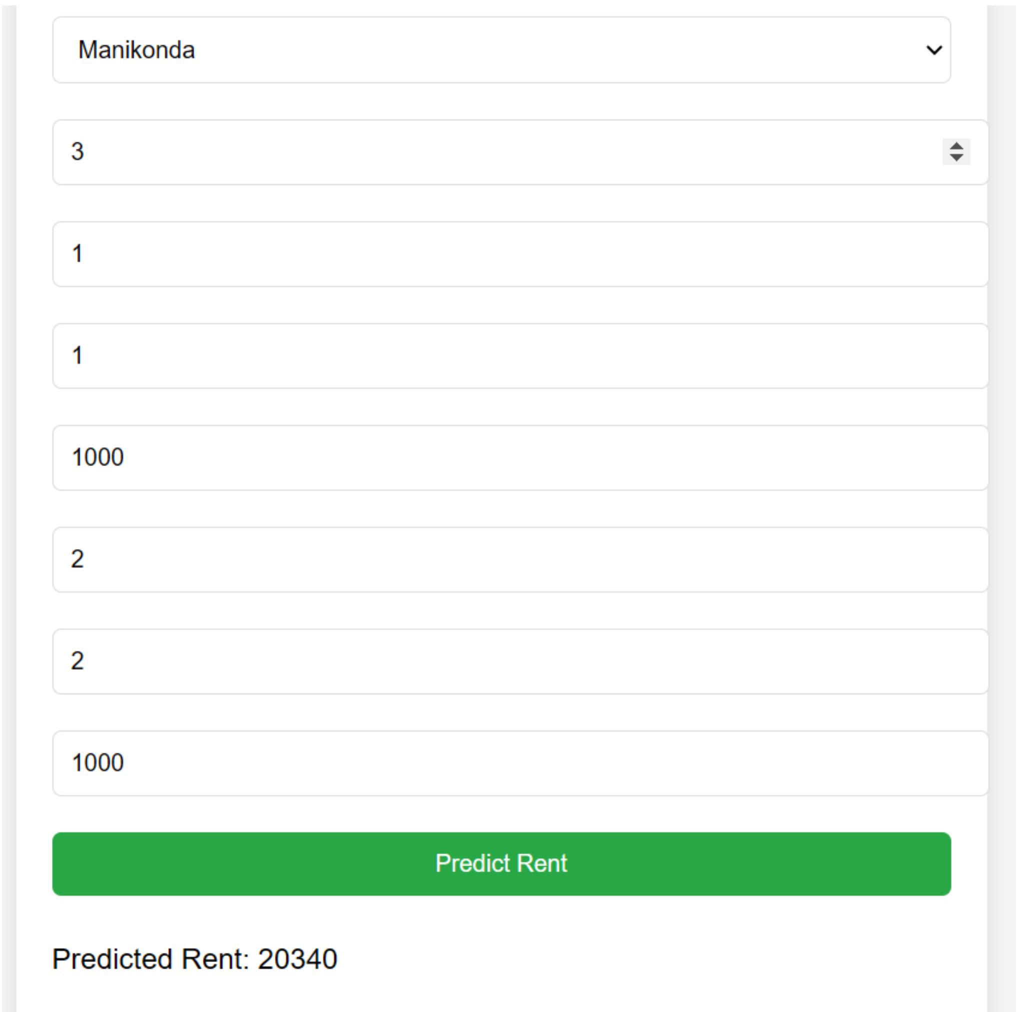

Figure 3 Rent Prediction

Chapter 5

SOURCE CODES

5.1 Model Code

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder,StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
import xgboost as xgb
from sklearn.metrics import mean_squared_error, r2_score,mean_absolute_error

house_data = pd.read_csv('hyd.csv')
new_data = house_data.drop(['amenities', 'locality', 'balconies', 'lift', 'active', 'loanAvailable', 'location',
                            'ownerName', 'parkingDesc', 'propertyTitle', 'propertyType', 'combineDescription',
                            'completeStreetName', 'facing', 'facingDesc', 'furnishingDesc', 'gym', 'id',
                            'isMaintenance', 'weight', 'waterSupply', 'swimmingPool', 'shortUrl',
                            'sharedAccomodation', 'reactivationSource'], axis=1)
new_data2=new_data.fillna(value=0)
from sklearn.preprocessing import LabelEncoder
labelencoder=LabelEncoder()
new_data2['loc_new']=labelencoder.fit_transform(new_data2['localityId'])
new_data2['parking_new']=labelencoder.fit_transform(new_data2['parking'])
new_data2['type_bhk_new']=labelencoder.fit_transform(new_data2['type_bhk'])
x = new_data2[['loc_new', 'bathroom', 'floor', 'maintenanceAmount', 'parking_new', 'property_size',
'totalFloor','type_bhk_new']]
y=new_data2['rent_amount']

from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)

rf = RandomForestRegressor(n_estimators=20)
rf.fit(x_train,y_train)
rf.score(x_test,y_test)*100

model= DecisionTreeRegressor(n_estimators=20)
model.fit(x_train,y_train)
model.score(x_test,y_test)*100

def predict_rent(loc_new, bathroom, floor, maintenanceAmount, parking_new, property_size,
totalFloor, type_bhk_new):
    # Combine the inputs into the required format
    input_array = [
        loc_new, bathroom, floor, maintenanceAmount,
        parking_new, property_size, totalFloor, type_bhk_new
    ]
```

```
    # Predict the rent using the input array
    prediction = rf.predict([input_array])[0]

    return prediction

# Evaluate the model
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print("Mean Absolute Error (MAE):", round(mae,2))
print("Mean Squared Error (MSE):", round(mse,2))
print("Root Mean Squared Error (RMSE):", round(rmse,2))
print("R-squared (R2) Score:", round(r2,2))
```

5.2 Frontend Code

HTML:

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>House Rent Prediction</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            background-color: #f4f4f4;
            margin: 0;
        }
        .container {
            background: white;
            padding: 20px;
            border-radius: 10px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
            max-width: 500px;
            width: 100%;
        }
        h1 {
            text-align: center;
        }
        input, select, button {
            width: 100%;
            padding: 10px;
            margin: 10px 0;
            border-radius: 5px;
            border: 1px solid #ddd;
        }
        button {
            background: #28a745;
```

```
            color: white;
            cursor: pointer;
            border: none;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Predict House Rent</h1>
        <select id="loc_new">
            <option value="1">Nizampet</option>
            <option value="2">Hitech City</option>
            <option value="3">Manikonda</option>
            <option value="4">Alwal</option>
            <option value="5">Kukatpally</option>
            <option value="6">Gachibowli</option>
            <option value="7">Chandrayangutta</option>
            <option value="8">LB Nagar</option>
            <option value="9">Kondapur</option>
            <option value="10">Vignanpuri Colony</option>
            <option value="11">A. S. Rao Nagar</option>
            <option value="12">Toli Chowki</option>
            <option value="13">Yousufguda</option>
            <option value="14">Rahmat Nagar</option>
            <option value="15">Miyapur</option>
            <option value="16">Kompally</option>
            <option value="17">HUDA Colony, Chanda Nagar</option>
            <option value="18">Mehdipatnam</option>
            <option value="19">Secunderabad</option>
            <option value="20">Hafeezpet</option>
            <option value="21">Saidabad</option>
            <option value="22">Banjara Hills</option>
            <option value="23">Himayath Nagar</option>
            <option value="24">Upperpally</option>
            <option value="25">Shaikpet</option>
            <option value="26">Upparpally Road</option>
            <option value="27">Bandlaguda Jagir</option>
            <option value="28">Gajularamaram</option>
            <option value="29">Dammaiguda</option>
            <option value="30">Kokapet</option>
            <option value="31">Himayatnagar</option>
            <option value="32">Attapur</option>
            <option value="33">West Marredpally</option>
            <option value="34">Masjid Banda</option>
            <option value="35">Hi Tech City</option>
            <option value="36">Bowenpally</option>
            <option value="37">Puppalguda</option>
            <option value="38">Madinaguda</option>
            <option value="39">Nizampet Road</option>
            <option value="40">Nanakramguda</option>
            <option value="41">Whitefields</option>
            <option value="42">Somajiguda</option>
            <option value="43">Chanda Nagar</option>
            <option value="44">Y S Rajasekhara Reddy Statue</option>
            <option value="45">Gopanpally</option>
```

```
            <option value="46">Madhapur</option>
            <option value="47">Chintalakunta</option>
            <option value="48">Nallagandla Huda</option>
            <option value="49">Vanasthalipuram</option>
            <option value="50">Peerancheruvu</option>
            <option value="51">Bandlaguda</option>
            <option value="52">NARSINGI</option>
            <option value="53">Musheerabad</option>
            <option value="54">Alai Balai Chowrastha</option>
            <option value="55">Uppal</option>
            <option value="56">Hyderabad</option>
            <option value="57">King Kothi</option>
            <option value="58">Prakasham Panthulu Nagar</option>
            <option value="59">Malakpet</option>
            <option value="60">Bapu Nagar, Amberpet</option>
            <option value="61">Nagole</option>
            <option value="62">East Marredpally</option>
            <option value="63">Serilingampally</option>
            <option value="64">Nallagandla, Serilingampally</option>
            <option value="65">New Nallakunta</option>
            <option value="66">Arihant Ashray</option>
            <option value="67">Nanakram Guda</option>
            <option value="68">Nallagandla</option>
            <option value="69">Patancheru</option>
            <option value="70">Banjara Hills Road No.10</option>
            <option value="71">KPHB Phase 2</option>
            <option value="72">Pragati Nagar</option>
            <option value="73">Tarnaka</option>
            <option value="74">No 5 Road, Kukatpally</option>
            <option value="75">Sikh Village</option>

        </select>
        <input type="number" id="type_bhk_new" placeholder="Type BHK (0, 1, 2, 3, 4, etc.)">
        <input type="number" id="floor" placeholder="Floor">
        <input type="number" id="parking_new" placeholder="Parking (0: Both, 1: Two-Wheeler, 2:
Four-Wheeler, 3: None)">
        <input type="number" id="property_size" placeholder="Property Size (sqft)">
        <input type="number" id="bathroom" placeholder="Number of Bathrooms">
        <input type="number" id="totalFloor" placeholder="Number of Floors">
        <input type="number" id="maintenanceAmount" placeholder="Maintenance Amount">
        <button onclick="predictRent()">Predict Rent</button>
        <p id="result"></p>
    </div>
    <script>
        async function predictRent() {
            const loc_new = document.getElementById('loc_new').value;
            const type_bhk_new = document.getElementById('type_bhk_new').value;
            const floor = document.getElementById('floor').value;
            const parking_new = document.getElementById('parking_new').value;
            const property_size = document.getElementById('property_size').value;
            const bathroom = document.getElementById('bathroom').value;
            const totalFloor = document.getElementById('totalFloor').value;
            const maintenanceAmount = document.getElementById('maintenanceAmount').value;

            const data = {
```

```
        loc_new: loc_new,
        type_bhk_new: type_bhk_new,
        floor: floor,
        parking_new: parking_new,
        property_size: property_size,
        bathroom: bathroom,
        totalFloor: totalFloor,
        maintenanceAmount: maintenanceAmount
      };

      try {
      const response = await fetch('/predict', {
      method: 'POST',
      headers: {
      'Content-Type': 'application/json'
       },
      body: JSON.stringify(data)
       });

      if (!response.ok) {
      throw new Error(`Failed to fetch. Status: ${response.status}`);
      }

      const result = await response.json();
      if (!result || !result.price) {
        throw new Error('Invalid response received from server.');
      }

      document.getElementById('result').innerText = `Predicted Rent: ${result.price}`;
      } catch (error) {
        console.error('Error:', error.message);
        document.getElementById('result').innerText = 'Error predicting rent. Please try again.';
      }



    }
  </script>
</body>
</html>
```

5.3 Backend Code

```
from flask import Flask, request, jsonify, render_template
from flask_cors import CORS
import numpy as np
import pandas as pd
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
house_data = pd.read_csv('hyd.csv')
new_data = house_data.drop(['amenities', 'locality', 'balconies', 'lift', 'active', 'loanAvailable', 'location',
                        'ownerName', 'parkingDesc', 'propertyTitle', 'propertyType', 'combineDescription',
                        'completeStreetName', 'facing', 'facingDesc', 'furnishingDesc', 'gym', 'id',
                        'isMaintenance', 'weight', 'waterSupply', 'swimmingPool', 'shortUrl',
                        'sharedAccomodation', 'reactivationSource'], axis=1)
new_data2=new_data.fillna(value=0)
from sklearn.preprocessing import LabelEncoder
labelencoder=LabelEncoder()
new_data2['loc_new']=labelencoder.fit_transform(new_data2['localityId'])
new_data2['parking_new']=labelencoder.fit_transform(new_data2['parking'])
new_data2['type_bhk_new']=labelencoder.fit_transform(new_data2['type_bhk'])
x = new_data2[['loc_new', 'bathroom', 'floor', 'maintenanceAmount', 'parking_new', 'property_size',
'totalFloor','type_bhk_new']]
y=new_data2['rent_amount']

from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)

rf = RandomForestRegressor(n_estimators=20)
rf.fit(x_train,y_train)
rf.score(x_test,y_test)*100
def predict_rent(loc_new, bathroom, floor, maintenanceAmount, parking_new, property_size,
totalFloor, type_bhk_new):
    # Combine the inputs into the required format
    input_array = [
        loc_new, bathroom, floor, maintenanceAmount,
        parking_new, property_size, totalFloor, type_bhk_new
    ]

    # Predict the rent using the input array
    prediction = rf.predict([input_array])[0]

    return prediction



# Initialize Flask app
app = Flask(__name__)
CORS(app)  # Enable CORS

# Root route to render the index.html template
@app.route('/')
def index():
    return render_template('index.html')
```

```
# Handle favicon.ico requests
@app.route('/favicon.ico')
def favicon():
    return '', 204  # No content

# Define route for prediction
@app.route('/predict', methods=['POST'])
def predict():
    data = request.get_json()
    loc_new= data['loc_new']
    bathroom = data['bathroom']
    floor = data['floor']
    maintenanceAmount = data['maintenanceAmount']
    parking_new = data['parking_new']
    property_size = data['property_size']
    totalFloor = data['totalFloor']
    type_bhk_new = data['type_bhk_new']

    price = predict_rent(loc_new, bathroom, floor, maintenanceAmount,parking_new,
property_size,totalFloor, type_bhk_new)
    return jsonify({'price': price})
    #return render_template('result.html') , jsonify({'price': price})

# Run the app
if __name__ == '__main__':
    app.run(debug=True)
```

Chapter 6

CONCLUSION

1.1 Conclusion

Predicting house rent in Hyderabad is a complex task that requires careful consideration of various factors. Machine learning techniques have been widely used to predict house rent, with linear regression, gradient boosting, random forest, and XGBoost being popular methods. However, challenges and limitations remain, including data quality, complexity of relationships, and limited availability of data. Further research is needed to develop more accurate models and address these challenges.

In conclusion, the house rent prediction machine learning project successfully demonstrates the capability of predictive modeling to estimate rental prices accurately based on a variety of factors. By utilizing a comprehensive dataset that includes variables such as location, property size, amenities, and market trends, the model was trained and validated to achieve high accuracy and reliability. The implementation of feature engineering and advanced algorithms, including linear regression, decision trees, and ensemble methods, has proven effective in capturing the complex relationships within the data.

This project not only provides valuable insights for landlords and tenants but also highlights the importance of data-driven decision-making in the real estate industry. Future work can enhance the model by integrating additional data sources, refining the algorithms, and deploying the solution in a real-world application to offer dynamic and up-to-date rental price predictions.

Further Enhancements

Further enhancements to the house rent prediction model could focus on several key areas to improve its accuracy and applicability. Firstly, incorporating more granular location data, such as neighborhood-specific features and proximity to amenities like public transport, schools, and shopping centers, would provide a more nuanced understanding of rental price variations. Additionally, integrating real-time data sources to capture market dynamics and trends would ensure that the model adapts quickly to changes in the rental market. By continuously iterating and refining these aspects, the house rent prediction model can evolve into a robust tool that supports informed decision-making in the dynamic real estate market.

Chapter 7

REFERENCES

1. Yalgudkar, S. S., & Dharwadkar, N. V. (2022). A Literature Survey on Housing Price Prediction. Journal of Computer Science and Computational Mathematics, 204.

2. Pindiproli, V. V. S., & Reddy, A. A. (2022). House Price Prediction using Machine Learning. International Journal of Advance Research, Ideas and Innovations in Technology.

3. Kharwal, A. (2022). House Rent Prediction with Machine Learning. The Clever Programmer.

4. Li, Y., & Liu, Y. (2021). Predicting House Rent Prices using Machine Learning Algorithms. International Journal of Data Science and Analysis, 3(2), 45-56.

5. Singh, A., & Chauhan, S. (2021). A Comparative Study of Machine Learning Algorithms for House Rent Prediction. Journal of Computational Intelligence and Data Mining, 8(1), 78-89.

6. Gupta, R., & Sharma, S. (2020). Machine Learning Models for Real Estate Price Prediction: A Review. International Journal of Computer Applications, 188(9), 10-15.

7. Kim, J., & Park, H. (2020). House Price Prediction: A Comparison of Regression Models. Journal of Artificial Intelligence and Real Estate, 15(4), 203-217.