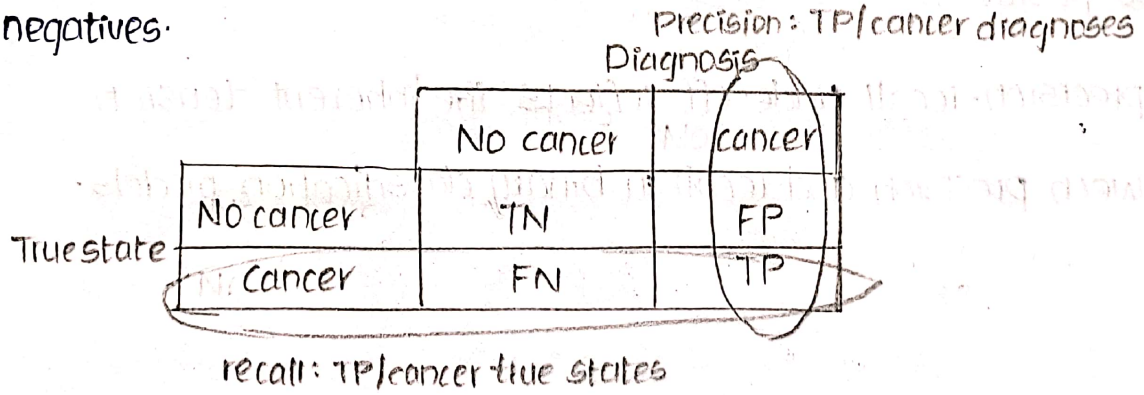
1.a) What is Precision-Recall trade off?

A. Precision and recall are performance metrics used for pattern recognition and classification in machine learning.

- Precision is how many times an accurate prediction of a particular class occurs per a false prediction of that class.
- Recall is the percentage of the data belonging to a particular class which the model properly predicts as belonging to that class.
- Precision : It is the number of true positive predictions divided by the total number of positive predictions. (true positives plus false positives
- Recall : It is number of true positive predictions divided by the total number of ~~the~~ actual positive samples (true positives plus false negatives.



- The Idea behind the precision-recall trade-off is that when a person changes the threshold for determining if a class is positive or negative it will till the scales.

- The precision-recall trade-off arises from the fact that these two metrics are often inversely related.
- Increasing the model's threshold for classifying instances as positive typically leads to higher precision but lower recall.
- Conversely, lowering the threshold increases recall but decreases precision.
- This trade-off occurs because as the threshold increases, the model becomes more conservative in predicting positive instances, reducing the number of false positives but potentially missing some true positives.
- Conversely, lowering threshold leads to more positive predictions, capturing more true positives but potentially introducing more false positives.
- The precision-recall trade-off reflects the inherent tension between precision and recall in binary classification models.

b.) How to choose good threshold for the precision-recall curve?

A. Choosing a good precision-recall tradeoff curve in machine learning involves understanding the problem, considering specific requirements and constraints and evaluating the performance tradeoffs associated with different thresholds.

Here are some steps to choose a suitable precision-recall curve

1. Precision-Recall curve Analysis:

- Plot the precision-recall curve for your model using different threshold values. The curve illustrates the trade-off between precision and recall.
- Analyze the curve and identify the threshold that provides the desired balance between precision and recall based on requirements.

2. F1 Score:

- F1 score is metric that combines precision and recall into a single value
- Compute the F1 score for various threshold values and choose the threshold that maximizes the F1 score.

3. Domain knowledge:

- Consider the specific characteristics of your problem domain
- Depending on the application, you may have domain specific insights that can guide the selection of a threshold

4. Risk Tolerance:

- Evaluate the tradeoff between false positives and false negatives based on your risk tolerance.

- If false positives or false negatives carry significantly different risks, adjust the threshold accordingly to prioritize the outcome.

5. Cost-Sensitive Learning:
   - If the misclassification costs for different classes or error types are known, you can incorporate them into your model training process.
   - By assigning them, the model can learn to optimize the threshold based on the associated costs.

6. Validation and Testing:
   - Use a validation set or cross-validation to assess the performance of your model at different thresholds.
   - Experiment with different threshold values and evaluate the precision, recall and metrics to determine the threshold that best meets your requirements.

- It's important that the choice of threshold is problem-specific and may require an iterative process of experimentation and evaluation.

2. a. How to compare performance of classifier using ROC curve.

Comparing the performance of classifier using ROC curve is a common pratice in machine learning. The ROC curve (Receiver operating characteristic curve) is a graphical representation of the performance of a binary classifier at different classification thresholds. it plots the true positive rate (TPR) against the false positive rate (FPR) for various threshold values.

To compare the performance of classifiers using ROC curves, you can follow these steps

1. Train your classifiers : Start by training multiple classifiers on your labeled training data. Ensure that you have the predicted probabilites & confidence scores for each classifier's predictions.

2. Make predictions : Use the trained classifiers to make predictions on seperate validation (or) test dataset collect the predicted probabilities & confidence scores for each classifier.

3. Compute TPR and FPR : For each classifier, calculate the TPR and FPR at various threshold values varied from 0 to 1

True positive Rate (TPR) : Also known as sensitivity (or) Recall, TPR is calculated as the ratio of true positives and false negatives

TPR = True positives / (True positivies + False Negatives)

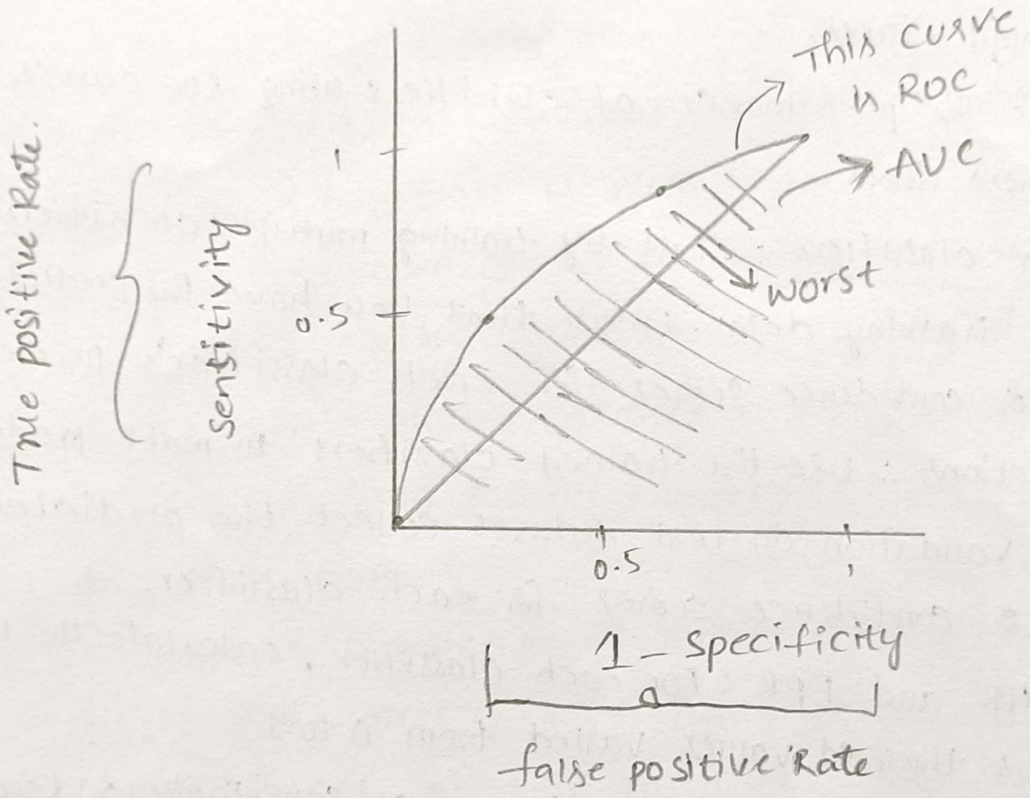
False positive Rate (FPR) : FPR ratio of false positives to the sum of false positives and True negatives.

FPR = False positives / (False positives + True Negatives)

4. plot the ROC curve : Once you have TPR and FPR values for each classifier at different thresholds plot the ROC curve. The ROC curve is typically a line starts from the point (0,0) and ends at (1,1)

5. compare ROC curves : the classifier with the curve closest to the top-left corner (i.e (0,1)) generally indicates the better performance because it acheives higher TPR while keeping FPRs low.

6. Compute AUC-ROC : Another way to compare classifiers is by computing the AUC (Area under curve)-ROC. The more the AUC the better the Model is.



2.

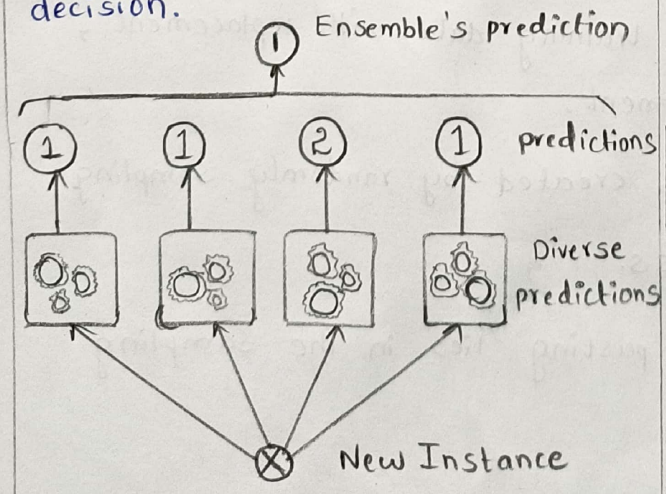
b. When to prefer precision-Recall to Roc curve

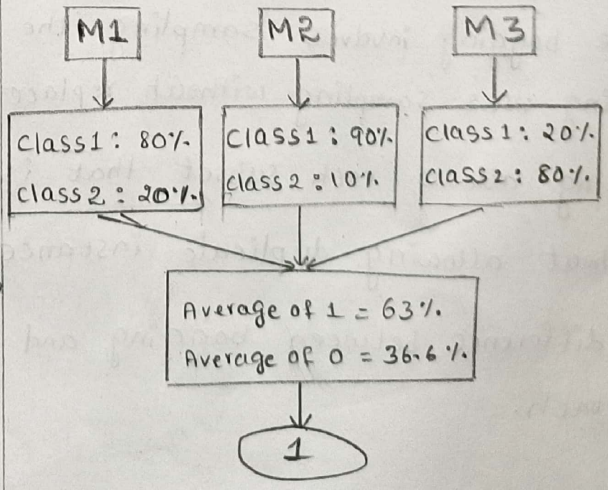
Precision-recall curves and Roc curves are both commonly used evaluation metrics in machine learning, particularly in binary classification problems. while Roc curves are widely used, there are specific scenarios when precision-recall curves may be preferred over Roc curves. Here are few cases

1. Class imbalance : precision-recall curves are particularly useful when dealing with imbalanced datasets, where the number of instances in one class is much higher than another one. In such cases the Roc curve can be misleading because it focuses on TPR and FPR which can be heavily influenced by majority class.
2. Skewed positive class importance : In certain applications, correctly identifying positive instances (high-recall) might be more critical than minimizing false positives (high-precision). precision-Recall curves prioritize recall, making them suitable when the importance of correctly identifying positive instances outweighs the consequences of false positives.
3. Anomalies and Rare Events : precision-Recall curves are benaficial when dealing with anomaly detection or rare event prediction. Since Roc curves are based on ranking the predictions they can be influenced by performance on the majority class.
4. It's important to note that there is no definitive rule on when to prefer precision-recall curves over Roc.

3a) Differentiate Hard Voting Classifier and soft voting classifier.

Hard voting Classifier	Soft voting classifier
* In Hard voting classifier the final prediction is based on the majority vote of the models.	* In soft voting classifier the final prediction is based on the class label with the highest average probability.
* This approach assumes that each model has an equal say in the final decision, regardless of their confidence levels.	* soft voting takes into account the confidence levels of each model's predictions.
* Hard voting classifier is less accurate than soft voting classifier	* Soft voting classifier is more accurate than Hard voting Classifier.
* In this, each model in the ensemble provides a discrete class label prediction.	* In this, each model assigns probabilities to each class label instead of making a discreate prediction.
* Hard voting classifiers do not require the individual models to provide probability estimates.	* soft voting classifiers require the individual models to provide probability estimates for their predictions.
* It assumes that each model contributes equally to the ensemble's decision.	* models with higher confidence have a greater influence on the ensemble's decision.





Write about Bagging and pasting.

Bagging: Bagging, also known as bootstrap aggregation, it is a machine learning ensemble technique that combines the predictions of multiple models to improve the overall performance and generalization of the learning algorithm.

→ The basic idea behind bagging is to create an ensemble of models by training each model on a different subset of the training data.

→ It often used in the context of decision trees.

→ By generating multiple subsets and training models on each subset, we create a diverse set of models that have been exposed to different variations in the training data.

Main advantages of bagging

1. Reduction of variance.
2. Improved generalization.
3. Handling complex relationships.
4. Easy parallelization.

→ It is powerful technique for improving model performance, but it may not always lead to better results.

Pasting:

→ Pasting is another ensemble technique used in machine learning, similar to bagging.

→ while bagging invdves sampling the training data with replacement, pasting uses sampling without replacement.

→ pasting allows each subset that is created by randomly sampling without allowing duplicate instances.

→ Key difference between bagging and pasting lies in the sampling approach.

→ pasting avoids repetition of instances within a subset, which means each instance is used only once in the training of each model.

→ This can be advantageous in situations where training data is limited, and the goal is maximize the utilization of available instances.

Main advantages of pasting

① Utilization of all instances

② Reduced overfitting

③ parallelization.

4 (a) Write about RandomForestClassifier ?

→ Random Forest is a versatile machine learning method capable of performing both regression and classification tasks.

→ It also undertakes dimensional reduction methods, treats missing values, outlier values and other essential steps of data exploration, and does a fairly good job.

→ It is a type of ensemble learning method,

→ where a group of weak models combine to form a powerful model

→ The forest choose the classification having the most votes and in case of regression it takes the average of outputs by different trees

→ A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models

Algorithm works :-

→ Assume number of cases in the training set is N. Then, sample of these N cases (2/3) is taken at random but with replacement. The sample will be the training set for growing the tree

Cons:-

→ It surely does a good job at classification but not as good as for regression problem as it does not give continuous output. In case of regression, it doesn't predict beyond the range in the training data. and that they may over-fit data sets that are particularly noisy

→ Random Forest can feel like a black box approach for statistical modelers you have very little control on what the model does you can at best - try different parameters and random seeds!.

What is the Benefits of out of Bag evaluation?

→ A mechanism for evaluating the quality of a decision forest by testing each decision tree against the examples not used during training of the decision tree.

→ Notice that the system train each decision-tree on about two-thirds of the examples and then evaluates against the remaining one-third of the examples

→ out-of-evaluation is a computationally efficient and conservative approximation of the ~~Cross-validation~~

out-group homogeneity bias:

→ The tendency to see out-group members as more alike than in-group members when comparing attitudes, values, personality traits, and other characteristics that -exists

→ In-group refers to people you interact with regularly

→ out-group refers to people you do not interact with regularly.

→ if you create a data set by asking people to provide attributes about out-groups.

→ these attributes may be less nuanced and more stereotyped than attributes that participants list for people in their in-group

→ outlier detection the process of identifying outliers in a training set

→ input data whose values are more than roughly 3 standard deviations from the mean

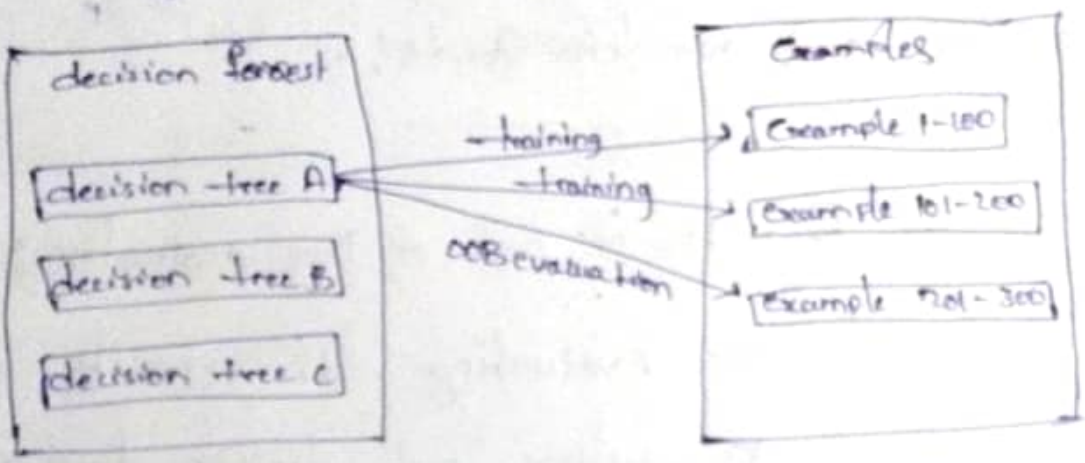
→ weights with high absolute values

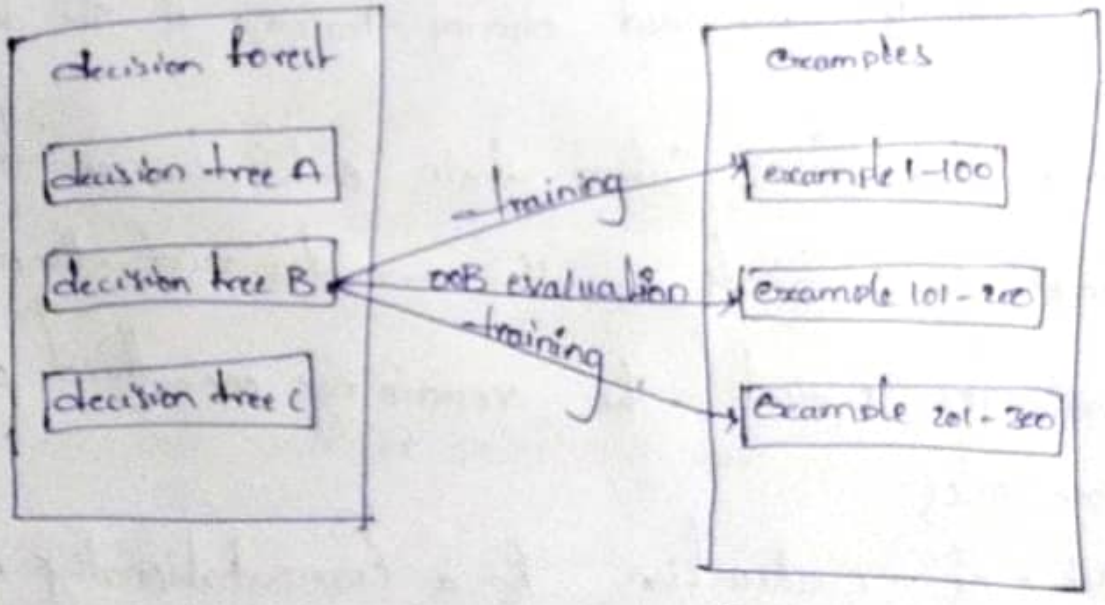
→ Predicted values relatively far away from the actual values

Cross-validation mechanism.

→ In Cross-validation, one model is trained for each Cross-validation round with OOB evaluation, a single model is trained.

→ Because bagging withholds some data from each tree during training, OOB evaluation can use that data to approximate Cross-validation.





5a) Differentiate boosting and Bagging

Bagging	Boosting
1. The simplest way of combining predictions that belong to the same type.	1. The way of combining Predictions that belong to the different types.
2. Aim to decrease variance, not bias.	2. Aim to decrease bias, not variance.
3. Each model receives equal weight.	3. models are weighted according to their performance
4. Each model is built independently.	4. New models are influenced by the performance of previously built models.
5. Different training data subsets are selected using row sampling with replacement and random sampling method from the entire training dataset.	5. every new subset contains the elements that were misclassified by previous models.
6. Bagging tries to solve the over-fitting problem.	6. Boosting tries to reduce bias.
7. if the classifier is unstable (high, variance), then apply bagging.	7. If the classifier is stable and simple then apply boosting.
8. classifiers are trained Parallely.	8. In this base, classifiers are trained sequentially.

9. Ex: The random forest model uses Bagging.

9. Ex: The AdaBoost uses Boosting techniques.

5b) write about adaptive boosting method.

Adaptive boosting method is a powerful method in machine learning.

⇒ AdaBoost is best used to boost the performance of decision trees on binary classification problems.

⇒ AdaBoost was originally called AdaBoost.M1 by the authors of the technique Freund and schapire

⇒ The most suited and therefore most common algorithm used with adaboost are decision trees with one level.

⇒ Because, these trees are so short and only contains one decision for classification, they are often called decision stumps.

⇒ Each instance in the training dataset is weighted. The initial weight is set to:

$$\text{weight}(x_i) = 1/n$$

where x_i is the ith training instance and 'n' is the number of training instances.

AdaBoost algorithm:

⇒ Initialization: Assign equal weights to all training instances, indicating their importance.

⇒ Train a weak learner on the weighted dataset

⇒ Increase the weight of misclassified instances.

⇒ Repeat steps for a predetermined number of iterations or until a specific criterion is met.

⇒ combine the weak learners into a strong classifier using a weighted majority vote.

a) Improved predictive Accuracy:
AdaBoost has been shown to enhance the predictive accuracy of weak classifiers significantly.

b) Robustness to overfitting: AdaBoost's iterative approach and the emphasis on misclassified instances prevent overfitting.
The algorithm reduces the chances of memorizing noise or outliers in the training data. making it more resilient and robust.

c) Versatility: AdaBoost can be used with various types of weak learners, such as decision trees, neural networks, or Support vector machines.
The flexibility allows practitioners to choose the weak learner that best suits the problem at hand,

optimizing overall performance.

d) Handling imbalanced data:

AdaBoost is particularly effective in dealing with imbalanced datasets, where the number of instances in different classes is significantly different.

⇒ By assigning higher weights to misclassified instances, AdaBoost ensures that rare classes receive more attention during the training process, leading to better classification performance.