# ECE 133A Final Project

Build and explain a least-squares predictor

## 1 What you are making

This project gives you hands-on practice with the least-squares, feature-engineering, and validation ideas from ECE 133A. Choose a real prediction problem, construct three core least-squares predictors, and use five-fold cross-validation to compare them. Document intermediate stages when the engineered model is built in steps, then write a four-page scientific paper that explains what the comparison taught you about the problem. Your task can predict a numeric outcome, a class, or another clearly defined target.

Every project needs a future use, because the historical rows already have their outcomes. Explain what will be unknown for a future case, which information will be available before that outcome, who will use the prediction, and which decision it informs. AI use is encouraged for dataset exploration, code, debugging, and critique, but you remain responsible for every choice and claim in the paper. Submit one four-page PDF; figures, tables, and references count toward the page limit.

## 2 Set up Overleaf and the paper template

Most engineering researchers write papers in LaTeX, and Overleaf provides a collaborative browser-based LaTeX editor. Using this workflow gives you experience writing equations, figures, tables, citations, and source files in the format common in engineering research. The supplied USENIX layout is the same kind of conference-paper format you will encounter in research. The handout, fill-in template, and worked example are separate files so that you can see how a LaTeX project is organized.

1. Go to UCLA's Overleaf portal. Sign in through UCLA Single Sign-On, or link an existing Overleaf account to your UCLA affiliation.
2. Open the ECE 133A USENIX starter project linked on the course site. Choose **Menu** and then **Copy Project** so that you edit your own copy.
3. Open `main.tex` and change `\showhandouttrue` to `\showhandoutfalse`. This setting displays the paper template alone in the PDF you develop and submit.
4. Edit `ECE133A-overleaf-template-main.tex`. The left pane contains LaTeX source and the right pane shows the compiled PDF; the green **Recompile** button requests a fresh build.
5. Replace every blue bracketed prompt with your own project-specific prose, figures, tables, and citations. Keep the USENIX layout and section structure.
6. When the paper is complete, choose **Menu** and **Download PDF**. Submit that compiled PDF through the course submission system.

## 3 Step 1: Choose the dataset and future-use task

Choose one real, public, documented dataset with at least **1,000 usable observations** and at least **three candidate predictor variables before feature engineering**. Select observation units, fields, and measurement limitations that you can explain clearly. Build the predictor from fields available before the future outcome occurs. Connect its target and prediction-time information to a decision or comparison that becomes more informed.

Before modeling, write down the answers that will become the core of your Introduction and Dataset subsection:

- What does one observation represent?
- What outcome is unknown for a future observation? Is it a number, a class, or another defined target?
- Which fields are available before that outcome is known?
- Who would use the prediction, what decision or comparison would it improve, and why does that improvement matter?

- What makes this prediction problem interesting? Which available features might separate future outcomes, and what variation or uncertainty must the predictor handle?
- What is the dataset's original source and documentation URL?
- What are the candidate predictors and their units or categories?

## 4 Step 2: Plan the three least-squares models

The baseline, base, and final engineered models show the predictive contribution of increasingly specific information for the same future-use task. Decide the model sequence and the feature-engineering idea before comparing cross-validation results. Each model uses fields available at the time of the future prediction. A staged final model can include intermediate planned additions, and its retained engineered feature passes the column check in Step 3.

1. **Naïve baseline.** Use a constant rule with no input features. For regression, predict the training-fold mean outcome. For classification, use a training-fold majority class or class-probability baseline. State the rule that turns a least-squares score into a class: for a binary target, a common rule is $\hat{c} = 1[\hat{y} \geq 0.5]$; for multiple classes, choose the largest fitted class score.
2. **Base model.** Fit a least-squares model using original predictor variables. Explain why those variables are available when the future prediction is made.
3. **Feature-engineered model.** Fit a least-squares model with at least one purposeful engineered feature: a transformation, interaction, threshold, polynomial, logarithm, or carefully constructed category feature. Explain the pattern it is intended to capture and the assumption it adds. If the construction adapts a method from research literature, cite that work and state what you adapted.

## 5 Step 3: Form meaningful folds and check feature columns

Cross-validation works when each held-out fold represents the future cases for which the predictor will be used. If observations are exchangeable records from the same setting, randomly assign them to five folds and explain why that is a reasonable approximation. If a future case arrives later in time, select a cutoff $t_0$, fit on records with $t \leq t_0$, and evaluate on records with $t > t_0$; randomizing would place later information in training for an earlier prediction. If a future case belongs to a previously unseen person, device, product, organization, or location, give each record a group label $g(i)$ and place every record with the same $g(i)$ in one fold, so $G_{\text{train}} \cap G_{\text{test}} = \varnothing$ in every comparison.

Feature columns also need a separate mathematical check. For each training fold, begin with the intercept and process the remaining columns in model order; for column $a_j$, compute $q_j = a_j - \text{proj}_{\text{span}(q_1,\ldots,q_{j-1})}(a_j)$ and report $\|q_j\|$. A zero or near-zero remainder identifies a redundant proposed column; redesign or remove that column, then retain a different nonredundant engineered contribution. Explain the result in ordinary language, including how category indicators stay distinct from the intercept.

## 6 Step 4: Compare the models with five-fold cross-validation

Five-fold cross-validation estimates how a fixed choice of features generalizes to held-out observations for exchangeable or grouped records. Keep the Step 3 fold assignment fixed for every model comparison. For fold $k$, fit each model on the other four folds and evaluate it on fold $k$. For a numeric target, pool every held-out residual $r_i$ across the five folds and report cross-validation RMS $\sqrt{\sum_i r_i^2/N}$ in the target's units; this construction weights each held-out observation equally when fold sizes differ.

For a time-prediction task, use chronological out-of-sample validation instead: fit on an earlier block of observations and evaluate on a later block. For a class target, report the fraction correctly classified on each held-out fold and include a confusion matrix that shows which classes are mistaken for which. If one class is rare, report the class counts and explain why raw accuracy alone can hide errors on that class. Compare the fitted coefficients across folds when they are interpretable; similar coefficients strengthen the case that the feature choice is stable.

## 7 Step 5: Evaluate the models

One comparison table makes the planned model sequence visible across all five folds. Explain what changed from one model to the next and whether the new feature earned its added assumption through lower held-out error or better held-out classification. Add one diagnostic that shows where predictions succeed or fail: residuals, predicted-versus-actual values, or errors across a feature range for regression; a confusion matrix or class-specific errors for classification. Identify one observation type or setting where the predictor performs less well, and discuss unusually large errors or influential observations when applicable.

## 8 Step 6: Make a substantive finding

The Results section gives a substantive conclusion about the prediction problem. A regression project can show a nonlinear pattern captured by an engineered feature or compare fitted predictions across a meaningful feature range. A classification project can show which classes, conditions, or input patterns remain difficult to distinguish. Tie the finding to a figure or table and keep the conclusion within the coverage of the dataset.

## 9 Step 7: State a limitation and a next test

Every predictor is tied to the records used to evaluate it. Name the actual dataset structure that limits the conclusion, the exact scope boundary or evaluation failure it creates, and one next split, diagnostic, measurement, or data source that would address it. Identify the future observations or settings for which the current predictor is least trustworthy.

## 10 Step 8: Fill in the four-page paper

The supplied USENIX template gives the paper a coherent scientific structure. Use its prompts to help a reader follow the future-use case, data choices, feature construction, cross-validation comparison, and substantive finding. Replace prompts with your own paper's prose, figures, tables, and citations. Keep the final report concise, visually legible, and focused on one coherent claim.

1. **Title and abstract.** Make the title state the paper's main finding. State the data type and usable sample size, the planned model sequence and cross-validation design, the practical prediction result, and the substantive finding.
2. **Introduction.** Use three paragraphs, each with four to six complete sentences. Develop the future task and utility, then the prediction-time fields and why the held-out folds represent future cases, then the model progression and main finding.
3. **Methodology.** Use **Dataset and prediction-time information**, **Three least-squares models and feature design**, and **Feature-column independence** subsections. Document the data source, direct URL, observation unit, usable sample size, target, candidate predictors, model equations, engineered feature, and Gram–Schmidt calculation.
4. **Validation.** Describe the five folds, the chosen metric, the random/group/time construction, and the coefficient stability or variation across folds when applicable.
5. **Results.** Use **Model evaluation** for the planned-model-sequence cross-validation comparison. Use **Data findings and error analysis** for a domain conclusion, a diagnostic, and less-successful cases.
6. **Limitations and next step.** State the dataset-specific boundary and one concrete next test.
7. **References.** Cite the dataset documentation with its direct URL and every research source that motivated a sophisticated feature construction. Include a related-work section when a source supplies domain context or motivates the feature design.

Write a concise, impersonal, data-grounded scientific paper. Every narrative paragraph has four to six complete sentences; its first sentence states a concrete takeaway, and every later sentence develops that point. Make tables and figures part of the argument, refer to them in the text, and begin every caption with a bold, project-specific takeaway followed by what is shown, its units or comparison, and reader relevance.

## 11 What strong work looks like

Technically sound work comes before an elegant story. A strong project uses a model sequence whose assumptions are explicit, a fold construction that matches the future use, and feature columns that remain nonredundant after projection. Its cross-validation table reports the same metric for every planned model and fold, and its figures and tables make the calculations interpretable and reproducible. The paper then uses those correct calculations to make a bounded claim about the problem.

Strong projects also tell one clear story: a future user has a specific prediction problem; the data and folds genuinely represent that problem; the engineered feature adds a defensible idea beyond the baseline; and the analysis explains both the main finding and its boundary. The worked Sacramento example illustrates the structure of such a paper while using a distinct topic and model design. Its 774 observations sit below this project's 1,000-observation minimum.

# [What substantive finding about the prediction problem can this title communicate?]

*[Author Name]*
*ECE 133A, University of California, Los Angeles*

**This is a template.** It includes all five paper sections. The guiding questions identify information that completes each section.

## Abstract

**[In a few short sentences: What is the dataset and the prediction task? Who needs the future prediction, and what decision does it inform? Which model is used and how accurate is it? What did that result reveal about the problem or the future decision?]**

## 1 Introduction

**[Write a few paragraphs for the introduction. Strive to answer the guiding questions from Section 3 of the assignment instructions.]**

## 2 Methodology

**[Use Sections 3–5 to guide the methodology. How do the records, model sequence, engineered feature, and column check construct this predictor?]**

### 2.1 Dataset and prediction-time information

**[Use Section 3. What records, future outcome, prediction-time fields, and source documentation define this problem?]**

### 2.2 Three least-squares models and feature design

**[Use Section 4. What are the baseline, base, and engineered models, and what pattern motivates the engineered feature?]**

Table 1. **[What does this table establish about the records used for the future prediction?] [Use Section 3 to name the observation, target, usable-record count, and prediction-time fields.]**

| Item | Project-specific description |
|---|---|
| Observation | **[What does one record represent?]** |
| Usable records | **[How many usable records remain?]** |
| Target | **[What outcome is unknown for a future record?]** |
| Candidate predictors | **[Which fields begin the model?]** |
| Prediction-time fields | **[What is known before the target occurs?]** |

**[Which plot helps a reader see the data pattern behind this prediction task?]**

Figure 1. **[What data pattern does this figure reveal?] [Use a distribution, a key predictor against the outcome, or time or group coverage; label the axes and connect the pattern to the future prediction.]**

$$\widehat{\text{target}}_{\text{baseline}} = \textbf{[baseline rule?]}$$

$$\widehat{\text{target}}_{\text{base}} = \textbf{[base rule?]}$$

$$\widehat{\text{target}}_{\text{engineered}} = \textbf{[engineered rule?]}$$

**[What is the prediction rule at each stage? What named field or transformation replaces each symbol, and what pattern does the engineered feature represent?]**

### 2.3 Feature-column independence

Table 2. **[What pattern does the engineered feature let the final predictor represent?] [Use Section 4 to define each final-model feature, its unit or coding, and its prediction-time value.]**

| Feature column | Definition, interpretation, and assumption |
|---|---|
| Intercept | **[What constant offset anchors the prediction?]** |
| **[Which first base feature enters?]** | **[How is it defined at prediction time?]** |
| **[Which second base feature enters?]** | **[How is it defined at prediction time?]** |
| **[Which engineered feature enters?]** | **[Which pattern and assumption does it add?]** |

Table 3. **[Which final feature columns add information beyond earlier columns?] [Use Section 5 to report Gram–Schmidt remainder lengths across the training folds and explain the smallest one.]**

| Column in processing order | Remaining length across five folds |
|---|---|
| Intercept | **[What is its minimum–maximum remainder length?]** |
| **[Which first feature enters the processing order?]** | **[What is its minimum–maximum remainder length?]** |
| **[Which second feature enters the processing order?]** | **[What is its minimum–maximum remainder length?]** |
| **[Which engineered feature enters the processing order?]** | **[What is its minimum–maximum remainder length?]** |

**[Use Section 5. What does the Gram–Schmidt calculation say about the information added by each final feature column?]**

## 3 Validation

**[Use Sections 5 and 6. How are the folds built, why do they represent future cases, and which metric measures prediction quality?]**

## 4 Results

**[Use Sections 7 and 8. What does model evaluation show, and what does the final model reveal about the problem?]**

### 4.1 Model evaluation

**[Use Section 7. How does each planned model perform across the same held-out folds, and which feature earns its place in the final predictor?]**

**[Which figure makes the held-out model comparison easy to read?]**

Figure 2. **[What held-out comparison does this figure make clear?] [Use a fold-by-fold error plot, a model-stage comparison, or predicted-versus-observed values; identify the models, metric, axes, and conclusion.]**

**[Which diagnostic makes the most important error pattern visible?]**

Figure 3. **[Where does the predictor make its largest errors or classification mistakes?] [Use the diagnostic ideas in Section 7; label the comparison and explain why the pattern matters for the future user.]**

### 4.2 Data findings and error analysis

**[Use Sections 7 and 8. What domain pattern appears in the data, and where does the final predictor make its largest errors?]**

## 5 Limitations and next step

**[Use Section 9. What boundary comes from these records, and what next split, diagnostic, or data collection would test it?]**

## References


[1] **[Where can a reader find the dataset documentation?]**

[2] **[Which source helps a reader understand the domain or feature design?]**

Table 4. **[Which stage in the planned model sequence performs best on held-out records?] [Use Section 7 to report the same metric for every stage and fold, then state the summary used to choose the final predictor.]**

| Model | Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5 | Summary |
|---|---|---|---|---|---|---|
| **[Baseline training?]** | **[F1?]** | **[F2?]** | **[F3?]** | **[F4?]** | **[F5?]** | **[All?]** |
| **[Baseline held-out?]** | **[F1?]** | **[F2?]** | **[F3?]** | **[F4?]** | **[F5?]** | **[All?]** |
| **[Base-model training?]** | **[F1?]** | **[F2?]** | **[F3?]** | **[F4?]** | **[F5?]** | **[All?]** |
| **[Base-model held-out?]** | **[F1?]** | **[F2?]** | **[F3?]** | **[F4?]** | **[F5?]** | **[All?]** |
| **[Engineered training?]** | **[F1?]** | **[F2?]** | **[F3?]** | **[F4?]** | **[F5?]** | **[All?]** |
| **[Engineered held-out?]** | **[F1?]** | **[F2?]** | **[F3?]** | **[F4?]** | **[F5?]** | **[All?]** |
| **[Additional training?]** | **[F1?]** | **[F2?]** | **[F3?]** | **[F4?]** | **[F5?]** | **[All?]** |
| **[Additional held-out?]** | **[F1?]** | **[F2?]** | **[F3?]** | **[F4?]** | **[F5?]** | **[All?]** |
| **[Additional training?]** | **[F1?]** | **[F2?]** | **[F3?]** | **[F4?]** | **[F5?]** | **[All?]** |
| **[Additional held-out?]** | **[F1?]** | **[F2?]** | **[F3?]** | **[F4?]** | **[F5?]** | **[All?]** |
| **[Additional training?]** | **[F1?]** | **[F2?]** | **[F3?]** | **[F4?]** | **[F5?]** | **[All?]** |
| **[Additional held-out?]** | **[F1?]** | **[F2?]** | **[F3?]** | **[F4?]** | **[F5?]** | **[All?]** |