Code NO: R22DS704PC

CMR ENGINEERING COLLEGE :: HYDERABAD
UGC AUTONOMOUS

IV - B.TECH - I-SEM End Examinations (Reg) - December - 2025

Predictive Analytics
(CSD)

[Max. Marks: 60]

Key & Scheme of Evaluation

Part - A [10 Marks]

1. a) Define ridge regression and mention when it is used. [1 M]

Ans: Ridge regression is a regularization technique used in linear regression to address problems like multicolinearity and overfitting. It is also called as L2-regularization. It penalizes the sum of the standard coefficients.

$$\text{Min}(\|y - X\theta\|^2 + \lambda\|\theta\|^2)$$

b) Define Logistic Regression. [1 M]

Ans: Logistic Regression is a supervised learning algorithm used for classification problems. It is a type of classification algorithm that predicts a discrete or categorical outcomes.

c) What is Bias-Variance trade-off? [1 M]

Ans: Bias is the error from overly simple assumptions in a model, leading to underfitting, while variance is the error from excessive sensitivity to training data leading to overfitting. Bias-variance tradeoff is the fundamental challenge of balancing these two sources of error to create a model that generalizes well to new, unseen data.

d) What is the purpose of Bootstrapping? [1 M]

Ans: It is a procedure for estimating the distribution of an estimator by resampling technique that allows you to estimate properties of an estimator. estimator by repeatedly drawing samples from the original data.

e) What is gradient Boosting [1 M]

Ans: Gradient Boosting constructs models in a sequential manner where each week learner minimizes the residual error of the previous one using Gradient descent.

f) Define Ada Boosting [1 M]

Ans Ada boost is a boosting technique that combines several weak classifiers in sequence to build a strong one.

g) Define the concept of Backpropagation. [1M]

Ans: It is a core algorithm in neural networks that works by calculating the error between a network's output and the desired output, then using chain rule to propagate that error backward through the network.

h) Define K-Nearest Neighbor classifier. [1 M]

Ans It is non parametric, supervised machine learning algorithm that categorizes the new data points by finding the "K" closest data points in the training set and assigning the new point to the most common class among its neighbors.

i) Define Unsupervised learning. [1 M]

Ans: It is a type of machine learning algorithms analyze unlabeled data to discover hidden patterns, structures, and relationships without any prior human guidance.

j) List the applications of Association rule Mining. [1M]

Ans:
* Credit card transactions
* tele Com uses for call waiting, call forwarding
* Banking services
* Market Basket Analysis
* Insurance claim frauds
* Medical patient histories.

Part - B

[50 Marks]

2. a) How does LDA differ from logistic regression for classification? [5 M]

Ans LDA assumes features follow a multivariate normal distribution and have equal covariance matrices across classes, while logistic regression is more flexible and does not require these assumptions, making it more robust to violations. ——— 1 M

LDA estimates the joint probability of the data and class, where as logistic regression estimates the conditional probability of the class given the data ——— 2 M.

Key differences

Feature	LDA	Logistic Regression
Assumptions	Assumes features are multivariate Normal ~~distributions~~ and have equal covariance matrices across classes.	makes fewer assumptions; does not require normality or equal covariance matrices.
Approach	Models the joint Probability $P(X, Y)$ or assumes a specific distribution for X and Y	Models the conditional probability $P(Y)$. ——— 3 M
Robustness	less robust to outliers and non normal data	More robust to non-normal data and violations of assumptions ——— 4 M
Performance	performs well when its assumptions are met	Often performs better in practice, especially with non-normal data, due to its flexibility
Decision boundary	Creates a linear decision boundary under the assumption of equal covariance matrices.	Creates a linear boundary, but can be extended to non-linear boundaries through feature engineering ——— 5 [M]

2b) Analyze the effect of regularization parameters in Lasso and Ridge [5 M]

Ans:- Regularization parameter, often denoted as α (alpha), controls the strength of the penality applied to the models coefficients in both Lasso and Ridge regression. — 1 M

A higher α increases the penality, leading to smaller coefficients for Ridge and the possibility of some coefficients becoming exactly zero for Lasso. This effect shrinks coefficients to combat overfitting, but Ridge retains all features while Lasso performs automatic feature selection. — 2 M

Effect of the regularization parameter (α)

* Low α (closer to zero):
  * The penality is weak
  * Coefficients are shrunk less, closer to the original least square values.
  * A model with a very low α can still overfit if its too complex for the data. — 3 M

* High α:
  * The penality is strong
  * Coefficients are shrunk more aggressively towards zero
  * In Lasso, a higher α increases the likelihood of some coefficients being set to exactly zero, effectively removing features from the model. — 4 M
  * In Ridge, coefficients are shrunk, but none are set to zero. exactly zero.
  * If α is too high, model can become too simple, leading to underfitting and loss of predictive power. — 5 M.

3@ What is Lasso regression and Ridge regression? Explain. [5M]

Ans: A regularization technique L1 Regularization in a regression model. It is also called as Lasso (Least Absolute Shrinkage and selection operator) regression. — 1M

It adds absolute value of magnitude of the coefficient as a penalty term to the loss function (L). This penalty can shrink some coefficients to zero which helps in selecting only the important features and ignoring the less important ones. — 2M

$$Cost = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{i=}^{m} |w_i|$$

Where:
- m - no. of features;
- n - no. of examples
- y_i - Actual Target value
- $\hat{y}_i$ - Predicted Target value

Ridge Regression

It is a model that uses the L2 regularization technique is called Ridge regression. It adds the squared magnitude of the coefficient as a penalty term to the loss function (L). It handles multicolinearity by shrinking the coefficients of correlated features instead of eliminating them.

$$Cost = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{i=1}^{m} w_i^2$$ — 4M

Where
- n - No. of examples or datapoints
- m - No. of features i.e; predictor variables
- y_i - Actual target value for the i^{th} example
- $\hat{y}_i$ - Predicted target value for the i^{th} example.
- w_i - coefficients of the features.
- λ - Regularization Parameter that controls the strength of regularization.

— 5 M.

3.b) Derive the cost function in Logistic Regression. [5 M]

Ans Logistic Regression is a supervised learning algorithm used for classification problems. To measure how well the model is predicting or performing. we use a cost function, which tells us how far the predicted values are from the actual ones. In Logistic Regression, the cost function is based on log loss (cross-entropy loss) instead of mean squared ~~effo~~ error.

— 2 M

* It measures the error between the predicted probability and the actual class label (0 or 1)
* Instead of a straight line (like LinearRegression). Logistic regression works with probabilities between 0 and 1 using sigmoid function.
* The cost penalizes wrong predictions more heavily when the model is confident but wrong.
* The cost function is defined as

$$cost(h_\theta(x), y) = -y \cdot \log(h_\theta(x)) - (1-y) \cdot \log(1 - h_\theta(x))$$ — 4 M

Where:

* $h_\theta(x)$: Predicted probability using sigmoid
* y: Actual value (0 or 1)

For all training examples, the cost function (Log loss) becomes:

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \left[y^{(i)} \log(h_\theta(x^{(i)})) + (1-y^{(i)}) \log(1 - h_\theta(x^{(i)})) \right]$$

— 5 M.

(11) Demonstrate K-fold cross validation with an example. [10]

Ans: K-fold cross validation is a technique that divides a dataset into K-subsets or K-folds, to reliably evaluate a machine learning models performance. In each iteration, one fold is used as a test-set while the remaining K-1 folds are used for training, a process that is repeated K-times with each fold acting as the test set once. — 2M

The models final performance is then calculated by averaging the results from all K iterations. This method helps estimate how well the model will generalize to unseen data by training and testing it on different data partitions. — 2M

How it works:-

Divide the data into K equal sized folds A common value for K is 10.

* Iterate K-times this process repeats K times.
* In each iteration, one fold is designated as the validation set.
* The model is trained on the remaining K-folds.
* The models performance is evaluated on the validation set.

— 6M

Calculating the final score:

After all K-iterations the performance matrices from each fold are averaged to get a final, more robust estimate of the models performance.

— 8M

purpose of K-fold or When use it:

* Reliable performance estimation
* Model comparision
* Generalization.

Considerations

* Choosing K.
* Data Leakage
* Stratified K-fold.

10M

5.A) Derive the relationship between bias, variance and MSE. [5M]

Ans The Relationship between bias, variance and MSE is $MSE = Bias^2 + Variance$.

Here

MSE (Mean squared error) is the sum of the squared bias and the variance of an estimator. This means an estimators avg error is composed of two parts. How far off the avg prediction is from the true value (bias) and how spread out the predictions are from that average (variance). — 2M

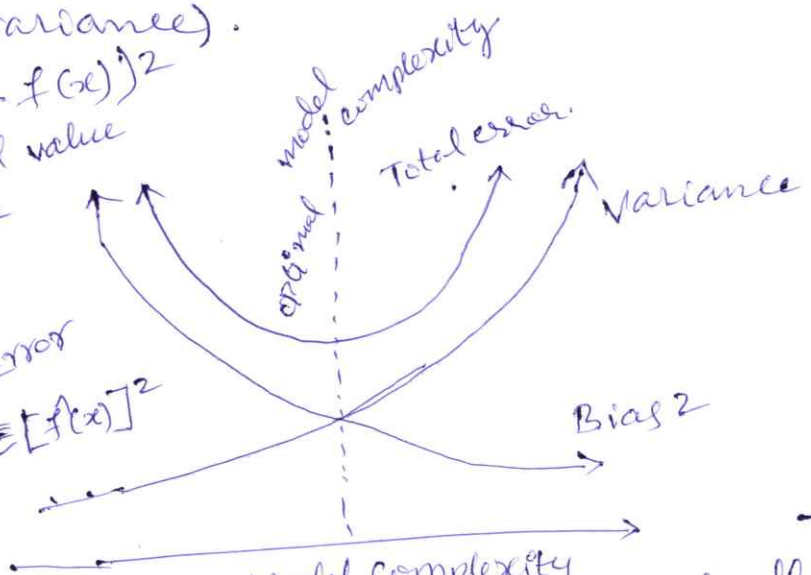
$Bias^2 = (E[\hat{f}(x)] - f(x))^2$

$\hat{f}(x)$: predicted value

$f(x)$: true value

$E(\hat{f}(x))$ expected prediction over different testing sets.

$Variance = E[(\hat{f}(x) - E[\hat{f}(x)])^2$



— 4M

Model type	Bias	Variance	Result
underfitting	High	Low	poor training & test performance
optimal	Moderate	Moderate	Best generalization
overfitting	Low	High	poor test-performance

— 5M.

5 B. How can effective no. of parameters be used in model selection. [5 M]

Ans:- The effective no. of parameters is used in model selection to quantify the trade-off between a model's complexity and its goodness of fit, guiding the choice towards the model that balances simplicity with accurate prediction. — 1 M

It measures the actual degrees of freedom a model uses, which can be less than the nominal no. of parameters due to regularization or other factors like non-linearity. — 2 M

By comparing models based on this effective numbers, techniques like Akaike Information Criterion (AIC) or Bayesian Information Criterion can help selecting a more parsimonious model that generalizes better to new data and avoids overfitting. — 3 M.

How it works

* Balancing fit & complexity
* Quantifying complexity.
* Model selection Criteria
  - AIC
  - BIC.

— 4 M

Applications: This concept is used for tasks like

* choosing the order of a polinomial.
* selecting the no. of features or clusters.
* comparing different models, with non linearities

— 5 M

6.A) Discuss bagging and boosting techniques. [5 M]

Ans Bagging & Boosting are two types of Ensemble learning. These two decrease the variance of a single estimate as they combine several estimates from different models. — 1 M

Bagging is a homogeneous weak learners model that learns from each other independently in parallel and combines them for determining the model average.

Boosting also homogeneous weak learners model but works ~~effectively~~ differently from bagging. — 2 M.

In this model, learners learn sequentially and adaptively to improve model predictions of a learning algorithm.

Bagging - Bootstrapping Aggregating also known as Bagging. Example of Bagging: Random forests for applications — 4 M.

Differences and similarities — [5 M]

6(B). Illustrate the use of additive models of nonlinear relationships. [5M]

Ans: Additive models illustrate non linear relationships by extending linear models to include a sum of non-linear functions, each representing the relationship of one predictor variable to the outcome.

Instead of a single line, these models use flexible curves or "smooths" of the predictor variables, rather than just a linear combination. — 1M

Instead of a single line, relationship is represented by a flexible curve for each predictor, which is built using techniques like splines. This approach provides a balance between the flexibility of complex models and the interpretability of linear models. — 2M.

How it works

* Extending linear models: An additive model is like a linear model ($Y = b_0 + b_1X_1 + b_2X_2$) but replaces the linear terms with smooth, non-linear functions.
* Non linear functions: The model is expressed as:

$$Y = b_0 + f_1(X_1) + f_2(X_2) + \cdots + c.$$

* Smooth functions (f_i): Each function (f_i) captures the non linear relationship between its corresponding predictor X_i and outcome Y.
* Splines: These functions are often built using splines, which are flexible curves that can take on various shapes to fit the data structures. — 4 M

Benefits & limitations:

* Interpretability
* Flexibility
* Balance
* Additive assumption
* Complexity

— 5 M

(7) Compare additive models, regression trees, and boosting in predicting New Zealand fish size data [10 M]

Ans: For predicting New Zealand fish size data, boosting generally offers superior predictive performance compared to generalized additive models (GAMs) and single regression trees, particularly when dealing with complex ecological data involving non-linear relationships and interactions.

— 2 [2 M]

Model Comparision

Feature	Additive Models (GAMs)	Regression Trees	Boosting (BRTs)
Predictive performance	Generally good, but often outperformed by ensemble methods like boosting. can overfit if not carefully managed	can have limited predictive power & are prone to overfitting with noisy data or a lack of strong patterns. — 3 M	High Predictive accuracy: often the best among the 3. Avoids overfitting more flexible, effectively than GAMs or single trees.
Model Complexity / Interpretability	High interpretability: Clearly shows the smooth, non-linear relationships of each predictor variable with the response.	Smooth & highly interpretable. — 4 M	A "black-box" approach: models are complex ensembles of many trees, making
Handling Non-Linearity	Handles non-linear relationships well using smooth functions	Excels at modeling non linear dependencies and interaction through recursive binary splits. — 5 M	Combines many simple trees to effectively model complex non-linear responses and interactions.

Data Requirements	Requires careful handling & interpretation terms.	less data preparation needed; can handle mixed datatypes.	Requires careful tuning of parameters and is computationally intensive.
Application in Fisheries	widely used for species distribution & catch standardization studies	used but less frequently than GAMs or BRTs due to performance limitations.	Increasingly used in ecology and fisheries for its strong performance in modeling complex environmental data.

— 6 M

— 7 M

Conclusion for Newzealand Fish Size Data

For the specific task of predicting Newzealand fish size data, which likely involves complex, non-linear relationships between fish size and various environmental factors (e.g., water temparature, depth, food availability, location), boosting is the strongest choice. — 8 M

Research in similar ecological contexts has shown that Boosting Regression Trees (BRTs) provide substantially superior predictive performance and better handle complex interactions compared to generalized additive models, and they inherently outperform single regression trees. While BRTs are less interpretable than the other methods, their accuracy makes them a powerful tool for robust prediction in fisheries management. — 9 M — 10 M

(8) Illustrate the working of K-Nearest Neighbor with a numerical example.

A: KNN is a supervised learning method that classifies a new data point based on the majority class of its 'k' nearest neighbors in the training data.

1M.

How KNN works: A numerical example.

Let's use a simple dataset to predict if a new fruit is an Apple (class A) or a Banana (class B), based on its "weight" & "diameter".

Fruit	weight (gm)	Diameter (cm)	class
F1	70	9	A
F2	90	10	A
F3	100	8	A
F4	110	11	B
F5	120	9	B
F6	130	10	B

New data point (P): we want to classify a new fruit with weight = 95 gm and — 2M
Diameter = 9cm.

Choose k: Let's set k=3 (we will look at the 3 nearest neighbors).

Step1: calculate the distance
We use Euclidean distance formula to find the distance between the new point P (95, 9) and all other points in the dataset.

$$\text{Distance}(P, F_i) = \sqrt{(\text{weight}_P - \text{Weight}_{F_i})^2 + (\text{Diameter}_P - \text{Diameter}_{F_i})^2}$$

— 2M

Fruit	coordinates	calculation	Distance
F1	(70, 9)	$\sqrt{(95-70)^2+(9-9)^2} = \sqrt{25^2+0^2}$	25
F2	(90, 10)	$\sqrt{(95-90)^2+(9-10)^2} = \sqrt{5^2+(-1)^2}$	5.09
F3	(100, 8)	$\sqrt{(95-100)^2+(9-8)^2} = \sqrt{(-5)^2+1^2}$	5.09
F4	(110, 11)	$\sqrt{(95-110)^2+(9-11)^2} = \sqrt{(-15)^2+(-2)^2}$	15.13
F5	(120, 9)	$\sqrt{(95-120)^2+(9-9)^2} = \sqrt{(-25)^2+0^2}$	25
F6	(130, 10)	$\sqrt{(95-130)^2+(9-10)^2} = \sqrt{(-35)^2+(-1)^2}$	35.01

— 2 M

Step1: Sort & Find k-nearest neighbors

Sort the distances in ascending order & select the top k (3) entries.

Distance	Fruit	class
5.09	F2	A
5.09	F3	A
15.13	F4	B
25	F1	A
25	F5	B
35.01	F6	B

Step3: Predict the class

We look at the classes of the 3 nearest neighbors we selected in the previous step.

- F2 : class A
- F3 : class A
- F4 : class B

— 2 M

By majority vote, class A appears twice, while class B appears once. Therefore, the new fruit is classified as an Apple (class A).

— 1 M

9a). Compare SVM and neural networks for classification tasks.

A: SVMs are better for smaller, high-dimensional datasets and provide a simpler, more interpretable model with a guaranteed optimal boundary, while neural networks excel on large, complex datasets, learn hierarchical features, and are better suited for tasks like image & speech recognition. — 1M

SVM:

Strengths:
- effective in high-dimensional space
- works well for smaller datasets.
- good at finding a clear, optimal decision boundary (hyperplane) by maximizing the margin between classes.
- less prone to overfitting
- can handle non-linear classification using kernels. — 2M

Weakness:
- computationally expensive
- scales poorly with very large datasets.
- Requires more careful data pre-processing (missing values)
- less effective with large no. of classes.

Interpretability: higher, as the decision boundary is based on support vectors.

Best for: Text, image classification. — 3M

Neural Nws:

Strengths:
- Excellent for large datasets & complex, non-linear problems.
- can automatically learn hierarchical features, which is crucial for unstructured data like images & audio.
- Extremely versatile, with deep learning variants. — 4M

Weaknesses:
- Requires significant computational resources
- Hyperparameter tuning is more complex
- less interpretable than SVMs, as the decision boundary can be complex & difficult to visualize.
- More prone to overfitting

Interpretability - lower, as the internal workings of deep nws are often considered a "black box"

Best for - NLP, speech recognition, complex image recognition. — 5M

9b). Write the issues related to overfitting in neural networks.

A: Overfitting in NN occurs when a model learns the training data too well, including noise and ~~irrel~~ idiosyncrasies, leading to poor performance on new, unseen data. The key issues are: (1/2 M)

- Poor Generalization:

The most significant issue is the model's inability to generalize to new data. (1/2 M)

- High variance: Overfit models exhibit high variance, meaning they are highly sensitive to the specific training data. (1 M)

- Memorization Vs. learning: Instead of learning the underlying patterns and relationships in the data, an overfit model essentially memorizes the training examples. (1 M)

- Wasted Computational Resources: Training an overly complex model for too long can lead to overfitting and consume significant computational resources. (1 M)

- Misleading Performance Metrics: High training accuracy can be deceptive, as it doesn't reflect the model's true performance on real-world data. (1 M)

10). ~~Compare~~ Discuss how random forests handle missing data and feature correlation with suitable examples.

A: Random forests can handle missing values without explicit imputation in two primary ways:

- Surrogate Splits:

During tree construction, if a primary splitting feature has missing values for certain instances, the algorithm can identify a "surrogate split" based on a correlated feature.

- Proximity-Based Imputation:

Some implementation of random forests leverage the concept of "proximity" between data points to impute missing values. After building the forest, a proximity matrix is generated, indicating how often two instances end up in the same leaf node.

2 M

Handling Feature Correlation:

Random forests are inherently less sensitive to multicollinearity compared to linear models for following reasons:

- Random feature subsampling:

At each split in a decision tree, only a random subset of features is considered for finding the best split. This reduces the chance that highly correlated features will consistently be choosen together, mitigating their combined influence.

- Ensemble Averaging: The final prediction in a random forest is an aggregation of the predictions from multiple independent decision trees. Even if some individual trees are influenced by correlated features, the averaging process tends to smooth out these individual biases. — 2 M (4 marks)

Handling Missing values with Random forest using python.

Here we focus on predicting missing "Age" values in Titanic dataset which is a classic dataset used in Machine learning and data analysis.

* Implementing program.

```
# import libraries.
import pandas as pd
import numpy as np
```

— 2 M

* Loading datasets

```
# importing dataset & setting 'PassengerId' as Index
Data = pd.read_csv('Data.csv', index_col='PassengerId')
Data.head()
```

output will display dataset

* Data preprocessing

Handling Missing values.

code Data.Isnull().sum() is used to check for missing values in a Data frame called Data.

```
# Missing values
Data.isnull().sum()
```

```
# Dropping 'cabin' column due to missing values,
Data = Data.drop(columns=['Cabin'], axis=1)
Data.head()
```

— 2 M

Data.Embarked.value_counts()

* Model Building

1. Importing Random forest Regressor
2. Creating Random forest Regressor
3. Training the model

* Prediction

Predicted_Ages = rf_age.predict(Independent_Features)

11a) Write the Hierarchical clustering and its linkage methods.

5M.

A: Hierarchical clustering groups data points into a tree-like hierarchy, either by progressively merging clusters or splitting a single cluster (divisive). The key to this process is the linkage method, which determines the the distance between clusters. Common linkage methods include single linkage (minimum distance), complete linkage (maximum distance), average linkage (average distance), and ward's method (minimizing the increase in squared error).

—1M

5M.

Hierarchical Clustering Methods:

→ Agglomerative clustering:

This is a bottom-up approach where each data point starts as its own cluster. Clusters are then merged based on similarity until only one cluster remains.

→ Divisive clustering:

This is a top-down approach where all data points is in its own cluster.

→ Dendrogram:

The final output is a tree diagram called a dendrogram, which visualizes the hierarchy of clusters of clusters and the relationships between them. (2 Marks)

~~(2 Marks)~~

Linkage Methods:

→ Single Linkage: The distance between two clusters is the minimum distance between any ~~two points~~, in the two clusters.

(1 M).

→ Complete Linkage:

The distance between two clusters is the maximum distance between any two points in the two clusters.

($\frac{1}{2}$ M).

11 b). Compare the Random forests with Decision Trees?

A: Random Forests and Decision Trees are both widely used machine learning algorithms, but they differ significantly in their structure and performance characteristics.

Decision Trees:

- Single Model: A decision tree is a single, tree-like model that makes predictions by following a series of rules based on input features.
- Interpretability: They are highly interpretable and easy to visualize, making it straightforward to understand the decision-making process.
- Overfitting: Decision trees are prone to overfitting, especially when allowed to grow deep without pruning, as they can become too specific to the training data and fail to generalize to new data.
- Computational Efficiency: They are computationally efficient and fast to train and predict. (2½ marks)

Random Forests:

- Ensemble Model: A random forest is an ensemble learning method that combines multiple decision trees. It builds a "forest" of trees, each trained on a random subset of the data.

- Reduced Overfitting: ✓
  By averaging the predictions of multiple trees, random forests significantly reduce the risk of overfitting and improve generalization.
- Interpretability: Less interpretable than single decision trees due to the ensemble nature, making it harder to visualize the entire decision-making process.
- Computational cost:
  More computationally intensive than single decision trees due to the training of multiple trees.

(2½ Marks)