Financial Assistant Web Application for a Real Estate Company

Objective
Build a complete Financial Assistant Web Application for a selected real estate company. The platform must combine structured data querying, classic machine learning through regression and classification models, and generative AI capabilities. This assignment evaluates your ability to assemble an end to end AI driven system that retrieves financial and property insights, performs predictive analytics, and provides natural language summaries using cloud hosted LLMs.

Web Application Interface Requirements

Create a clean and intuitive web interface, preferably using Streamlit.

The interface must include:
• A conversational chat panel for natural language questions
• A results section that displays retrieved financial data, property data, press release summaries, and predictive outputs
• A separate section that demonstrates regression and classification predictions from the ML models you will build

---

Data Sources and Backend System Requirements

Your system must pull information from three distinct sources.

A. SEC EDGAR Financial Reports
• Use annual and quarterly regulatory filings such as ten K and ten Q reports
• Retrieve and extract metrics such as revenue, net income, and operating expenses

B. Postgres Database for Properties and Financials
Create two tables as shown below.

Properties
• property_id (primary key)
• address

- metro_area
- sq_footage
- property_type

Financials
- property_id (foreign key)
- revenue
- net_income
- expenses

Populate at least 20 sample property records.

C. Company Press Releases
- Scrape, download, or mock recent press releases
- Store them in JSON or a structured table
- Extract insights such as acquisitions, expansions, or quarterly business updates

---

Classic Machine Learning Models

The Financial Assistant must include two classic machine learning components. These models represent structured prediction capabilities that complement the chatbot.

Both models must be trained locally and then deployed to a cloud platform such as Amazon SageMaker.

Regression Task

Dataset
Use the California Housing dataset from scikit learn. This dataset contains numeric features that allow prediction of median house values.

Model
Use Random Forest Regressor.

Steps
- Load the dataset using scikit learn
- Perform exploratory data analysis
- Normalize or standardize numeric features where appropriate
- Split into train and test subsets
- Train a Random Forest Regressor using reasonable defaults
- Evaluate using metrics such as:

• Root Mean Squared Error
• Mean Absolute Error
• R squared score
• Save the model artifact
• Create an inference script that accepts feature inputs and returns predicted housing values

Deployment
Deploy the trained model to Amazon SageMaker as a hosted endpoint.
Your Streamlit interface must be able to send sample inputs to the SageMaker endpoint and display the predicted housing value.

Classification Task

Dataset
Use the Bank Marketing dataset from the UCI Machine Learning Repository.
This dataset supports a binary classification task where the goal is to predict whether a customer will subscribe to a service based on demographic and interaction data.

Model
Use a Logistic Regression classifier.

Machine Learning Pipeline for Financial Assistant

The assignment requires developing a robust classification model for the financial assistant. The core steps include:

1. **Data Preparation and Exploration:**
    - Load and initially inspect the dataset.
    - Conduct comprehensive exploratory data analysis (EDA).
    - Pre-process the data by:
        - Encoding all categorical variables.
        - Normalizing all numeric variables.
    - Split the prepared data into distinct training and testing subsets.
2. **Model Development and Training:**
    - Train a Logistic Regression classifier on the training data.
    - Save the resulting trained model as an artifact.
3. **Model Evaluation:**
    - Evaluate the model's performance on the test set using a comprehensive set of metrics:
        - Accuracy
        - Precision
        - Recall
        - F1 Score

        - Confusion Matrix
4. **Deployment Component:**
    - Create a complete inference script that takes a set of input features and returns the model's classification prediction along with the corresponding probability score.

Deployment
Deploy the trained classifier to Amazon SageMaker.
Your Streamlit interface must allow input of customer features and display a subscription prediction along with probability scores.

AI Powered Chatbot with Vertex AI ADK

Use GCP Vertex AI and the Agent Development Kit to build a chatbot capable of:
• Interpreting natural language questions
• Determining which data source to query
• Fetching financial, property, or press release insights
• Providing natural language answers using a generative model
• Combining results across datasets

Example queries include:
• What was the net income reported last quarter
• Show industrial properties in the Chicago region with revenue details
• Did the company announce any acquisitions recently

Multi Cloud Awareness

Demonstrate basic familiarity with cloud systems outside GCP by adding at least one small integration such as:
• AWS Bedrock for summarization or fallback model support
• Azure Cognitive Services or Azure OpenAI for basic extraction or summarization

---

Expected Deliverables

Screen Recorded Demo
• End to end demonstration of the application
• Chatbot conversations

• Property queries
• Press release summaries
• Regression and classification predictions from SageMaker hosted endpoints

Cloud Configuration Recording
• Vertex AI ADK setup
• SageMaker model deployment steps
• Any Azure or AWS supplemental components

Deployment URL
• Public link to the hosted application

Google Drive Folder
• Source code
• Requirements file
• SQL scripts
• Training notebooks
• Model artifacts
• Inference code

README Documentation
Include:
• Setup instructions for local and cloud environments
• Architecture and data flow explanation
• Details on how the chatbot routes queries
• Training and deployment steps for both ML models

Evaluation Criteria

You will be evaluated on:
• Completion of the end to end system
• Correctness of the regression and classification models
• Quality of the SageMaker deployments
• Effective use of Vertex AI ADK
• Data modeling quality
• UI clarity and flow
• Multi cloud awareness
• Code quality and documentation clarity