SunRidge Medical

Date: 2/18/2026

Re: Shahbaz Ahmed Raza

Message:

Dear Shahbaz Ahmed Raza,

Please find the attached documents from SunRidge Medical.

Best regards,
SunRidge Medical Team

2 document(s) attached:
1. Fax_Cover_Page_20260218_133805.pdf
2. Real Estate AI Model Documentation.pdf

SUNRIDGE MEDICAL
14200 N Northsight Blvd, Suite 160,
Scottsdale, AZ, 85260
480-659-9135
info@sunridgemedical.com

Fax Recipient
Shahbaz Ahmed Raza
Fax: +19725329272
Date: 02/18/2026

FAX TRANSMISSION

FAX DETAILS

TO: Shahbaz Ahmed Raza
FAX NUMBER: +19725329272
FROM: Sunridge Medical
TOTAL PAGES: 2 (including this cover page)
DATE: 02/18/2026 1:38 PM

MESSAGE

Dear Shahbaz Ahmed Raza,

Please find the attached documents from SunRidge Medical.

Best regards,
SunRidge Medical Team

ATTACHED DOCUMENTS

1. Real Estate AI Model Documentation.pdf

Real Estate AI Model Documentation

Introduction

This documentation provides a comprehensive explanation of the real estate price prediction model and natural language query system. This AI system is designed to help users understand real estate pricing in Pakistan and query property information using everyday language.

Table of Contents

System Overview

What does this AI system do?

This AI system serves two main purposes:

1. **Price Prediction**: It predicts real estate prices based on features like location, size, number of bedrooms, etc.
2. **Natural Language Understanding**: It allows users to ask questions in everyday language and get meaningful answers.

Why is it useful?

- **For Buyers**: Helps understand fair market prices before making purchase decisions
- **For Sellers**: Assists in setting appropriate listing prices
- **For Agents**: Provides quick insights about market trends and property availability
- **For Researchers**: Enables analysis of real estate market patterns

How does it work?

The system works in three main stages:

1. **Data Analysis**: Processes real estate data to identify patterns
2. **Machine Learning**: Uses these patterns to make predictions about property prices
3. **Natural Language Processing**: Translates user questions into specific data queries

Data Processing

The Data We're Using

Our system works with Pakistani real estate data that includes:

- **Property details**: Size, bedrooms, bathrooms, property type
- **Location information**: City, area/society, coordinates
- **Price data**: Actual listing prices
- **Additional features**: Purpose (sale/rent), date added

Cleaning the Data

Raw real estate data often contains problems that would confuse our AI:

- **Invalid values**: Prices of 0 or 1 (clearly errors)
- **Extreme outliers**: Properties priced at billions (likely typos)
- **Inconsistent formats**: Different ways of writing the same information
- **Missing information**: Incomplete property details

Our cleaning process:

1. Removes properties with impossible values (0 bedrooms, 0 bathrooms)
2. Caps extremely high prices and areas to reasonable values
3. Standardizes property types (e.g., "Flat" and "Apartment" are treated the same)
4. Fills critical missing information where possible

Feature Engineering

To help our model make better predictions, we create new useful information from the existing data:

- **Price per unit area**: How much each marla or square foot costs
- **Bedroom-to-bathroom ratio**: Relationship between rooms and bathrooms

- **Area per bedroom/bathroom**: How spacious the property is
- **Logarithmic transformations**: Mathematical adjustments that help the model work with wide price ranges

Model Training

What is XGBoost?

We use an algorithm called **XGBoost** (Extreme Gradient Boosting) for our predictions. Here's why:

- **Superior accuracy**: Consistently outperforms other algorithms for real estate pricing
- **Handles complex relationships**: Understands non-linear connections between features
- **Robust to outliers**: Not easily confused by unusual properties
- **Feature importance**: Tells us which factors most affect property prices
- **GPU acceleration**: Can use graphics cards to learn faster

How the Model Learns

The training process involves:

1. **Splitting data**: We divide our data into a "training set" (what the model learns from) and a "test set" (to verify its accuracy)
2. **Feature preprocessing**: Converting all property information into numbers the model can understand
3. **Model configuration**: Setting up the XGBoost algorithm with appropriate parameters
4. **Training**: The model analyzes thousands of properties to identify pricing patterns
5. **Evaluation**: We measure how accurately it predicts prices on properties it hasn't seen before

Evaluation Metrics

We use several measurements to check our model's performance:

- **Mean Absolute Error (MAE)**: The average difference between predicted and actual prices
- **Root Mean Squared Error (RMSE)**: Similar to MAE but penalizes large mistakes more heavily
- **R^2 Score**: How much of the price variation our model explains (closer to 1.0 is better)

Natural Language Processing

Understanding User Questions

The system can interpret natural language questions through:

1. **Query classification**: Determining if the user wants a price prediction, property count, or other information
2. **Entity extraction**: Identifying important elements in the question (bedrooms, city, area, etc.)
3. **Pattern matching**: Recognizing common phrases and numeric values
4. **Context understanding**: Determining what the user is actually asking for

Types of Questions Supported

Our system handles:

- **Price predictions**: "What is the price of a 3 bedroom house in DHA Lahore?"
- **Property counts**: "How many houses are available for rent in Islamabad?"
- **Market statistics**: "What's the average price of 10 marla houses in Karachi?"
- **Specific property searches**: "Find me a 5 marla house with 3 bedrooms in Bahria Town"

Response Enhancement

Improving Responses with AI Language Models

Our system uses **Llama 4** (through the Together API) to transform technical results into friendly, informative responses:

1. **Context creation**: We provide the raw result and user query to the language model
2. **Prompt engineering**: We instruct the model on how to respond helpfully and accurately
3. **Response generation**: The model creates a natural-sounding answer with the essential information
4. **Fallback mechanisms**: If the language model fails, we still provide basic information

Benefits of Enhanced Responses

- **Natural conversation**: Feels like talking to a knowledgeable real estate expert
- **Additional context**: Can provide market insights beyond the raw numbers
- **Clearer explanations**: Helps users understand what factors influenced the prediction
- **Suggested refinements**: Guides users to ask better questions if initial query was unclear

Technical Components

Libraries and Tools Used

Our system relies on several specialized software libraries:

- **pandas & numpy**: For data manipulation and numerical operations

- **scikit-learn**: For data preprocessing and model evaluation
- **XGBoost**: For the core prediction algorithm
- **Gradio**: For creating the user interface
- **Together API**: For connecting to the Llama 4 language model

System Architecture

The system is organized in layers:

1. **Data layer**: Cleaned dataset with real estate information
2. **Model layer**: Trained XGBoost model and preprocessing components
3. **NLP layer**: Query understanding and entity extraction
4. **Enhancement layer**: Llama 4 integration for better responses
5. **Interface layer**: User-friendly chat interface

Code Explanation

Here's a breakdown of the main components in our code:

1. Model Loading

```python
# Load the model from the local file
with open('real_estate_model_package.pkl', 'rb') as f:
    model_package = pickle.load(f)

xgb_model = model_package['xgb_model']
preprocessor = model_package['preprocessor']
feature_names = model_package['feature_names']
```

What this does: This code opens our saved model file and loads three important components:

- **xgb_model**: The trained XGBoost model that makes predictions
- **preprocessor**: A component that converts raw property features into a format the model understands
- **feature_names**: The list of property characteristics the model uses

2. Query Processing Class

```python
class RealEstateQueryProcessor:
    # Class code...
```

What this does: This is the core of our system that:

- Interprets natural language questions
- Extracts relevant property features
- Determines if the user wants a price prediction or property count
- Processes the data to answer the question

3. Entity Extraction

```python
def _extract_entities(self, query):
    entities = {}

    # Extract numeric patterns
    for entity, pattern in self.patterns.items():
        matches = re.findall(pattern, query.lower())
        if matches:
            # Processing code...

    # Extract city, property type, etc.
    # ...
```

What this does: This function finds important information in the user's question:

- Number of bedrooms and bathrooms
- Property size (in marla or square feet)
- Location (city and society/area)
- Property type (house, apartment, etc.)
- Purpose (rent or sale)

4. Price Prediction

python

```
def _predict_price(self, entities):
    # Create sample for prediction
    sample = pd.DataFrame([{key: value for key, value in entities.items()
                            if key in self.df.columns}])

    # Fill missing values, transform features, predict price
    # ...
```

What this does: This function:

- Creates a representation of the property the user is asking about
- Fills in any missing details with reasonable values
- Transforms the data into the format our model understands
- Gets a price prediction from the XGBoost model

5. Property Counting

python

```
def _count_properties(self, entities):
    # Create query conditions
    query_conditions = []

    for key, value in entities.items():
        if key in self.df.columns:
            # Build query
            # ...

    # Filter and count
    # ...
```

What this does: This function:

- Creates filters based on what the user is looking for
- Applies these filters to our database of properties
- Counts how many properties match the criteria
- Calculates the average price if requested

6. Response Enhancement

```python
def enhance_response_with_llm(query, result):
    # Format context for LLM
    # ...

    # Call the Together API
    response = client.chat.completions.create(
        model="meta-llama/Llama-4-Maverick-17B-128E-Instruct-FP8",
        messages=[{"role": "user", "content": prompt}]
    )
```

What this does: This function:

- Takes the raw result from our model
- Creates a context description for the language model
- Sends this to the Llama 4 model via the Together API
- Gets back a natural-sounding, informative response

7. User Interface

```python
# Create Gradio interface
iface = gr.Interface(
    fn=chatbot,
    inputs=gr.Textbox(placeholder="Ask a question..."),
    outputs=gr.Textbox(label="Response"),
    # ...
)
```

What this does: This code creates a simple web interface where:

- Users can type their real estate questions
- The system processes the question and shows the enhanced response
- Example questions are provided to help users get started

Usage Examples

Example 1: Price Prediction

User Query:

```
What is the price of a 3 bedroom house in DHA Lahore?
```

System Process:

1. Extracts: bedrooms=3, city="Lahore", society="DHA", property_type="House"
2. Determines this is a price prediction query
3. Creates a property sample with these features
4. Gets prediction from XGBoost model (e.g., 25,000,000 PKR)
5. Enhances with Llama 4 for a natural response

Enhanced Response:

```
Based on current market trends, a 3-bedroom house in DHA Lahore is estimated
to cost around PKR 25,000,000. This area is known for its premium locations and
well-planned infrastructure, which contributes to the property values. The number
of bedrooms is a significant factor, with each additional bedroom typically adding
15-20% to the property's value in this area.
```

Example 2: Property Count

User Query:

```
How many houses are available for rent in DHA Lahore?
```

System Process:

1. Extracts: purpose="Rent", city="Lahore", society="DHA", property_type="House"
2. Determines this is a count query
3. Filters the dataset based on these criteria
4. Counts matching properties and calculates average price
5. Enhances with Llama 4 for a natural response

Enhanced Response:

```
I found 298 houses currently available for rent in DHA Lahore. The average
rental price is PKR 220,356 per month. DHA Lahore is one of the most sought-after
residential areas in the city, offering good security and amenities. Rental prices
can vary significantly based on the phase, with Phase 5 and 6 typically commanding
higher rents due to their central location.
```

Presentation Guide

When presenting this project, emphasize these key points:

1. Problem Statement

"Real estate pricing is complex and often lacks transparency. Our AI system addresses this by providing accurate price predictions and answering natural language queries about the market."

2. Technical Innovation

"Our system combines advanced machine learning (XGBoost) for predictions with state-of-the-art language models (Llama 4) to create a seamless, conversational experience for users."

3. Business Value

"This technology can help real estate platforms, agencies, buyers, and sellers make more informed decisions, reducing information asymmetry in the market."

4. Demonstration

Show live examples of:

- Price prediction for different property types and locations
- Counting available properties with specific criteria
- How the system handles variations in how questions are asked

5. Future Potential

Discuss possible enhancements:

- Incorporating time-based trends for future price forecasting
- Adding visual elements like maps and charts
- Expanding to more regions or property types
- Integration with existing real estate platforms

By explaining both the technical aspects and practical benefits, you'll demonstrate the value of this AI system even to those without a technical background.