Name: VISWABHARATHI KALAIARASI ANBAZHAGAN

USC ID: 5462914247

Read the following rules carefully:

- Write your name and ID number in the solution you submit.
- Please sign and submit the code of honor in the exam with your solutions. The exam cannot be graded without a signed code of honor. You are supposed to do all of the problems on your own without receiving help from others.
- Cheating in the exam will not be tolerated. If you are caught cheating, you will be reported to the authorities. The recommendation of the instructor will be at least an F in the course in such cases.
- Do not post any questions on Piazza about the exam. The TAs have been instructed to refrain from answering questions about the midterm. In case of ambiguity or problems in the questions, just do your best.
- The use of generative AI is prohibited in answering the questions.
- Problems are not sorted in terms of difficulty. Please avoid guess work and long and irrelevant answers.
- Instructions on submitting the solutions to paper and pencil and coding problems will be provided shortly by the TAs. You can handwrite or typeset your solutions to paper and pencil problems.
- Show all your work and your final answer. Showing only the final answer of a question may not receive full credit and you must show your solution and reasoning behind the answer. Simplify your answer as much as you can.
- The exam has 8 questions, 16 pages, and a total of 100 points.
- The submission deadline for this midterm is 11:59 PM, Friday, October 24, 2025.
- As this is a take home exam that extends over several days, OSAS students DO NOT receive extra time, per OSAS guidelines.
- Any change in the midterm (paper and pencil or coding) after the deadline is considered late submission. One second late is late. The midterm is graded based on *when it was submitted, not when it was finished.* The midterm can be submitted up to three days late, with 10% penalty per late day. Homework late days cannot be used for the midterm.
- Submission after the grace period will receive a zero. One second late is late.

Grading Breakdown

Problem	Score	Earned
1	10	
2	10	
3	10	
4	10	
5	10	
6	10	
7	20	
8	20	
Total	100	

Honor Code

I pledge on my honor that I have not given or received any unauthorized assistance on this examination.

Name: VISWABHARATHI KALAIARASI ANBAZHAGAN

Signature: 

1. You are working with a dataset where you train five polynomial regression models (Model A to E) of increasing complexity (degrees 1, 3, 5, 7, and 9 respectively) on the same training data of size $n = 80$. The performance of these models has been evaluated using 5-fold cross-validation. The table below shows the average training and validation Mean Squared Errors (MSE) and Mean Absolute Errors (MAE) for each model:

Model	Degree	Training MSE	Validation MSE
A	1	21.4	24.7
B	3	12.8	14.6
C	5	6.9	8.2
D	7	3.0	9.9
E	9	1.2	23.3

Instructions:

(a) Define and explain the mathematical concepts of bias and variance. How do they relate to model complexity in supervised learning?

(b) Using the above table, calculate the bias^2, variance, and Expected Prediction Error (EPE) for each model. Assume:

- Training MSE $\approx$ Variance
- Validation MSE $\approx$ Total EPE
- Irreducible Error = 1.0

(c) For $x = 2, 4, 6, 8$, use the following table of true function values $f(x)$ and model predictions $\hat{f}(x)$ from Model C. Compute the absolute prediction error and squared error for each x. Then compute the average squared error (MSE):

x	2	4	6	8
$f(x)$ (True)	5.0	9.5	13.0	16.5
$\hat{f}(x)$ (Predicted)	4.8	10.0	12.2	17.0
$\lvert f(x) - \hat{f}(x) \rvert$				
$(f(x) - \hat{f}(x))^2$				

(d) Interpret the trends observed in the table. Which model provides the best tradeoff between bias and variance? Justify your choice using comparisons across at least three models, and classify which models suffer from underfitting or overfitting.

(a) Bias:

It measures how far, on average, the model's predictions are from the true function. High bias means the model is too simple (underfit). Usually in supervised learning, models with low complexity (low degree polynomial) usually have high bias because they cannot fit the data well.

Variance:

It measures how sensitive the model is to the specific training data. High variance means the model changes drastically if the training data changes. Usually in supervised learning models with high complexity have high variance because they capture the noise too (overfit).

As model complexity increases:

Bias ↓ Variance ↑

There is a tradeoff and a sweetspot that balances this tradeoff.

(b) $EPE = (Bias)^2 + Variance + Irreducible\ error$

For each model:

Variance $\approx$ Training MSE

EPE $\approx$ Validation MSE

Irreducible error ≈ 1.0

$(bias)^2 = EPE - Variance - Irreducible\ error$

Model	Var	EPE	Irreducible error	$(bias)^2$
A	21.4	24.7	1	$24.7 - 21.4 - 1 = 2.3$
B	12.8	14.6	1	$14.6 - 12.8 - 1 = 0.8$
C	6.9	8.2	1	$8.2 - 6.9 - 1 = 0.3$
D	3	9.9	1	$9.9 - 3 - 1 = 5.9$
E	1.2	23.3	1	$23.3 - 1.2 - 1 = 21.1$

(c)

x	2	4	6	8
$f(x)$ (True)	5.0	9.5	13	16.5
$\hat{f}(x)$ (Predicted)	4.8	10	12.2	17.0
$\lvert f(x) - \hat{f}(x) \rvert$	0.2	0.5	0.8	0.5
$(f(x) - \hat{f}(x))^2$	0.04	0.25	0.64	0.25

$$\text{MSE} = \frac{0.04 + 0.25 + 0.64 + 0.25}{4} = \frac{1.18}{4} = 0.295$$

(d) From the table in (b)

	Training MSE	Val MSE	Bias^2
Model A	21.4	24.7	2.3
Model E	1.2	23.3	21.1
Model C	6.9	8.2	0.3
Model D	3	9.9	5.9
Model B	12.8	14.6	0.8

For model A both training and validation errors are very large (underfit).
For model E, it fits training set well but fails to generalize (overfit).

Model C has moderate training MSE and the lowest validation error among all models. The $bias^2$ is also small.

Best tradeoff - Model C (degree 5)
Underfitting - Model A (followed by ModelB)
Overfitting - Model E and Model D

2. Consider a logistic regression problem in which there are no features, which means that:

$$\Pr(Y=1) = \frac{e^{\beta_0}}{1+e^{\beta_0}}$$

Assume that we have m data points with label $Y=1$ and n data points with label $Y=0$ (remember that features are irrelevant).

(a) Write down the likelihood function $l(\beta_0)$.

(b) Find the Maximum Likelihood estimate $\hat{\beta}_0$ for this data set. [**Hint**: maximize $\log_e l(\beta_0)$].

(c) Determine conditions under which this simple classifier classifies data points into $Y=1$ or $Y=0$.

$\Pr(Y=1) = \frac{e^{\beta_0}}{1+e^{\beta_0}}$

m data points $\Rightarrow Y=1$

n data points $\Rightarrow Y=0$

(a) Bernoulli likelihood with m ones and n zeroes:

$$l(\beta_0) = p^m(1-p)^n = \left(\frac{e^{\beta_0}}{1+e^{\beta_0}}\right)^m \left(1-\frac{e^{\beta_0}}{1+e^{\beta_0}}\right)^n$$

$$= \left(\frac{e^{\beta_0}}{1+e^{\beta_0}}\right)^m \left(\frac{1}{1+e^{\beta_0}}\right)^n$$

$$= \frac{e^{m\beta_0}}{(1+e^{\beta_0})^{m+n}}$$

log - likelihood :

$$\log l(\beta_0) = \log \left(\frac{e^{m\beta_0}}{(1+e^{\beta_0})^{m+n}} \right)$$

$$= \log e^{m\beta_0} - \log (1+e^{\beta_0})^{m+n}$$

$$= m\beta_0 - (m+n) \log (1+e^{\beta_0})$$

(b) $l(\beta_0) = p^m (1-p)^n$

$$L(\beta_0) = \log l(\beta_0) = m \log p + n \log (1-p)$$

$$= m\beta_0 - (m+n) \log (1+e^{\beta_0})$$

from (a)

$$\frac{d L(\beta_0)}{d \beta_0} = 0 \quad \ldots\ldots \quad (1)$$

$$\frac{d L(\beta_0)}{d \beta_0} = m - (m+n) \frac{e^{\beta_0}}{1+e^{\beta_0}}$$

$$= m - (m+n) p$$

From ①

$$m - (m+n)p = 0$$

$$\boxed{p = \frac{m}{m+n}} \quad \ldots\ldots \text{②}$$

We know $\beta_0 = \ln\left(\frac{p}{1-p}\right)$

sub ② here

$$\hat{\beta}_0 = \ln\left(\frac{\frac{m}{m+n}}{1-\frac{m}{m+n}}\right) = \ln\left(\frac{m}{n}\right)$$

$$\boxed{\hat{\beta}_0 = \ln\left(\frac{m}{n}\right)}$$

checking it's a maximum

$$\frac{d^2L}{d\beta_0^2} = -(m+n)\frac{dp}{d\beta_0} \qquad \left(p = \frac{e^{\beta_0}}{1+e^{\beta_0}}, \ \frac{dp}{d\beta_0} = p(1-p)\right)$$

$$= -(m+n)p(1-p) \qquad (\text{As } 0<p<1, \ p(1-p) > 0)$$

so, $\frac{d^2L}{d\beta_0^2} < 0$

It is concave, hence it is a maximum

Edge cases:

$m=0 \Rightarrow \hat{\beta}_0 \to -\infty$

$n=0 \Rightarrow \hat{\beta}_0 \to \infty$

(c) if $\hat{p} \geq 0.5$ then $Y=1$ else $Y=0$

$\frac{m}{m+n} \geq 0.5 \Rightarrow 2m \geq m+n \Rightarrow m \geq n$

so, if $m \geq n$ the classifier predicts $Y=1$
else if $m < n$, the classifier predicts $Y=0$

In terms of $\hat{\beta}_0$:

$$\hat{\beta}_0 = \ln\left(\frac{m}{n}\right)$$

→ $m > n$: $\log > 0 \to \hat{\beta}_0 > 0 \to p > 0.5 \Rightarrow$ predict $Y=1$

→ $m < n$: $\log < 0 \to \hat{\beta}_0 < 0 \to p < 0.5 \to$ predict $Y=0$

→ $m = n$: $\log = 0 \to p = 0.5$

3. For the following data set for classification:

Index	X	Y
1	-1	1
2	0	0
3	3	0
4	1	1
5	-2	0

Assume that we want to construct a regularized logistic regression model for this dataset.

(a) Write down the $\mathscr{L}_1$-regularized loss function $J(\beta_0, \beta_1)$ for this dataset with regularization parameter $\lambda = 2$.

(b) Compare the bias variance of the regularized model with the unregularized model ($\lambda = 0$).

$$P(Y=1|x_i) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}} = p_i$$

(a) The negative log-likelihood loss on all data with L1 regularization and $\lambda=2$:

$$J(\beta_0,\beta_1) = \sum_{i=1}^{5} [-y_i \log(p_i) - (1-y_i)\log(1-p_i)] + 2|\beta_1|$$

Plugging in each (x_i, y_i) from the table.

→ For $x = -1, y = 1$

$$p_1 = \frac{e^{\beta_0 - \beta_1}}{1 + e^{\beta_0 - \beta_1}}$$

Loss component = $-(1) \log(p_1)$

→ For $x = 0, y = 0$

$$p_2 = \frac{e^{\beta_0}}{1 + e^{\beta_0}}$$

Loss component = $-(1) \log(1 - p_2)$

→ For $x = 3, y = 0$

$$p_3 = \frac{e^{\beta_0 + 3\beta_1}}{1 + e^{\beta_0 + 3\beta_1}}$$

Loss component = $-(1) \log(1 - p_3)$

→ For $x = 1, y = 1$

$$p_4 = \frac{e^{\beta_0 + \beta_1}}{1 + e^{\beta_0 + \beta_1}}$$

Loss component = $-(1) \log(p_4)$

→ For $x = -2, y = 0$

$$p_5 = \frac{e^{\beta_0 - 2\beta_1}}{1 + e^{\beta_0 - 2\beta_1}}$$

Loss component = $-(1) \log(1 - p_5)$

$$J(\beta_0, \beta_1) = \left[-\log \left(\frac{e^{\beta_0 - \beta_1}}{1 + e^{\beta_0 - \beta_1}} \right) \right] + \left[-\log \left(1 - \frac{e^{\beta_0}}{1 + e^{\beta_0}} \right) \right]$$

$$+ \left[-\log \left(1 - \frac{e^{\beta_0 + 3\beta_1}}{1 + e^{\beta_0 + 3\beta_1}} \right) \right] + \left[-\log \left(\frac{e^{\beta_0 + \beta_1}}{1 + e^{\beta_0 + \beta_1}} \right) \right]$$

$$+ \left[-\log \left(1 - \frac{e^{\beta_0 - 2\beta_1}}{1 + e^{\beta_0 - 2\beta_1}} \right) \right] + 2|\beta_1|$$

(b) Bias - Variance Comparison

For regularized model, we added $\lambda|\beta_1|$ component ($\lambda = 2$) with the negative log likelihood.

In unregularised model $\lambda = 0$.

So the regularized model discourages large $|\beta_1|$ and pushes β_1 towards 0.

This leads to:

→ Increase in bias and the model is less flexible because we are forcing small weights

→ Variance decreases as the model becomes less sensitive to noise in the training set as we limit the coefficient.

But in unregularized model ($\lambda = 0$).

→ The bias is lower as the model can fit the training data more closely.

→ The variance is higher as the β_1 can vary largely if the training data changes.

4. Consider the following data set for classification:

Index	X	Y
1	1	1
2	-1	0
3	-2	0

(a) Show all possible bootstrap samples of the dataset that have the same size as this dataset. Note that permutations of the same data set are considered the same dataset, for example $\{1, 2, 3\}$ and $\{2, 3, 1\}$ are the same dataset.

(b) Construct a KNN classification model for all bootstrap samples in part 4a with $K = 2$ and predict the label for the test point $x^* = 0$ by majority votes of the predictions of those bootstrap models. Break ties in favor of class 1.

A bootstrap sample - Sample 3 points with replacement from {1,2,3}

(a) All possible bootstrap samples

- {1,1,1}
- {2,2,2}
- {3,3,3}
- {1,1,2}
- {1,1,3}
- {2,2,1}
- {2,2,3}
- {3,3,1}

- {3,3,2}
- {1,2,3}

Sorting the unique bootstrap samples:

1) {1,1,1}
2) {1,1,2}
3) {1,1,3}
4) {1,2,2}
5) {1,2,3}
6) {1,3,3}
7) {2,2,2}
8) {2,2,3}
9) {2,3,3}
10) {3,3,3}

Each index corresponds to:

1 → (x=1, y=1)
2 → (x=-1, y=0)
3 → (x=-2, y=0)

(b) Constructing a KNN classification model for all bootstrap samples for test point $x^* = 0$

Distance is absolute distance on

the line $|x_{train} - x^*|$

- {1, 1, 1}

  Points : (1,1), (1,1), (1,1) (All samples are from class 1)

  Prediction = 1

- {1, 1, 2}

  Points : (1,1), (1,1), (-1,0)

  Distances : 1 1 1

  closest neighbors can be (1,1), (1,1) or (-1,0), (1,1).

  If we have labels 1 and 0 then we predict 1 as tie goes to class 1

  Prediction = 1

- {1, 1, 3}

  Points : (1,1), (1,1), (-2,0)

  Distance : 1, 1, 2

  Closest neighbors => (1,1), (1,1)

  Prediction = 1

- {1, 2, 2} *

  Points: (1, 1), (-1, 0), (-1, 0)

  Distance: 1, 1, 1

  Case A: Nearest neighbors → (-1, 0), (-1, 0)
  then we predict 0

  case B: If the model picks (1, 1), (-1, 0)
  then tie and prediction will be
  1 (Favours class 1)

But if we consider the rule of prioritising the duplicates ((-1, 0), (-1, 0)) when picking the two with smallest distance then prediction = 0

- {1, 2, 3}

  Points: (1, 1), (-1, 0), (-2, 0)

  Distance: 1, 1, 2

  Prediction = 1 (favour class 1)

- {1, 3, 3}

  Points: (1, 1), (-2, 0), (-2, 0)

  Distance: 1, 2, 2

  Prediction = 1 (favours class 1)

- {2,2,2}
  Points: (-1,0), (-1,0), (-1,0)
  Distance: 1, 1, 1
  Prediction = 0 (All are class 0)

- {2,2,3}
  Points: (-1,0), (-1,0), (-2,0)
  Distance: 1, 1, 2
  Prediction = 0 (All are class 0)

- {2,3,3}
  All points are class 0, so
  prediction = 0

- {3,3,3}
  All points are class 0, so
  prediction = 0

Summary

{1,1,1} → 1
{1,1,2} → 1
{1,1,3} → 1
{1,2,2}* → 0
{1,2,3} → 1
{1,3,3} → 1
{2,2,2} → 0
{2,2,3} → 0
{2,3,3} → 0
{3,3,3} → 0

{1,2,2} is a special case.
Where the model can predict class 0 or class 1

Class 1 → 5 votes (1 6 votes if {1,2,2} → 1)
Class 0 → 5 votes (4 votes if {1,2,2} → 1)

Final Prediction ⇒ class 1 (Break ties in favour of class 1 even if the vote got tied)

5. You are given the following dataset containing short text sequences and their associated labels:

Text	Label
I loved the movie, it was fantastic!	Positive
The **film** was boring and too long.	Negative
What a great performance by the actors.	Positive
The storyline was weak and predictable.	Negative

Task: Explain and demonstrate how this text data can be processed and classified into the correct sentiment class (Positive or Negative). Your answer should cover the following parts:

(a) **Preprocessing**

i. Tokenize and lowercase the sentence:
"I Loved the movie, it was fantastic!"
Show the processed output.

(b) **Feature Representation**

i. Explain the difference between:
- Bag-of-Words (BoW) representation
- TF–IDF (Term Frequency–Inverse Document Frequency) representation

ii. For the word *"movie"*, calculate its TF–IDF value. **Use the tokenized version of the first sentence** (*"I loved the movie, it was fantastic!"*) **to compute TF, and use the entire dataset above to compute IDF.** Show the formula and calculation steps.

(c) **Model Building** Use the Naïve Bayes classifier and binary TF (TF=1 if the term exists in the document and TF=0 if it does not) to classify the sentence "What a great movie." Use histograms for your density estimates.

(a) Lowercase:
→ "i loved the movie, it was fantastic!"

↳ Tokenize (space as delimiter)
["i", "loved", "the", "movie,", "it", "was", "fantastic!"]

↳ Tokenize: (including punctuation)
["i", "loved", "the", "movie", ",", "it", "was", "fantastic", "!"]

↳ Tokenize after removing punctuation:

["i", "loved", "the", "movie", "it", "was", "fantastic"]

(b) (i)

Bag-of-Words (BoW)	TF-IDF
Represents how many times each word appears in a document. It is unordered and disregards grammar.	Weights words by their importance (frequent in one doc but rare in other docs get higher weight)
$TF(t,d)$ = count of term t in doc d	$TF\text{-}IDF(t,d) = TF(t,d) \times \log \frac{N}{DF(t)}$
Stop words dominate all docs equally	Weight of the stop words are reduced

(ii) $TF(\text{movie}) = \frac{1}{7}$

[If we include punctuation terms $TF(\text{movie}) = 1/9$]

$$IDF(\text{movie}) = \log \frac{N}{DF(\text{movie})} = \log \frac{4}{1}$$

$$= \log(4) \approx 1.386$$

Case 1

$$TF\text{-}IDF = \frac{1}{7} \times \log(4) = \frac{1}{7} \times 1.386 =$$

0.198

Case 2

If we include punctuation terms. $TF\text{-}IDF = \frac{1}{9} \times \log(4) =$ **0.154**

Case 3

But if we tokenize using space as delimiter. Then TF(movie) = 0 which makes **TF-IDF = 0**. Because "movie" is not a token and "movie," is a token.

(c) Sentence – "What a great movie"

Use binary TF (TF = 1 if the word appears, else 0)

⇒ Prior probability

P(positive) = 2/4 = 0.5 P(negative) = 2/4 = 0.5

Vocabulary size = 21 (unique tokens from 4 sentences)

⇒ Presence of test words in each class

Word	Present in positive docs?	Present in negative docs?
what	Yes (doc 3)	No
a	Yes (doc 3)	No
great	Yes (doc 3)	No
movie	Yes (doc 1)	No

So, $N_{w,pos} = 1$

$N_{w,neg} = 1$

$N_{pos} = N_{neg} = 2$

⇒ Applying Laplace smoothing as negative docs contains zero occurrence for the given words

For binary TF:

$$F(w|pos) = \frac{N_{w,pos} + 1}{N_{pos} + 2} \qquad F(w|neg) = \frac{N_{w,neg} + 1}{N_{neg} + 2}$$

Word	P(w|pos)	P(w|neg)
What	(1+1)/(2+2) = 0.5	(0+1)/(2+2) = 0.25
a	0.5	0.25
great	0.5	0.25
movie	0.5	0.25

I used Laplace Smoothing by adding 1 to every doc occurrence count and adjusted the denominator accordingly.

So, if we haven't done smoothing. Then negative class gets 0 because it never contained any of the word for the given sentence and model would've predicted +ve class by default

$\Rightarrow$ Applying Naive-Bayes formula

Posterior $\approx$ Prior $\times$ likelihood

$$P(Pos|x) \approx P(Pos) \times P(what|pos) \times P(a|pos) \times P(great|pos) \times P(movie|pos)$$

$$\approx 0.5 \times (0.5)^4$$

$$P(Neg|x) \approx P(Neg) \times P(what|neg) \times P(a|neg) \times P(great|neg) \times P(movie|neg)$$

$$\approx 0.5 \times (0.25)^4$$

As $(0.5)^4 > (0.25)^4$

Predicted class = Positive.

6. A researcher studies the relationship between weekly exercise hours (X) and stress level score (Y) in graduate students. The sample size is $n = 26$ and the sample Pearson correlation is $r = -0.39$.

   (a) Test at significance level $\alpha = 0.05$ the null hypothesis $H_0 : \beta_1 = 0$ versus the two-sided alternative $H_1 : \beta_1 \neq 0$. Show the test statistic, decision rule, and conclusion in context.

   (b) Report and interpret the coefficient of determination R^2.

   (c) Briefly explain what the negative sign of r indicates in this scenario.

   Note: Everything needed to solve this question is contained in the exam; do not consult external tables.

Given:

$n = 26$

Sample correlation $r = -0.39$

$H_0 : \beta_1 = 0$

$H_1 : \beta_1 \neq 0$ at $\alpha = 0.05$

(a) To test if slope $\beta_1 = 0$, we can test if the correlation $r = 0$

$$t = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}} = \frac{-0.39\sqrt{26-2}}{\sqrt{1-(-0.39)^2}}$$

$$t = -0.39 \times \sqrt{\frac{24}{1-0.1521}} = -0.39 \times \sqrt{28.31}$$

$$t = -0.39 \times 5.321 = -2.076$$

Since $F = t^2$

$$F = (-2.076)^2 = 4.29$$

degrees of freedom :

Numerator $df_1 = p = 1$

Denominator $df_2 = (n-p-1) = n-2 = 24$

From the given F-table (The given F-Distribution table is for $\alpha = 0.01$ in the right tail but in the question $\alpha = 0.05$. But anyway getting $F_{0.01,(1,24)}$ value from the given table).

$$F_{0.01,(1,24)} = 7.8229$$

Since, $4.29 < 7.8229$

$\rightarrow$ Fail to reject Ho

At 1% significance level, calculated F-value (4.29) is less than critical value (7.8229). Hence, we fail to reject Ho.

(b) $R^2 = r^2 = (-0.39)^2 = 0.1521$

From this, we can interpret that about 15% of the variation in students' stress level scores can be reasoned by their weekly exercise hours. The remaining 85% of the variation is due to other factors that are not included in this model.

(c) The negative sign of r indicates an inverse relationship between exercise and stress. This means increase in exercise hours tend to decrease stress level among students.

7. You are given a binary dataset with two real-valued features (x, y) and a label in $\{0, 1\}$. Class 0 points are denoted by $\circ$ and class 1 points by $\triangle$.

Important rules for *all* parts.

- A point *may not* be its own neighbor.
- Break ties in neighbor votes in favor of class 0.
- Show work to justify your neighbor choices (distances and votes) whenever asked.

Dataset

idx	x	y	label
1	1.0	6.0	0
2	2.2	7.0	0
3	3.1	8.2	0
4	4.0	6.1	0
5	5.2	6.0	1
6	6.0	6.2	0
7	6.2	4.8	1
8	7.0	3.8	1
9	8.2	4.6	1
10	8.8	6.0	0
11	3.2	5.4	1
12	2.8	6.2	0
13	7.6	7.6	1
14	4.8	7.4	1

Distance metrics. For $p = (x_1, y_1)$ and $q = (x_2, y_2)$:

$$d_1(p, q) = |x_1 - x_2| + |y_1 - y_2| \quad \text{(Manhattan)},$$
$$d_2(p, q) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad \text{(Euclidean)},$$
$$d_\infty(p, q) = \max\{|x_1 - x_2|, |y_1 - y_2|\} \quad \text{(Chebyshev)}.$$

Tasks.

(a) **LOOCV of 1-NN across metrics.**
Consider 1-nearest-neighbor classification on (x, y). The distance function will be selected from the options provided. For each option, calculate the LOOCV error of 1-NN and indicate which distance gives superior performance. The options are: Manhattan d_1, Euclidean d_2, and Chebyshev d_∞.

(b) **Effect of k.**
For each metric d_1, d_2, and d_∞, compute the LOOCV error for $k \in \{1, 3, 5\}$. Report the error for each (metric, k) and list the misclassified indices. For the *best LOOCV setting*, show the full 5-NN calculation (neighbor IDs, labels, distances, vote) for *every* misclassified point.

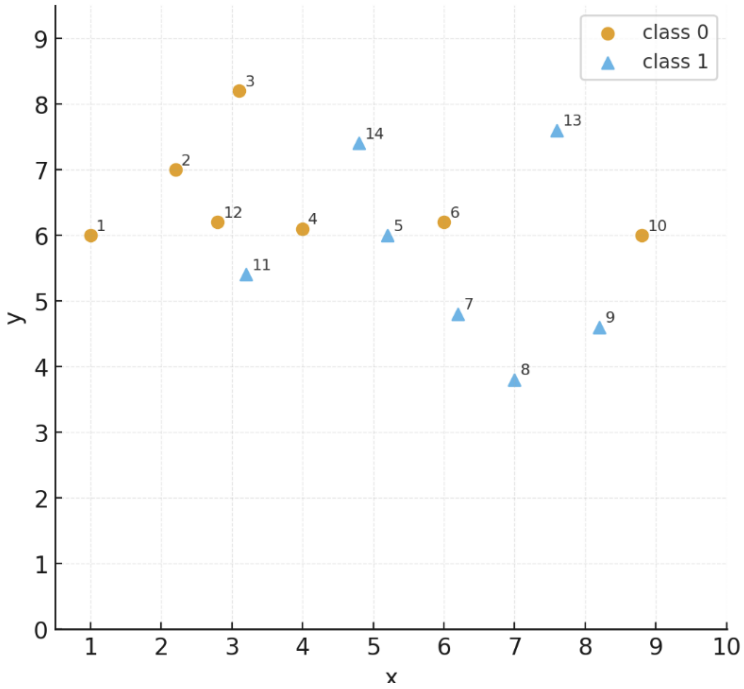


Figure 1: Dataset with indices and class markers. Use this for geometric intuition; grading is based on your calculations.

(c) **Fixed 5-fold CV (Euclidean).**
Using Euclidean d_2 and $k \in \{1, 3, 5\}$, evaluate the fixed folds (do not reshuffle):

$$\text{Fold } 1 = \{1, 7, 11\}, \quad \text{Fold } 2 = \{2, 8, 12\}, \quad \text{Fold } 3 = \{3, 9, 5\}, \quad \text{Fold } 4 = \{4, 6, 10\},$$

$$\text{Fold } 5 = \{13, 14\}.$$

For each k, report the error on each fold, the mean error across folds, and the misclassified index set per fold. Select the k recommended by this split and justify briefly.

(d) **1-NN decision boundary (Euclidean).**
Sketch the qualitative 1-NN (k=1) decision regions for d_2. Indicate at least two places where the boundary is clearly non-linear due to local class interleaving It is fine if you use software to plot it and you do not need to submit the code.

(a) For each point i:

→ Find the closest point(s) by the selected distance metric.

→ If multiple closest, vote and use tie rule if required

① Manhattan Distance ($d_1(p,q) = |x_1 - x_2| + |y_1 - y_2|$)

i	nearest	Predict label	True label	correct?
1	12	0	0	✓
2	12	0	0	✓
3	2	0	0	✓
4	(5,12)	0 (tie-favour 0)	0	✓
5	6	0	1	×
6	5	1	0	×
7	6	0	1	×
8	7	1	1	✓
9	(8,10)	0 (tie-favour 0)	1	×
10	9	1	0	×
11	12	0	1	×
12	11	1	0	×
13	10	0	1	×
14	5	1	1	✓

Wrongly classified points → 5, 6, 7, 9, 10, 11, 12, 13 (8)

Correctly classified points → 1, 2, 3, 4, 8, 14 (6)

LOOCV error → 8/14

≈ 57.14 %

② Euclidean distance $(d_2(p,q) = \sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$

i	nearest	Predict label	True label	correct?
1	2	0	0	✓
2	12	0	0	✓
3	2	0	0	✓
4	11	1	0	×
5	6	0	1	×
6	5	1	0	×
7	8	1	1	✓
8	7	1	1	✓
9	8	1	0	×
10	9	1	1	×
11	12	0	0	×
12	11	1	1	×
13	10	0	1	✓
14	5	1	1	

Wrong = 4, 5, 6, 10, 11, 12, 13

$= \frac{7}{14}$

LOOCV error using euclidean $= \frac{7}{14} \approx 50\%$

③ chebyshev $(d_\infty(p,q) = \max(|x_1 - x_2|, |y_1 - y_2|)$

i	nearest	Predict label	True label	correct?
1	2	0	0	✓
2	12	0	0	✓
3	2	0	0	✓
4	11	1	0	✗
5	6	0	1	✗
6	5	1	0	✗
7	8	1	1	✓
8	7	1	1	✓
9	8	1	1	✓
10	9	1	0	✗
11	(4, 12)	0 (both 0)	1	✗
12	(2, 11)	0 (tie-favour 0)	0	✓
13	(6, 10)	0	1	✗
14	6	0	1	✗

Wrong - 4, 5, 6, 10, 11, 13, 14

LOOCV error using chebyshev

$= \frac{7}{14} \approx 50\%$

Best performance - Euclidean and chebyshev (LOOCV error = 0.5)

Worst performance - Manhattan (LOOCV error = 0.5714)

(b) ① Manhattan Distance

idx	nearest_idx	labels
1	[12, 2, 11, 4, 5]	→ [0, 0, 1, 0, 1]
2	[12, 3, 1, 11, 4]	→ [0, 0, 0, 1, 0]
3	[2, 12, 14, 11, 4]	→ [0, 0, 1, 1, 0]
4	[5, 12, 11, 6, 14]	→ [1, 0, 1, 0, 1]
5	[6, 4, 14, 7, 11]	→ [0, 0, 1, 1, 1]
6	[5, 7, 4, 14, 13]	→ [1, 1, 0, 1, 1]
7	[6, 8, 9, 5, 4]	→ [0, 1, 1, 1, 0]
8	[7, 9, 6, 5, 10]	→ [1, 1, 0, 1, 0]
9	[8, 10, 7, 13, 6]	→ [1, 0, 1, 1, 0]
10	[9, 13, 6, 5, 7]	→ [1, 1, 0, 1, 1]
11	[12, 4, 2, 5, 1]	→ [0, 0, 0, 1, 0]

12 [11, 4, 2, 1, 3] ⟶ [1, 0, 0, 0, 0]

13 [10, 6, 14, 9, 5] ⟶ [0, 0, 1, 1, 1]

14 [5, 4, 6, 3, 13] ⟶ [1, 0, 0, 0, 1]

idx	distances	3NN label	5NN label	True label
1	[2, 2.2, 2.8, 3.1, 4.2]	0	0	0
2	[1.4, 2.1, 2.2, 2.6, 2.7]	0	0	0
3	[2.1, 2.3, 2.5, 2.9, 3]	0	0	0
4	[1.3, 1.3, 1.5, 2.1, 2.1]	1	1	0
5	[1, 1.3, 1.8, 2.2, 2.6]	0	1	1
6	[1, 1.6, 2.1, 2.4, 3]	1	1	0
7	[1.6, 1.8, 2.2, 2.2, 3.5]	1	1	1
8	[1.8, 2, 3.4, 4, 4]	1	1	1
9	[2, 2, 2.2, 3.6, 3.8]	1	1	1
10	[2, 2.8, 3, 3.6, 3.8]	1	1	0
11	[1.2, 1.5, 2.6, 2.6, 2.8]	0	0	1
12	[1.2, 1.3, 1.4, 2, 2.3]	0	0	0
13	[2.8, 3, 3, 3.6, 4]	0	1	1
14	[1.8, 2.1, 2.4, 2.5, 3]	0	0	1

k	Misclassified Indices	# Errors	LOOCV Error
1	[5, 6, 7, 9, 10, 11, 12, 13]	8	0.5714
3	[4, 5, 6, 10, 11, 13, 14]	7	0.5
5	[4, 6, 10, 11, 14]	5	0.35

Best LOOCV : $k = 5$, error = 35.7%

② Euclidean Distance

idx	nearest_idx	distances
1	[2, 12, 11, 4, 3]	[1.6, 1.8, 2.3, 3, 3.05]
2	[12, 3, 1, 11, 4]	[1, 1.5, 1.6, 1.9, 2]
3	[2, 14, 12, 4, 11]	[1.5, 1.9, 2, 2.3, 2.8]
4	[11, 5, 12, 14, 6]	[1, 1.2, 1.2, 1.5, 2]
5	[6, 4, 14, 7, 11]	[0.8, 1.2, 1.4, 1.6, 2]
6	[5, 7, 14, 4, 13]	[0.8, 1.4, 1.7, 2, 2.1]
7	[8, 6, 5, 9, 4]	[1.3, 1.4, 1.6, 2, 2.6]
8	[7, 9, 6, 5, 10]	[1.3, 1.4, 2.6, 2.8, 2.8]
9	[8, 10, 7, 6, 13]	[1.4, 1.5, 2, 2.1, 3]

10	[9, 13, 6, 8, 7]	[1.5, 2, 2.8, 2.8, 2.9]
11	[12, 4, 2, 5, 1]	[0.9, 1, 1.9, 2, 2.3]
12	[11, 2, 4, 1, 3]	[0.9, 1, 1.2, 1.8, 2]
13	[10, 6, 14, 5, 9]	[2, 2.1, 2.8, 2.9, 3]
14	[5, 4, 6, 3, 12]	[1.4, 1.5, 1.7, 1.9, 2.3]

idx	labels	3NN label	5NN label	True label
1	[0, 0, 1, 0, 0]	0	0	0
2	[0, 0, 0, 1, 0]	0	0	0
3	[0, 1, 0, 0, 1]	0	0	0
4	[1, 1, 0, 1, 0]	1	1	1
5	[0, 0, 1, 1, 1]	0	1	0
6	[1, 1, 1, 0, 1]	1	1	1
7	[1, 0, 1, 1, 0]	1	1	1
8	[1, 1, 0, 1, 0]	1	1	1
9	[1, 0, 1, 0, 1]	1	1	0
10	[1, 1, 0, 1, 1]	1	1	1
11	[0, 0, 0, 1, 0]	0	0	0
12	[1, 0, 0, 0, 0]	0	0	1
13	[0, 0, 1, 1, 1]	0	1	1
14	[1, 0, 0, 0, 0]	0	0	1

k	Misclassified Indices	# Errors	LOOCV error
1	[4,5,6,10,11,12,13]	7	0.5
3	[4,5,6,10,11,13,14]	7	0.5
5	[4,6,10,11,14]	5	0.3571

Best LOOCV : k=5, error = 35.71%

③ chebyshev distance

idx	nearest_idx	distances
1	[2,12,3,11,4]	[1.2, 1.8, 2.2, 2.2, 3.0]
2	[12,3,1,11,4]	[0.8, 1.2, 1.2, 1.6, 1.8]
3	[2,14,12,4,1,5]	[1.2, 1.7, 2, 2.1, 2.2, 2.2]
4	[11,5,12,4,2]	[0.8, 1.2, 1.2, 1.3, 1.8]
5	[6,4,7,14,11]	[0.8, 1.2, 1.2, 1.4, 2]
6	[5,14,7,13,4]	[0.8, 1.2, 1.4, 1.6, 2]
7	[8,5,6,9,4]	[1, 1.2, 1.4, 2, 2.2]
8	[7,9,5,10,6]	[1, 1.2, 2.2, 2.2, 2.4]
9	[8,10,7,6,5]	[1.2, 1.4, 2, 2.2, 3]
10	[9,13,8,7,6]	[1.4, 1.6, 2.2, 2.6, 2.8]

11	[4, 12, 2, 5, 14]	[0.8, 0.8, 1.6, 2, 2]
12	[2, 11, 4, 1, 3]	[0.8, 0.8, 1.2, 1.8, 2]
13	[6, 10, 5, 7, 14]	[1.6, 1.6, 2.4, 2.8, 2.8]
14	[6, 4, 5, 3, 11, 12]	[1.2, 1.3, 1.4, 1.7, 2, 2]

idx	labels	3NN label	5NN label	True label
1	[0, 0, 0, 1, 0]	0	0	0
2	[0, 0, 0, 1, 0]	0	0	0
3	[0, 1, 0, 0, 0, 1]	0	1	0
4	[1, 1, 0, 1, 0]	1	1	1
5	[0, 0, 1, 1, 1]	0	1	0
6	[1, 1, 1, 1, 0]	1	1	1
7	[1, 1, 0, 1, 0]	1	1	1
8	[1, 1, 1, 0, 0]	1	1	1
9	[1, 0, 1, 0, 1]	1	1	0
10	[1, 1, 1, 1, 0]	1	1	0
11	[0, 0, 0, 1, 1]	0	0	1
12	[0, 1, 0, 0, 0]	0	0	0
13	[0, 0, 1, 1, 1]	0	1	1
14	[0, 0, 1, 0, 1, 0]	0	0	1

K	Misclassified idx	Errors	LOOCV error
1	[4, 5, 6, 10, 11, 13, 14]	7	0.5
3	[4, 5, 6, 10, 11, 13, 14]	7	0.5
5	[4, 6, 10, 11, 14]	5	0.357

Best LOOCV: k=5, error = 35.7%

c) Fold 1 = {1, 7, 11}
Fold 2 = {2, 8, 12}
Fold 3 = {3, 9, 5}
Fold 4 = {4, 6, 10}
Fold 5 = {13, 14}

Misclassified points (Euclidean):

1NN : {4, 5, 6, 10, 11, 12, 13}
3NN : {4, 5, 6, 10, 11, 13, 14}
5NN : {4, 6, 10, 11, 14}

$\Rightarrow$ $\underline{K=1}$

Fold	Misclassified points	Fold error
Fold 1	{11}	$1/3 = 0.333$
Fold 2	{12}	$1/3 = 0.333$
Fold 3	{5}	$1/3 = 0.333$
Fold 4	{4, 6, 10}	$3/3 = 1$
Fold 5	{13}	$1/2 = 0.5$

$$\text{Mean error} = \frac{(0.333 + 0.333 + 0.333 + 1 + 0.5)}{5}$$

$$\approx 0.5 \text{ //}$$

$\Rightarrow$ $\underline{K=3}$

Fold	Misclassified points	Fold error
Fold 1	{11}	$1/3 = 0.333$
Fold 2	{}	0

Fold 3	{5}	1/3 = 0.333
Fold 4	{4, 6, 10}	3/3 = 1
Fold 5	{13, 14}	2/2 = 1

$$\text{Mean error} = \frac{(0.333 + 0.333 + 1 + 1)}{5}$$

$$= \frac{2.666}{5} \approx 0.533 //$$

$\Rightarrow$ $\underline{K = 5}$

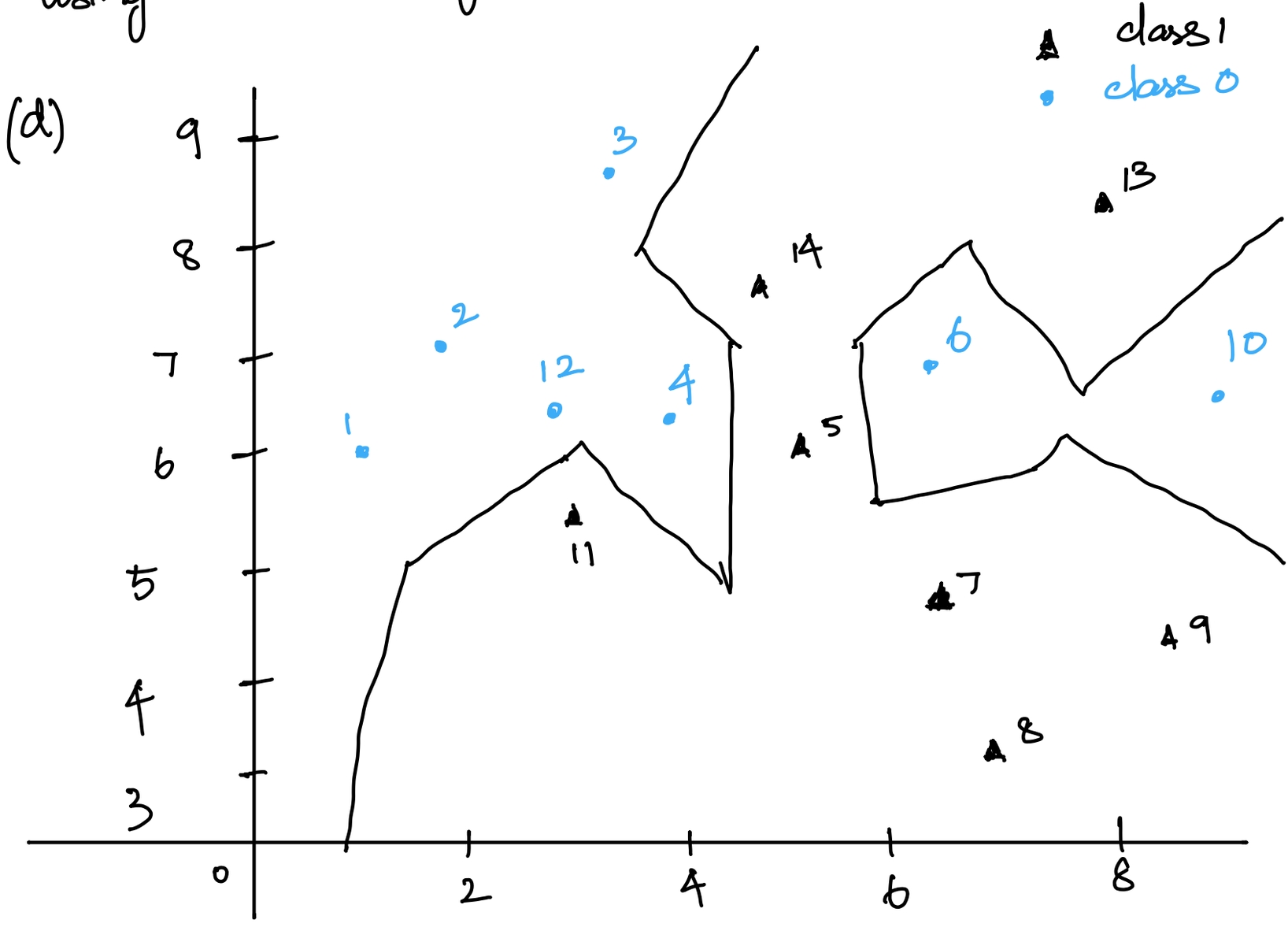
Fold	Misclassified points	Fold error
Fold 1	{11}	1/3 = 0.333
Fold 2	{ }	0
Fold 3	{ }	0
Fold 4	{4, 6, 10}	3/3 = 1
Fold 5	{14}	1/2 = 0.5

Mean error = $\frac{(0.333 + 1 + 0.5)}{5}$

$= \frac{1.833}{5} \approx 0.367$ //

$k=5$ has the lowest mean CV error (≈ 0.367). Hence $k=5$ is recommended as it smooths the decision boundary by using more neighbors to vote.

(d)



The figure above shows the Euclidean-1NN decision regions. The result is varonai style partition.

The boundary is clearly non-linear. It bends sharply anywhere points from different classes are interleaved in space.

→ Around points 4, 5, 6, 11, 12, 14 (roughly $x \approx 3-6$, $y \approx 5-7$), where class 0 and class 1 points sit very close together.
Some of these points actually get classified as the other class (for example, point 4 is true label 0 but predicted 1, point 11 is true label 1 but predicted 0)

→ Around points 7, 8, 9 and 10 (roughly $x \approx 6-9$, $y \approx 4-6$) where mostly a class-1 region surrounds a class-0 point.

8. **Programming Question: Predicting Housing Prices with Linear Regression**

Dataset: `student-mat.csv`

Link: https://archive.ics.uci.edu/dataset/320/student+performance

In this problem, you will build a linear regression model to predict housing prices in the USA based on various features. You will use Python and `scikit-learn` for this task.

(a) **Data Exploration and Pre-processing**

i. Import the `student-mat.csv` dataset as a pandas DataFrame.

ii. Select the following features: `age, studytime, schoolsup, goout, Dalc, Walc, health, absences, G3` (target/dependent variable). Encode the binary variable (`schoolsup`) values as 0s (no) and 1s (yes). Combine `Dalc` and `Walc` into one variable `alc` by taking averge, then remove `Dalc` and `Walc`. Display the first five rows of the pre-processed dataset.

iii. Find the number of outliers for each independent variable using the IQR method.

iv. Standardize and run PCA on the dataset. Create a scatterplot of PC1 vs PC2, coloring the dots by their final grade `G3`. Inspect the component loadings and determine which features contribute the most to PC1 and PC2. *Keep and use standardized data for remaining problems.*

(b) **Linear Regression**

i. Split data into training set and testing set with an 80:20 ratio. Use random seed 552 for reproducibility.

ii. Build three models using the training set: A. Linear Regression Model, B. Linear Regression Model with Ridge Regularization, and C. Linear Regression Model with Lasso Regularization. Set α=0.1.

iii. Test all three models on the test set. Find out the best performing model with repsect to each of the metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R^2.

iv. How do you interpret R^2 values from the three models?

v. Print coefficients of independent variables from the three models in one table.

vi. What's the relationship between each independent variable and the dependent variable?

vii. How do the regularization methods differ? What can you conclude about the dataset and features given the results?

viii. If the regularization strength (α) is increased, what would happen to performance metrics?

ix. What are some feature engineering methods you would suggest to improve model performance?

Expected Output: Your submission should include:

- Jupyter Notebook `.ipynb` with all the steps clearly commented.
- The output of each step as specified above (e.g., head of DataFrame, info, describe, missing value counts, evaluation metrics for all models, coefficients, and intercept).
- Visualizations for outlier detection and residual analysis.
- A brief discussion answering the interpretation questions.

Scratch paper

Name:

USC ID:

Scratch paper

Name:

USC ID:

F - Distribution ($\alpha = 0.01$ in the Right Tail)

Denominator Degrees of Freedom df_2 \ Numerator Degrees of Freedom df_1	1	2	3	4	5	6	7	8	9
1	4052.2	4999.5	5403.4	5624.6	5763.6	5859.0	5928.4	5981.1	6022.5
2	98.503	99.000	99.166	99.249	99.299	99.333	99.356	99.374	99.388
3	34.116	30.817	29.457	28.710	28.237	27.911	27.672	27.489	27.345
4	21.198	18.000	16.694	15.977	15.522	15.207	14.976	14.799	14.659
5	16.258	13.274	12.060	11.392	10.967	10.672	10.456	10.289	10.158
6	13.745	10.925	9.7795	9.1483	8.7459	8.4661	8.2600	8.1017	7.9761
7	12.246	9.5466	8.4513	7.8466	7.4604	7.1914	6.9928	6.8400	6.7188
8	11.259	8.6491	7.5910	7.0061	6.6318	6.3707	6.1776	6.0289	5.9106
9	10.561	8.0215	6.9919	6.4221	6.0569	5.8018	5.6129	5.4671	5.3511
10	10.044	7.5594	6.5523	5.9943	5.6363	5.3858	5.2001	5.0567	4.9424
11	9.6460	7.2057	6.2167	5.6683	5.3160	5.0692	4.8861	4.7445	4.6315
12	9.3302	6.9266	5.9525	5.4120	5.0643	4.8206	4.6395	4.4994	4.3875
13	9.0738	6.7010	5.7394	5.2053	4.8616	4.6204	4.4410	4.3021	4.1911
14	8.8616	6.5149	5.5639	5.0354	4.6950	4.4558	4.2779	4.1399	4.0297
15	8.6831	6.3589	5.4170	4.8932	4.5556	4.3183	4.1415	4.0045	3.8948
16	8.5310	6.2262	5.2922	4.7726	4.4374	4.2016	4.0259	3.8896	3.7804
17	8.3997	6.1121	5.1850	4.6690	4.3359	4.1015	3.9267	3.7910	3.6822
18	8.2854	6.0129	5.0919	4.5790	4.2479	4.0146	3.8406	3.7054	3.5971
19	8.1849	5.9259	5.0103	4.5003	4.1708	3.9386	3.7653	3.6305	3.5225
20	8.0960	5.8489	4.9382	4.4307	4.1027	3.8714	3.6987	3.5644	3.4567
21	8.0166	5.7804	4.8740	4.3688	4.0421	3.8117	3.6396	3.5056	3.3981
22	7.9454	5.7190	4.8166	4.3134	3.9880	3.7583	3.5867	3.4530	3.3458
23	7.8811	5.6637	4.7649	4.2636	3.9392	3.7102	3.5390	3.4057	3.2986
24	7.8229	5.6136	4.7181	4.2184	3.8951	3.6667	3.4959	3.3629	3.2560
25	7.7698	5.5680	4.6755	4.1774	3.8550	3.6272	3.4568	3.3239	3.2172
26	7.7213	5.5263	4.6366	4.1400	3.8183	3.5911	3.4210	3.2884	3.1818
27	7.6767	5.4881	4.6009	4.1056	3.7848	3.5580	3.3882	3.2558	3.1494
28	7.6356	5.4529	4.5681	4.0740	3.7539	3.5276	3.3581	3.2259	3.1195
29	7.5977	5.4204	4.5378	4.0449	3.7254	3.4995	3.3303	3.1982	3.0920
30	7.5625	5.3903	4.5097	4.0179	3.6990	3.4735	3.3045	3.1726	3.0665
40	7.3141	5.1785	4.3126	3.8283	3.5138	3.2910	3.1238	2.9930	2.8876
60	7.0771	4.9774	4.1259	3.6490	3.3389	3.1187	2.9530	2.8233	2.7185
120	6.8509	4.7865	3.9491	3.4795	3.1735	2.9559	2.7918	2.6629	2.5586
∞	6.6349	4.6052	3.7816	3.3192	3.0173	2.8020	2.6393	2.5113	2.4073