Task 2: Baseline Predictive Model

Objective:

Develop and evaluate machine learning models to predict apartment sale prices, following structured data science practices.

Dataset:

Kaggle Sberbank housing prices dataset
https://www.kaggle.com/competitions/sberbank-russian-housing-market/overview

Process

1. Define Model Requirements
    - Understanding the problem: Begin by outlining the business or research question this model aims to answer. In this case, it would be predicting apartment sale prices for potential buyers, sellers, or real estate investors.
    - Kaggle Competition Context:
        - Familiarize yourself with the format of a Kaggle competition's test set.
        - Determine the evaluation metric used (likely Root Mean Squared Error (RMSE) or a similar error metric).
        - Understand how model submissions are made and scored.
2. Data Cleaning (`Full_sq, life_sq, floor, max_floor, material, build_year, num_room, Kitch_sq, state, product_type, sub_area`)
    - Exploratory Data Analysis (EDA):
        - Calculate descriptive statistics (mean, median, standard deviation, min, max) for numerical features.
        - Examine frequency distributions of categorical features.
        - Generate visualizations (histograms, boxplots, scatterplots) to uncover patterns and potential anomalies.
    - Missing Values:
        - Identify missing values in each feature.
        - Develop appropriate imputation strategies (mean/median imputation, predictive modeling, or careful removal if justifiable).
    - Outliers:

        - Use box plots or statistical techniques (e.g., IQR method) to detect potential outliers.
        - Consider the impact of outliers and whether to remove, cap, or transform them.
    - Data Transformation (if needed):
        - Apply normalization or scaling to numerical features if required by certain models.
        - Explore one-hot encoding or label encoding for categorical features.

3. Basic Feature Engineering
    - Derived Features:
        - Create new features based on existing ones:
            - Price per square meter (price_doc / full_sq)
            - Rooms per square meter (num_room / full_sq)
            - Age of the building (timestamp - build_year)
    - Feature Interactions:
        - Try combining features in meaningful ways (e.g., multiplying or dividing related features to create ratios).

4. Accounting for Price Changes over Time
    - Analyze Temporal Trend: Plot sale prices over time to visualize any trends or seasonality.
    - Choose the data manipulation to deal with the temporal aspects of the dataset

5. Baseline Model Development and Evaluation
    - Split Data: Divide your dataset into training, validation, and testing sets (use an appropriate ratio like 70/15/15).
    - Random Forest Model
        - Instantiate a RandomForestRegressor
        - Feature Selection:
            - Use feature importance to identify the most important features.
            - Decide on the optimal number of features for the dataset
        - Train the model on the training set.
        - Calculate MSE on validation set using 5-Fold Cross-Validation
    - XGBoost Model
        - Instantiate an XGBoostRegressor
        - Hyperparameter Tuning
            - Use techniques like GridSearchCV or RandomizedSearchCV to optimize hyperparameters (e.g., learning rate, max_depth, n_estimators)
        - Train the model on the training set.

        - Calculate MSE on validation set using 5-Fold Cross-Validation.
6. Kaggle Submission and Evaluation
    - Prediction: Generate predictions on the Kaggle competition's test set using both of your trained models.
    - Submission Format: Adhere to Kaggle's submission file format requirements.
    - Submit and Score: Submit each prediction file and obtain your leaderboard scores.

Submission Instructions:

1. **Google Colab Usage**:
    a. Sign in to your Google account and go to Google Colab.
    b. Click on `File` > `New Notebook` to create a new notebook for your homework.
    c. Import pandas and any other required libraries at the beginning of your notebook.
    d. Write your code to solve each task in separate cells. Make sure to include comments or markdown cells of the original questions
    e. Please verify that when running the notebook using the “Run all” command in the Runtime tab.

2. **Exporting Your Notebook**:
    a. After completing all tasks, ensure your notebook is neatly organized and all cells run as expected.
    b. Click on `File` > `Download` > `Download .ipynb` to export your notebook as a ipynb document.
    c. Click on `File` > `Print`, and choose “print as pdf” printer
3. **Submission**:
    a. Submit the exported IPYNB and PDF document