

OJT Project Design Template

Student Name: Kasula Mahendra

Roll No: 240410700110

Year & Section: 2nd Year, A

Project Title (as assigned): California Housing — Linear Regression vs Random Forest.

Project Type: Data Scientist

Stack / Framework: Python + scikit-learn

1. Problem Understanding

1.1 What is the problem statement in your own words?

(Explain what you are trying to solve, not what you're trying to build.)

The goal is to predict the median house value in California districts using various tabular features like location, housing age, number of rooms, population, and median income. This involves comparing different regression approaches (linear vs non-linear) to understand which models work better for housing price prediction and why.

1.2 Why does this problem exist or matter?

(Who benefits from the solution — user, developer, community, etc.?)

Housing price prediction is a fundamental problem in real estate and economics. Accurate predictions help:

- *Buyers/Sellers: Make informed decisions about property values*
- *Real Estate Agents: Price properties competitively*
- *Policy Makers: Understand housing affordability and market trends*
- *Data Scientists: Learn trade-offs between interpretable linear models and more accurate but complex non-linear models*

1.3 Key inputs and expected outputs:

Inputs	Process	Expected Outputs
--------	---------	------------------

California Housing dataset from Kaggle (camnugent/california-housing-prices) with features: longitude, latitude, housing_median_age, total_rooms, total_bedrooms, population, households, median_income, ocean_proximity	1. EDA & data cleaning 2. Handle missing values 3. Feature scaling (StandardScaler) 4. Train-test split (80-20) 5. Train Linear Regression 6. Train Random Forest 7. Train Lasso (stretch) 8. Generate residual plots 9. Calculate RMSE & R ² 10. Extract feature importances	Jupyter notebook with: • Trained models (Linear Regression, Random Forest, Lasso) • Residual plots • Feature importance visualizations • Model comparison table (RMSE, R ²) • Insights on price drivers
--	---	--

2. Functional Scope

2.1 What are the core features you plan to build (must-haves)?

- EDA with multicollinearity checks - Create correlation matrix and heatmap to identify highly correlated features
- Linear Regression baseline - Train a simple linear model with feature scaling (StandardScaler) as a performance baseline
- Random Forest model - Train a non-linear ensemble model to compare against linear approach
- Residual analysis plots - Generate predicted vs actual plots and residual vs predicted plots to diagnose model errors
- Model comparison - Calculate and compare RMSE and R² metrics for both models to determine which performs better

2.4 What stretch goals could you attempt if time permits?

- Add Lasso regression for automatic feature selection to identify which features are most important
- Compare 2-3 models total (Linear Regression, Random Forest, and Lasso) instead of just two
- Geographic visualization - Create scatter plots showing house prices by longitude/latitude to see spatial patterns
- Correlation heatmaps - Advanced visualizations to better understand feature relationships
- Hyperparameter tuning - Use GridSearchCV or RandomizedSearchCV for Random Forest optimization

2.3 Which libraries or tools will you use?

pandas, numpy, scikit-learn (LinearRegression, RandomForestRegressor, Lasso),
matplotlib, seaborn

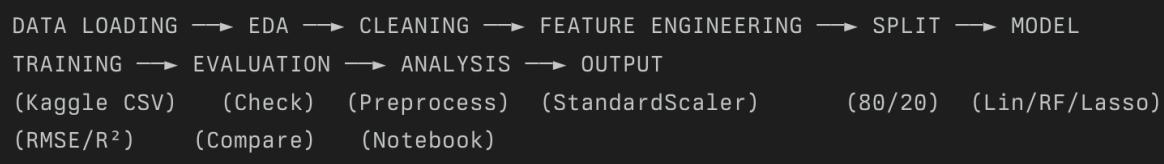
Key Points:

- Section 2.1 focuses on what you MUST complete for the project to be successful
- Section 2.2 is for "nice-to-have" features if you have extra time
- Section 2.3 lists the specific Python libraries you'll use

This aligns perfectly with your project from the spreadsheet and shows clear scope boundaries for your 4-week timeline!

3. System & Design Thinking

3.1 Sketch or describe your app flow / pipeline:



3.2 What data structures or algorithms are central to this project?

Data Structures:

- Pandas DataFrames - For storing and manipulating tabular housing data
- NumPy arrays - For numerical computations and model input/output
- Scikit-learn estimators - Model objects (LinearRegression, RandomForestRegressor, Lasso)
- Python dictionaries - For storing evaluation metrics and model comparisons

Algorithms:

- Ordinary Least Squares (OLS) - Used by Linear Regression to minimize squared errors
- Random Forest - Ensemble of decision trees with bootstrap aggregating (bagging)
- Lasso Regression - L1-regularized linear regression for automatic feature selection
- Train-Test Split - Random sampling algorithm for data splitting
- StandardScaler - Feature normalization using mean and standard deviation

3.3 How will you test correctness or performance? (e.g., unit tests, metrics like accuracy, latency, etc.)

Validation Strategy:

1. *Train-Test Split*
 - Use 80-20 or 70-30 split to ensure models are tested on unseen data
 - Never train and test on the same data (avoid overfitting)
2. *Performance Metrics*
 - RMSE (Root Mean Squared Error) - Measures average prediction error in dollars
 - R^2 (R-squared) - Measures how much variance the model explains (0-1 scale)
 - MAE (Mean Absolute Error) - Average absolute difference between predictions and actual values
3. *Residual Analysis*
 - Plot predicted vs actual values (should be close to diagonal line)
 - Plot residuals vs predicted values (should show random scatter, not patterns)
 - Check for heteroscedasticity (unequal variance in residuals)
4. *Sanity Checks*
 - Predictions should be positive (house prices can't be negative)
 - Predictions should be in reasonable range (e.g., \$50k - \$800k for California)
 - Feature importances should make domain sense (e.g., median_income should be important)
5. *Model Comparison*
 - Compare metrics across all models
 - Analyze trade-offs: interpretability (Linear) vs accuracy (Random Forest)
 - Document which model performs best and why

4. Timeline & Milestones (4 Weeks)

Week	Planned Deliverables	Mentor Checkpoint
W1	EDA + Check Multicollinearity - Load California Housing dataset from Kaggle	<input type="checkbox"/>

	<ul style="list-style-type: none"> - Explore data distributions, missing values, outliers - Create correlation matrix and heatmap - Identify highly correlated features - Document initial findings in notebook 	
W2	<p>Linear Baseline + Residuals</p> <ul style="list-style-type: none"> - Handle missing values (imputation/removal) - Encode categorical features (ocean_proximity) - Apply StandardScaler for feature scaling - Train Linear Regression model - Generate residual plots (predicted vs actual, residual vs predicted) - Calculate RMSE and R² metrics 	<input type="checkbox"/>
W3	<p>Random Forest + Lasso Compare</p> <ul style="list-style-type: none"> - Train Random Forest Regressor - Extract and visualize feature importances - (Stretch) Train Lasso regression for feature selection - Compare all 2-3 models side-by-side - Create model comparison table 	<input type="checkbox"/>
W4	<p>Insights + What Drives Price</p> <ul style="list-style-type: none"> - Analyze which features most impact house prices - Document interpretability vs accuracy trade-offs - Write conclusions and insights - Polish notebook with markdown explanations - Prepare final presentation/demo - Create README with project summary 	<input type="checkbox"/>

5. Risks & Dependencies

5.1 What's the hardest part technically for you right now?
(Be honest — data cleaning, unfamiliar library, deployment, etc.)

Main Challenges:

- *Interpreting residual plots - Understanding what patterns mean and how to fix model issues*
- *Feature engineering decisions - Knowing which features to create or transform*

- *Model comparison - Understanding trade-offs between interpretability (Linear) vs accuracy (Random Forest)*
- *Explaining WHY models perform differently, not just reporting metrics*

5.2 What dependencies or help do you need from mentors?

(E.g., feedback on model metrics, setup issues, data clarification.)

Need Help With:

- *Reviewing residual plots and diagnosing model problems*
- *Guidance on feature engineering (what derived features make sense?)*
- *Feedback on model evaluation approach and metrics interpretation*
- *Best practices for presenting ML results professionally*
- *Domain knowledge about California housing price drivers*

Dependencies:

- *Kaggle account for dataset access*
 - *Python environment (pandas, scikit-learn, matplotlib)*
 - *Weekly mentor check-ins (30 min)*
 - *Jupyter Notebook/Google Colab access*
-

6. Evaluation Readiness

6.1 How will you prove that your project “works”?

(Screenshots, metrics table, demo video, test cases, Deployment links and Git Links, etc.)

Proof of Working Project:

1. *Complete Jupyter Notebook*
 - *All cells execute without errors from top to bottom*
 - *Clear outputs showing data, visualizations, and model results*
2. *Trained Models*
 - *Working Linear Regression model*
 - *Working Random Forest model*
 - *(Stretch) Working Lasso model*
3. *Visualizations*
 - *Correlation heatmap*
 - *Residual plots (predicted vs actual, residual vs predicted)*
 - *Feature importance bar charts*
 - *Model comparison table*

4. Performance Metrics

- RMSE and R^2 calculated for all models
- Clear comparison showing which model performs best

5. Documentation

- Markdown cells explaining each step
- README with project overview and instructions
- Insights on what drives California house prices

6.2 What success metric or goal will you aim for?

(E.g., accuracy > 90%, PWA Lighthouse > 85, 100% CRUD functionality.)

Quantitative Metrics:

- Model Performance:
 - Linear Regression $R^2 > 0.60$ on test set
 - Random Forest $R^2 > 0.75$ on test set
 - RMSE within reasonable range for California housing prices
- Code Quality:
 - Notebook runs without errors
 - All visualizations render correctly
 - Proper train-test split (no data leakage)
- Deliverables:
 - Complete Jupyter notebook (100+ lines of code)
 - At least 5 meaningful visualizations
 - Model comparison table with RMSE and R^2 for each model

Qualitative Metrics:

- Understanding:
 - Can explain why Random Forest outperforms Linear Regression
 - Can identify top 3-5 features that drive house prices
 - Can interpret residual plots and what they reveal
- Communication:
 - Clear markdown explanations throughout notebook
 - Professional visualizations with labels and titles
 - Well-written README and insights summary
- Project Completeness:
 - All 4 weekly milestones achieved
 - Stretch goal attempted (Lasso)
 - Ready to present findings in 5-10 minutes

7. Responsibilities

7.1 Responsibilities

Task	Student Name	Mentor Notes
Task 1	<input type="checkbox"/>	
Task 2	<input type="checkbox"/>	
Task 3	<input type="checkbox"/>	
Task 4	<input type="checkbox"/>	
Task 5	<input type="checkbox"/>	



Signatures (Students):

Mentor Approval:

Date: