Technical Challenge: Real Estate Price Prediction Engine

Background

We're building an AI-powered real estate platform. A core component requires accurate property valuations based on real transaction data.

Your task is to build a production-ready price prediction model that will power our valuation engine.

Dataset Provided

You'll receive a CSV file containing **real property transactions** with the following attributes:

Field	Description
`TRANSACTION_NUMBER`	Unique transaction identifier
`INSTANCE_DATE`	Transaction date
`GROUP_EN`	Property group category
`PROCEDURE_EN`	Transaction type (e.g., Sale, Mortgage)
`IS_OFFPLAN_EN`	Off-plan vs Ready property
`IS_FREE_HOLD_EN`	Freehold vs Leasehold
`USAGE_EN`	Property usage type
`AREA_EN`	Area/Community name
`PROP_TYPE_EN`	Property type (Villa, Apartment, etc.)

Field	Description
`PROP_SB_TYPE_EN`	Property sub-type
`TRANS_VALUE`	**Transaction price (target variable)**
`PROCEDURE_AREA`	Registered area in transaction
`ACTUAL_AREA`	Actual built-up area (sqm/sqft)
`ROOMS_EN`	Number of rooms/bedrooms
`PARKING`	Number of parking spaces
`NEAREST_METRO_EN`	Nearest metro station
`NEAREST_MALL_EN`	Nearest shopping mall
`NEAREST_LANDMARK_EN`	Nearest landmark
`TOTAL_BUYER`	Number of buyers in transaction
`TOTAL_SELLER`	Number of sellers in transaction
`MASTER_PROJECT_EN`	Master development project
`PROJECT_EN`	Specific project/building name

Your Mission

Build a machine learning model that predicts `TRANS_VALUE` (property price) with the highest possible accuracy while being explainable and production-ready.

Deliverables (Timeline: 72 hours)

1. Jupyter Notebook with Complete Analysis (Required)

A. Exploratory Data Analysis (25% weight)

- Dataset overview: size, date range, transaction distribution
- Handle missing values and data quality issues
- Statistical analysis of `TRANS_VALUE` distribution
- Identify and handle outliers (justify your approach)
- Key insights:
- Price trends by property type, area, and time
- Impact of location features (metro, mall, landmarks)
- Off-plan vs ready property pricing differences
- Freehold vs leasehold impact

B. Feature Engineering (30% weight)

- How will you handle categorical variables (area names, project names, landmarks)?
- Time-based features from `INSTANCE_DATE` (seasonality, trends)
- Derived features (e.g., price per sqm, area popularity scores)
- Location encoding strategy (one-hot, target encoding, embeddings?)
- Treatment of high-cardinality features (projects, landmarks)
- Feature selection and importance analysis

C. Model Development (30% weight)

- Justify your model selection (Linear, Tree-based, Ensemble, Neural Network?)
- Training methodology:
- Train/validation/test split strategy
- Cross-validation approach
- Hyperparameter tuning process
- Model performance metrics:
- R^2 Score
- MAE (Mean Absolute Error)

- MAPE (Mean Absolute Percentage Error)
- Price range-wise accuracy breakdown
- Error analysis: where does the model fail and why?

D. Model Interpretability (15% weight)

- Feature importance visualization
- SHAP values or similar explainability technique
- Example predictions with reasoning
- Business insights from model learnings

---

2. FastAPI Prediction Service (Bonus - 20% extra credit)

Create a simple REST API with the following:

Endpoint: `POST /api/v1/predict-price`

Request Body:

```
{
  "property_type": "Apartment",
  "property_subtype": "Flat",
  "area": "Marina District",
  "actual_area": 1200,
  "rooms": 2,
  "parking": 1,
  "is_offplan": false,
  "is_freehold": true,
  "usage": "Residential",
  "nearest_metro": "Central Station",
  "nearest_mall": "City Mall",
  "master_project": "Marina Development",
  "project": "Marina Residence"
}
```

Response:

```
{
  "predicted_price": 1850000,
  "confidence_interval": {
    "lower": 1750000,
    "upper": 1950000
  },
  "price_per_sqft": 1541,
  "model_confidence": "high",
  "key_factors": [
    "Location: Premium area",
    "Property size: 1200 sqft",
    "Proximity to metro station"
  ]
}
```

Requirements: - Input validation with clear error messages - Model loaded from saved file (pickle/joblib) - Include a `/health` endpoint - Basic error handling - README with setup instructions

---

3. Documentation (10% weight)

Provide a `REPORT.md` covering:

1. Executive Summary

- Model performance headline metrics
- Key findings from the data
- Recommended approach for production

2. Technical Decisions

- Why you chose your modeling approach
- Trade-offs considered (accuracy vs interpretability vs speed)
- Handling of real estate market specifics

3. Production Readiness Assessment

- Model limitations and edge cases
- Data requirements for maintaining accuracy
- Recommended retraining frequency
- How to handle properties not in training data (new projects, areas)

4. Future Improvements

- What additional data would improve predictions?
- Suggested enhancements for v2
- Scalability considerations

Evaluation Criteria

Criterion	Weight	What We're Looking For
Model Performance	30%	Accuracy metrics, proper validation, realistic results
Feature Engineering	25%	Creative & effective handling of location/categorical data
Code Quality	20%	Clean, documented, reproducible code
Business Understanding	15%	Market insights, practical recommendations
Communication	10%	Clear explanations, visualizations, documentation
Bonus: API	+20%	Working FastAPI implementation

Submission Instructions

Create a GitHub repository with:

```
├── notebooks/
│   └── analysis.ipynb          # Main analysis notebook
```

```
├── src/
│   ├── model.py                # Model training code
│   ├── preprocessing.py        # Data preprocessing functions
│   └── api.py                  # FastAPI code (if bonus attempted)
├── models/
│   └── trained_model.pkl       # Saved model
├── REPORT.md                   # Your documentation
├── requirements.txt            # Python dependencies
└── README.md                   # Setup instructions
```

Share: GitHub repository link via email

Questions & Clarifications

Feel free to ask about: - Dataset specifics or data quality issues - Business requirements or priorities - Technical stack preferences - Scope clarifications

However, your approach to ambiguity and assumptions (clearly documented) is part of the evaluation.

Success Looks Like

✅ A model that achieves strong predictive performance with explainable predictions
✅ Thoughtful handling of real estate market characteristics
✅ Clean, reproducible code a team could build upon
✅ Clear communication of limitations and trade-offs
✅ Practical recommendations grounded in data insights

Timeline

72 hours from dataset receipt

Final Notes

This challenge is designed to mirror the actual work you'll be doing in this role. We're not looking for perfection - we're looking for:

- **Problem-solving approach**: How you break down and tackle the problem
- **Technical skills**: Your ability to build effective ML models
- **Communication**: How you explain your decisions and findings
- **Production mindset**: Thinking beyond notebooks to real-world deployment

Good luck! We're excited to see your approach to this real-world problem.