Week 1: Introduction to ML

MLP

Q. Find shape of the Wisconsin breast cancer data's feature matrix (accessed from sklearn.datasets).

A: (569, 30)

Q. How many benign (B) tumour cases are present in the Wisconsin breast cancer dataset ?

A. 357

Q. How many malignant (M) tumour cases are present in the Wisconsin breast cancer dataset.

A. 212

Load California Housing dataset from sklearn.datasets and answer the following questions:
Q. Find out the shape of the feature matrix in the California housing dataset.

A. (20640, 8)

Q. Find out the labels of the first five attributes in the California housing dataset.

A. [4.526, 3.585, 3.521, 3.413, 3.422]

Q. Find out the name of the class label in the California housing dataset(target matrix)?

A. ['MedHouseVal']

MLT

Q.1 What are the components of machine learning?

1. Data
2. Model
3. Loss function

4. Optimization
5. Evaluation

Q. 2 Define loss function.
Ans: Loss function captures the difference between the actual output and the output predicted from the machine Learning model.

Q.3 What is evaluation in machine learning?
Ans: Computing performance of model on test data based on some pre-decided metric.

Q. 4 What are the types of machine learning?
Ans: Supervised, semi supervised and unsupervised.

Q.5 Define accuracy, precision and recall:
Ans: Accuracy: correctly classified samples out of total samples.
Precision: correctly classified positive samples out of all positive predicted samples.
Recall: correctly classified positive samples out of all positive samples.

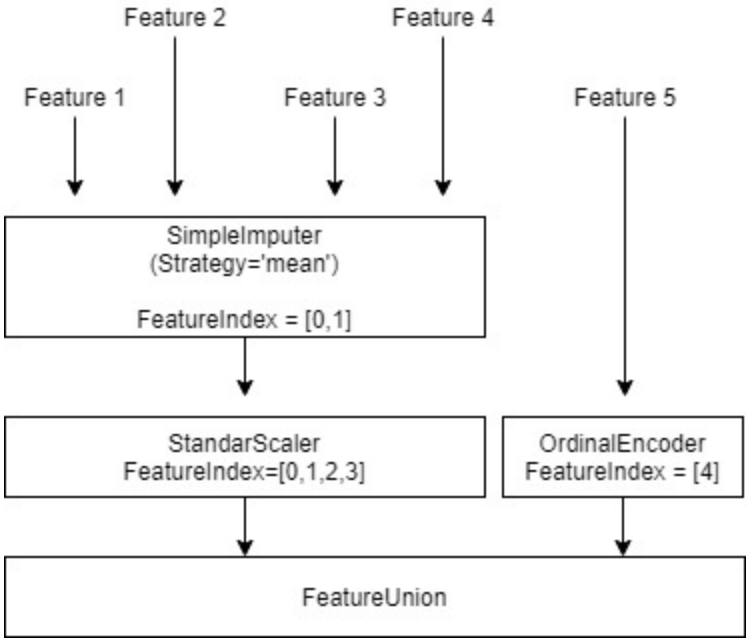
Week 2: Data Preprocessing

MLP

Download dataset from following:
https://drive.google.com/file/d/1lvvHj0v9LKwe6XUezgpIUY_c0HdJUcCb/view?usp=sharing

- What are the two most important features computed by RFE?

  Preprocess the data using pipeline shown in the diagram. Use `LogisticRegression` (with default parameters) for the `estimator`. Encode target variable via ordinal encoding.



A) 0 (i.e. feature V1)

B) 1 (i.e. feature V2)

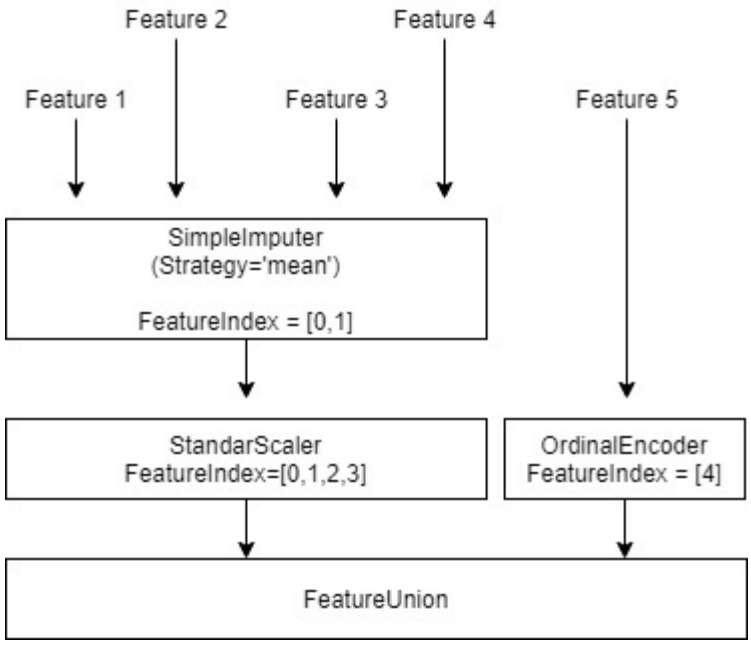
C) 2 (i.e. feature V3)

D) 3 (i.e. feature V4)

****Correct: B and D****

- What are the indices of two most important features computed by SFS (backward)?

  Preprocess the data using pipeline shown in the diagram. Use `LogisticRegression` (with default parameters) for the `estimator`. Encode target variable via ordinal encoding.



A) 0 (i.e. feature V1)

B) 1 (i.e. feature V2)

C) 2 (i.e. feature V3)

D) 3 (i.e. feature V4)

****Correct: C and D****

Week 3: Regression

MLP

Que:

Write a function called **model_test** in python having the following signature:

```
def model_test( X: "Feature matrix", y: "corresponding target vector",
model:"the given regression model", scorer:"the desired scoring routine",
p_threshold= "threshold for p value exceeding which the model would be
rejected", cv_options:" desired cross validation routine or an integer value") ->
Bool
```

Which takes in a regression model and returns True or False depending on whether the permutation_test_score for this model lies below the given p_threshold or not, which is interpreted as whether the model is satisfactory along the given threshold or not.

Answer: True

Week 4: Classification

Que: What is a perceptron?
Ans:
In machine learning, the **perceptron** is an algorithm for supervised learning of binary classifiers. It is a type of linear classifier, i.e. a classification algorithm that makes its predictions based on a activation function combining a set of weights with the feature vector.

in the modern sense, the perceptron is an algorithm for learning a binary classifier called a threshold function as follows:

Rubric:
Students may approach in different manners but the idea of 'weighted sum followed by the activation function' must be clear.

Que: Define the loss function in perceptron algorithm.
Ans:

Error corresponding to ith instance is given as

$$e^{(i)} = \begin{cases} 0, \text{ if } \hat{y}^{(i)} = y^{(i)} \\ -\mathbf{w}^T \phi(x^{(i)}) y^{(i)}, \text{otherwise (i.e. } \hat{y}^{(i)} \neq y^{(i)}) \end{cases}$$

Or

$$\begin{aligned} e^{(i)} &= \max(0, -\mathbf{w}^T \phi(x^{(i)}) y^{(i)}) \\ &= \max(0, -h_{\mathbf{w}}(x^{(i)}) y^{(i)}) \end{aligned}$$

Therefore, loss function is given by

$$
\begin{aligned}
J(\mathbf{w}) &= \sum_{i=1}^{n} e^{(i)} \\
&= \sum_{i=1}^{n} \max(0, -\mathbf{w}^T \phi(x^{(i)}) y^{(i)}) \\
&= \sum_{i=1}^{n} \max(0, -y^{(i)} h_{\mathbf{w}}(x^{(i)}))
\end{aligned}
$$

Que: Describe the optimization procedure in perceptron.
Ans:

1. Initialize $\mathbf{w}^{(0)} = \mathbf{0}$
2. For each training example $(\mathbf{x}^{(i)}, y^{(i)})$:

$$\hat{y}^{(i)} = \text{sign}\left(\mathbf{w}^T \phi(\mathbf{x}^{(i)})\right)$$

$$\mathbf{w}^{(t+1)} := \mathbf{w}^{(t)} + \alpha \left(y^{(i)} - \hat{y}^{(i)}\right) \phi(\mathbf{x}^{(i)})$$

Que:
Describe One-vs-One and One-vs-All approaches.

Ans:
In one-vs-All classification, for the n-class dataset, we have to generate the n-binary classifier models. The number of class labels present in the dataset and the number of generated binary classifiers must be the same. That is he One-vs-All strategy splits a multi-class classification into one binary classification problem per class.

In One-vs-One classification, for the n-class dataset, we have to generate the n* (n-1)/2 binary classifier models. Using this classification approach, we split the primary dataset into one dataset for each class opposite to every other class. That is The One-vs-One strategy splits a multi-class classification into one binary classification problem per each pair of classes.

Que: Can perceptron be used for non-separable binary classification dataset?
Ans: No

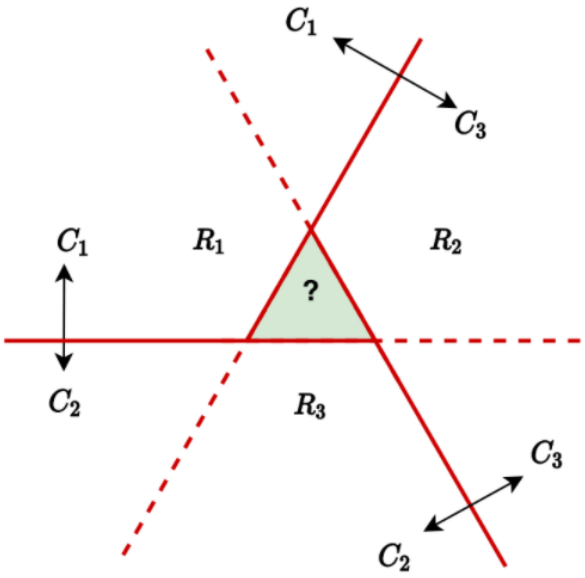
Que: Can perceptron be used for multi-label classification problem?
Ans: No

Que: What does w in the linear discriminant function ($w_0 + wx$) represent geometrically?
Ans:
The vector w is orthogonal to every vector lying within the decision surface, hence it determines the orientation of the decision surface.

Que: What does w_0 in the linear discriminant function ($w_0 + wx$) represent geometrically?
Ans: w_0 determines the location of the decision surface.

Que: What may be the possible issues with One-vs-One Issues?
Ans:

- $k(k-1)/2$ discriminant functions for each class pair C_i and C_j.
- Each discriminant function separates C_i and C_j.
- Each point is classified by majority vote.
- Region of ambiguity is in green.



Ans:

Week 4

MLP

Que 1) What do you understand by hyperparameter. Name some hyperparameter you learned.

Answer

-A hyperparameter is a parameter that is set before the learning process begins. These parameters are tunable and can directly affect how well a model trains.

Some examples of hyperparameters in machine learning:

- Number of Epochs
- Learning Rate
- Number of branches in a decision tree
- The penalty in Logistic Regression Classifier i.e. L1 or L2 regularization
- Number of clusters in a clustering algorithm (like k-means)

Rubric

Learner should be able to explain how hyperparameters are different from parameters and also he should be able to give few examples.

Que 2) Why hyperparameter tuning is needed.

Answer

-Hyperparameter tuning aims to find such parameters where the performance of the model is highest or where the model performance is best and the error rate is least.

Que 3) Mention some common strategy which can be used to optimize hyperparameters.

Answer

- Grid Search: Search a set of manually predefined hyperparameters for the best performing hyperparameter. Use that value. (This is the traditional method)

- Random Search: Similar to grid search, but replaces the exhaustive search with random search. This can outperform grid search when only a small number of hyperparameters are needed to actually optimize the algorithm.

- Bayesian Optimization: Builds a probabilistic model of the function mapping from hyperparameter values to the target evaluated on a validation set.

- Gradient-Based Optimization: Compute gradient using hyperparameters and then optimize hyperparameters using gradient descent.

Rubric
Learners should mention/explain atleast 3 strategy. Assign 1 marks for each strategy.

Que 4) What is the difference between Gridsearch and randomized search? Mention In which case Randomizedsearch will be useful ?

Answer

In Grid Search, we try every combination of a preset list of values of the hyper-parameters and choose the best combination based on the cross-validation score.Whereas Randomizedsearch tries random combinations of a range of values (we have to define the number iterations).

Rubric
Learners should be able to explain the difference between Gridsearch and randomized search clearly.

Week 5: Logistic regression

MLT

Question-1

What is the decision boundary learnt by a logistic regression model?

Answer

Linear

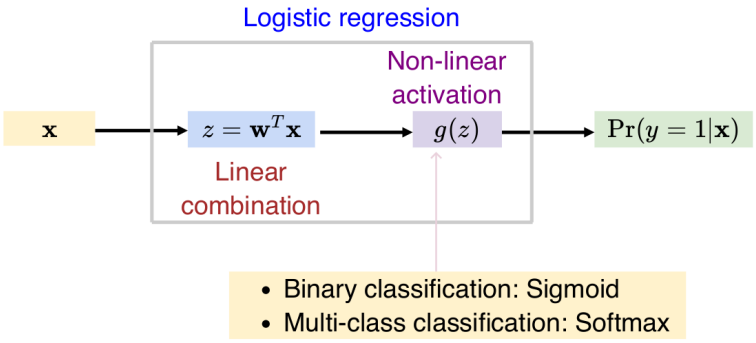
Rubric

Learners should be able to explain why the decision boundary is linear. It is not enough if they just give the correct answer.

Question-2

What is the output of a logistic regression model? Give the mathematical form.

Answer



Rubric

Learners must be able to describe the sigmoid function. They must also be able to say that the output is a probability.

Question-3

What is the loss function used in the case of a logistic regression model for a binary classification problem?

Answer

Binary cross entropy / negative log-likelihood

Question-4

How does inference happen in logistic regression? Give the equation.

Answer

$$y = \begin{cases} 1, & \text{if } \Pr(y = 1|\mathbf{x}) > 0.5 \\ 0, & \text{otherwise.} \end{cases}$$

Rubric

The learners should use the term **threshold.**

Week 6: Logistic Regression

Q: Is the decision boundary Linear or Non-linear in the case of a Logistic Regression model?

A: The decision boundary is a line or a plane that separates the target variables into different classes that can be either linear or nonlinear. In the case of a Logistic Regression model, the decision boundary is a straight line.

Logistic Regression model formula = α+1X1+2X2+….+kXk. This clearly represents a straight line.

It is suitable in cases where a straight line is able to separate the different classes. However, in cases where a straight line does not suffice then nonlinear algorithms are used to achieve better results.

Q: Why is Logistic Regression called Regression and not Classification?

A: Although the task we are targeting in logistic regression is a classification, logistic regression does not actually individually classify things for you: it just gives you probabilities (or log odds ratios in the logit form).

The only way logistic regression can actually classify stuff is if you apply a rule to the probability output. For example, you may round probabilities greater than or equal to 50% to 1, and probabilities less than 50% to 0, and that's your classification.

Q: Between SVM and logistic regression, which algorithm is most likely to work better in the presence of outliers? Why?

SVM is capable of handling outliers better than logistic regression. SVM is affected only by the points closest to the decision boundary. Logistic regression, on the other hand, tries to maximize the conditional likelihood of the training data and is therefore strongly affected by the presence of outliers.

Q: Can you use SGDRegressor to implement logistic regression?
A: SGDRegressor implements logistic regression when given the parameter loss="log".

MLT

Question-1

What is the meaning of the term naive in the context of the Naive Bayes algorithm?

Answer

The features are conditionally independent given the classes. This is a naive assumption and is usually not true real world datasets.

Rubric

The students must mention the conditional independence assumption. That is the most important thing.

Question-2

What are the different kinds of Naive Bayes models? When would you use a Gaussian NB model?

Answer

The choice of model depends on the nature of the features. In general, we have four types discussed in the course.

- Gaussian NB
- Bernoulli NB
- Categorical NB
- Multinomial NB

Gaussian NB is used when the features are continuous.

Question-3

How does inference happen in the case of NB models? That is, given a data-point x, how will you predict the class to which it belongs?

Answer

Bayes theorem. Using one of the models, we can estimate P(x | y). Then we will use the Bayes theorem to estimate P(y | x).

Question-4

N1 and N2 are the number of points that belong to classes 1 and 2. There are two binary features.

A: number of points that belong to class-1 with first feature being 1
B: number of points that belong to class-2 with first feature being 1
C: number of points that belong to class-1 with second feature being 1
D: number of points that belong to class-2 with second feature being 1

What are the parameter estimates for a Bernoulli NB model without smoothing? Wij is the probability that feature i is one given that it belongs to class j.

Answer

W11 = A / N1
W12 = B / N2
W21 = C / N1
W22 = D / N2

Week 7: kNN and softmax regression

MLT:

Q.1 How is a kNN model trained?
Ans: It remembers/copies the training data. No computation is required.

Q.2 How value of k in kNN is associated with overfitting and underfitting?
Ans: kNN models with low values of k tend to overfit and higher values of k tend to underfit..

MLP

Que 1) Why is KNN algorithm called Lazy Learner?

Answer
When it gets the training data, it does not learn and make a model, it just stores the data. It does not derive any discriminative function from the training data. It uses the training data when it actually needs to do some prediction. So, KNN does not immediately learn a model, but delays the learning, that is why it is called lazy learner.

Rubric
Learners should be able to explain KNN doesn't create the model immediately as soon as it gets training data.

Que 2) Why should we not use KNN algorithm for large datasets?

Answer
KNN works well with smaller dataset because it is a lazy learner. It needs to store all the data and then makes decision only at run time. It needs to calculate the distance of a

given point with all other points. So if dataset is large, there will be a lot of processing which may adversely impact the performance of the algorithm.

KNN is also very sensitive to noise in the dataset. If the dataset is large, there are chances of noise in the dataset which adversely affect the performance of KNN algorithm.

Rubric

Learners should mention processing time and its sensitivity to outlier.

Que 3) Why is the odd value of "K" preferable in KNN algorithm?

Answer

K should be odd so that there are no ties in the voting. If square root of number of data points is even, then add or subtract 1 to it to make it odd.

Que 4) How does the KNN algorithm make the predictions on the unseen dataset? Explain the steps involved.

Answer

The following operations have happened during each iteration of the algorithm. For each of the unseen or test data point, the kNN classifier must:

Step-1: Calculate the distances of test point to all points in the training set and store them

Step-2: Sort the calculated distances in increasing order

Step-3: Store the K nearest points from our training dataset

Step-4: Calculate the proportions of each class

Step-5: Assign the class with the highest proportion

Rubric

Learners should clearly explain all steps.

Que 5) Can the KNN algorithm be used when the dependent variable is continuous?

Answer

Yes, KNN can be used for regression problem statements. For regression problem statements, the predicted value is given by the average of the values of its k nearest neighbours.

Rubric

Answering yes or no is not enough. He/She should also explain How output will be calculated in case of regression problem.

Week 8

MLP

Question

- Consider the MNIST dataset, split it into training and test set in 50:50 ratio with `random_state = 42`. Fit a SVM model using pipeline with StandardScalar, SVM classifier `kernel='poly'` and `degree = 3`, `decision_function_shape='ovr'`and `class_weight='balanced'`, `C=10`. Train the model on training data, and make predictions for test data. Generate the Classification report and choose the correct value for weighted avg of f1_score.

(a) 0.96

(b) 0.97

(c) 0.98

(d) 0.99

****Option b****

Question

- Write the function `compute_score(X_train, y_train, X_test, y_test)` to do the following on the `Iris` dataset-

Write your code keeping in mind:
- Split the Iris dataset into train and test set with 70:30 ratio
- Import svm.SVC as 'model'
- kernel as 'poly', regularization parameter as 10 and gamma as 'auto'
- Train the 'model' and mark the computed 'score'

(a) 2.0

(b) 1.0

(c) -1.0

(d) -2.0

****Option b****

Question

- Write the function `compute_score(X_train, y_train, X_test, y_test)` to do the following on the `Iris` dataset-

Write your code keeping in mind:
- Split the Iris dataset into train and test set with 70:30 ratio
- Import svm.SVC as 'model'
- kernel as 'sigmoid', regularization parameter as 25 and gamma as 'auto'
- Train the 'model' and mark the computed 'score'

(a) 0.2666666666666666

(b) 0.9555555555555556

(c) 1.8122222222222222

(d) 2.2111111111111111

Option a

Question

- **Import the `iris` dataset and drop the rows where `class=Iris-setosa`. Apply a pipeline containing a `MinMaxScaler()`function called`Scaler` and a `svm.svc()` called `classifier`. Split the `iris` dataset into 75:25 ratio with `random_state=0`. Mark the correct precision score.**

(a) 0.00

(b) 1.22

(C) 0.96

(d) 2.33

Option c

Week 9: Decision Trees

Concept related questions

Q. What is a Decision Tree Algorithm? Is it parametric or non-parametric?

- Decision Tree is a supervised machine learning algorithm that uses a flowchart like a tree structure to show the predictions that result from a series of feature-based splits. It starts with a root node and ends with a decision made by leaves.
- It is non-parametric as the algorithm doesn't have assumptions about the space distribution and classifier structure. It tries to learn from the data set by traversing to depth till all the leaves are pure or no more splits can be done.

Q. Can Decision Trees deal with non-linear boundaries in the dataset?

Or

Why would you prefer Decision Trees over linear and logistic regression models?

- Decision Trees can deal with non-linear boundaries in the dataset.

- Logistic Regression is (like Linear Regression) are linear classifiers. They cannot be trained on dataset with non-linear boundaries with high accuracy.

Q. Name few Decisions Trees algorithms

What is ID3?

- In decision tree learning, ID3 (Iterative Dichotomiser 3) is an algorithm invented by Ross Quinlan used to generate a decision tree from a dataset. ID3 is the precursor to the C4.5 algorithm, and is typically used in the machine learning and natural language processing domains.
- The ID3 algorithm begins with the original set S as the root node. On each iteration of the algorithm, it iterates through every unused attribute of the set S and calculates the entropy H(S) or the information gain IG(S) of that attribute. It then selects the attribute which has the smallest entropy (or largest information gain) value. The set S is then split or partitioned by the selected attribute to produce subsets of the data. (For example, a node can be split into child nodes based upon the subsets of the population whose ages are less than 50, between 50 and 100, and greater than 100.) The algorithm continues to recurse on each subset, considering only attributes never selected before

Q. What is C4.5?

- C4.5 is an algorithm used to generate a decision tree developed by Ross Quinlan and is an extension of Quinlan's earlier ID3 algorithm.
- C4.5 made a number of improvements to ID3. Some of these are:
  - Handling both continuous and discrete attributes - In order to handle continuous attributes, C4.5 creates a threshold and then splits the list into those whose attribute value is above the threshold and those that are less than or equal to it.[5]
  - Handling training data with missing attribute values - C4.5 allows attribute values to be marked as ? for missing. Missing attribute values are simply not used in gain and entropy calculations.
  - Handling attributes with differing costs.
  - Pruning trees after creation - C4.5 goes back through the tree once it's been created and attempts to remove branches that do not help by replacing them with leaf nodes.

Q. What is different in C5.0 when compared with C4.5?

- Speed - C5.0 is significantly faster than C4.5 (several orders of magnitude)
- Memory usage - C5.0 is more memory efficient than C4.5
- Smaller decision trees - C5.0 gets similar results to C4.5 with considerably smaller decision trees.

- Support for boosting - Boosting improves the trees and gives them more accuracy.
- Weighting - C5.0 allows you to weight different cases and misclassification types.
- Winnowing - a C5.0 option automatically winnows the attributes to remove those that may be unhelpful.

Q. What attribute measure is selected in CART?

- CART uses GINI index, ID3 uses information gain, C4.5 uses gain ratio.

Q. What is Gini Impurity?

- Gini impurity is a measure of how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset

Q. How is Gini Impurity different from Entropy?

- Gini Index has values inside the interval [0, 0.5] whereas the interval of the Entropy is [0, 1].
- Computationally, entropy is more complex since it makes use of **logarithms** and consequently, the calculation of the Gini Index will be faster.

Q. What are the components of a Decision Tree?

- Root Node: A root node is at the beginning of a tree. It represents entire population being analyzed. From the root node, the population is divided according to various features, and those sub-groups are split in turn at each decision node under the root node.
- Splitting: It is a process of dividing a node into two or more sub-nodes.
- Decision Node: When a sub-node splits into further sub-nodes, it's a decision node.
- Leaf Node or Terminal Node: Nodes that do not split are called leaf or terminal nodes.
- Pruning: Removing the sub-nodes of a parent node is called pruning. A tree is grown through splitting and shrunk through pruning.
- Branch or Sub-Tree: A sub-section of decision tree is called branch or a sub-tree, just as a portion of a graph is called a sub-graph.
- Parent Node and Child Node: These are relative terms. Any node that falls under another node is a child node or sub-node, and any node which precedes those child nodes is called a parent node.

Q. Are Decision Trees susceptible to noise in data?

- Yes. Any addition of noise in training data may lead to vastly different features/thresholds/splits and therefore may lead to drastic difference in performance

Q. Do Decision Trees require feature scaling?

- No. Decision Trees are agnostic to different scales of the feature variables.

Q. Is it possible to have a leaf node with two data points having same feature variables but opposite labels/dependent variables?

- Yes. This may be due to mis-labeling data or an anomaly/outlier.

Q. What is Cost Complexity Pruning?

- The **cost complexity** pruning algorithm used in CART is an example of the postpruning approach. This approach considers the cost complexity of a tree to be a function of the number of leaves in the tree and the error rate of the tree (where the **error rate** is the percentage of tuples misclassified by the tree). It starts from the bottom of the tree. For each internal node, *N*, it computes the cost complexity of the subtree at *N*, and the cost complexity of the subtree at *N* if it were to be pruned (i.e., replaced by a leaf node). The two values are compared. If pruning the subtree at node *N* would result in a smaller cost complexity, then the subtree is pruned. Otherwise, it is kept.
- A **pruning set** of class-labeled tuples is used to estimate cost complexity. This set is independent of the training set used to build the unpruned tree and of any test set used for accuracy estimation. The algorithm generates a set of progressively pruned trees. In general, the smallest decision tree that minimizes the cost complexity is preferred.

Q. Is it possible to have a shorter tree with better performance than a bigger tree?

- Yes. Pruning reduces the complexity of a decision tree, and hence improves predictive accuracy by the reduction of overfitting

Q. What are pure and impure leaves?

- Pure Leaves - All the data in the leaf node have same labels
- Impure Leaves – Data labels in the leaf node not being homogenous and having different labels

Q. Decision Trees are prone to overfitting? Explain and what are the remedies available?

- Decision Trees try to recursively partition the data and try to place different data points into pure leaves. In the process, they may overfit to training data and susceptible to high error rate when tested on unseen/test data.
- Some of the remedies available are
  - Cost Complexity Pruning
  - Reduced Error Pruning
  - Limiting the depth till which the tree can travel

Increase the number of different training points so that the model is more robust

Scikit-learn Implementation related questions

Q. What are the different split methods available in DecisionTreeClassifier? How are they different?

- **splitter**{"best", "random"}, default=" best"
- The strategy used to choose the split at each node. Supported strategies are "best" to choose the best split and "random" to choose the best random split.

Q. What happens when you increase max_depth when running a DecisionTreeClassifier?

- By keeping max_depth very high, the DecisionTreeClassifier, generally, overfits

Q. What is the default value of min_samples_split? What is the impact on accuracy if you increase or decrease its value?

- 2
- Depends on the data. It is usually used for reducing overfitting

Q. What is max_features in DecisionTreeClassifier?

The number of features to consider when looking for the best split:

i. If int, then consider max_features features at each split.
ii. If float, then max_features is a fraction and int(max_features * n_features) features are considered at each split.
iii. If "auto", then max_features=sqrt(n_features).
iv. If "sqrt", then max_features=sqrt(n_features).
v. If "log2", then max_features=log2(n_features).
vi. If None, then max_features=n_features.
vii. Note: the search for a split does not stop until at least one valid partition of the node samples is found, even if it requires to effectively inspect more than max_features features.

Q. Which feature in DecisionTreeClassifier helps in setting a threshold for splitting?

- min_impurity_decrease

Q. Can DecisionTreeClassifier handle an imbalanced dataset? Which feature can help with this?

- No. Use class_weight module in DecisionTreeClassifier
- We can use different data balancing techniques using imblearn (preferred)

Q. Which method of tree pruning has been implemented in DecisionTreeClassifier?

- Cost Complexity Pruning

Q. Are functions for measuring the quality of split different in DecisionTreeRegressor and DecisionTreeClassifier?

- DecisionTreeClassifier – gini or entropy

- DecisionTreeRegressor - squared_error , friedman_mse, absolute_error , poisson

Q. How to extract the decision rules from scikit-learn decision-tree?

A. You can use Scikit learn export_text to extract the rules from a tree. Once you've fit your model, you just need two lines of code.

```
from sklearn.tree import export_text
rules = export_text(loan_tree, feature_names=(list(X_train.columns)))
print(rules)
```

Week 10: Ensemble methods

Q: What are the differences between Bagging and Boosting?

A: Bagging mostly aims at reducing variance.
Boosting is mainly focused on reducing bias.
The base models that are considered for boosting are models with a low variance but high bias.

Bagging can be parallelized. The different models are fitted independently from each other.
Boosting can not be parallelized, and it can become too expensive to fit sequentially several complex models.

Q. How is a Random Forest related to Decision Trees?

A. Random forest is an ensemble learning method that works by constructing a multitude of decision trees. A random forest can be constructed for both classification and regression tasks.
Random forest outperforms decision trees, and it also does not have the habit of overfitting the data as decision trees do.

A decision tree trained on a specific dataset will become very deep and cause overfitting. To create a random forest, decision trees can be trained on different subsets of the training dataset, and then the different decision trees can be averaged with the goal of decreasing the variance.

Q: Since Ensemble Learning provides better output most of the time, why do you not use it all the time?

A: Although it provides a better outcome many times, it is not true that it will always perform better.

There are several ensemble methods, each with its own advantages/disadvantages, and choosing one to use depends on the problem at hand.

- If there are models with high variance, then it will benefit from bagging. If the model is biased, it is better to use boosting.
- If the work is in probabilistic setting, the ensemble methods may not work because it is known that boosting delivers poor probability estimates.

Q: What are Weak Learners?

A: In ensemble learning theory, we call weak learners (or base models) models that can be used as building blocks for designing more complex models by combining several of them. Most of the time, these basics models perform not so well by themselves either because they have a high bias (low degree of freedom models, for example) or because they have too much variance to be robust (high degree of freedom models, for example).

Q. How does the AdaBoost algorithm work?

A: Ada-boost or Adaptive Boosting is one of the ensembles boosting classifiers, it means that it will combine multiple poorly performing classifiers (or weak classifiers) to obtain a high accuracy strong classifier.

It works by following the next steps:

- Initially, it selects a training subset randomly.
- It iteratively trains the AdaBoost machine learning model by selecting the training set based on the accurate prediction of the last training.
- It assigns the higher weight to wrong classified observations so that in the next iteration these observations will get a high probability for classification.
- Also, It assigns the weight to the trained classifier in each iteration according to the accuracy of the classifier. The more accurate classifier will get high weight.

- This process iterates until the complete training data fits without any error or until it reaches the specified maximum number of estimators.
- To classify, it performs a vote across all of the learning algorithms built.

Q: In what situations do you not use Ensemble Classifiers?

A: Ensemble classifiers should not be used when the model needs to be interpretable and explainable.

Ensembles can be highly competitive and provide good results, but in situations where the models are too difficult to implement, maintain, modify, or port, it is not a good idea to use ensemble models.

The model that is closest to the true data generating process will always be the best, and will beat most ensemble methods.

Week 11: Clustering

Que 1) What is K-means clustering algorithm?

Answer

-K Means algorithm is a centroid-based clustering (unsupervised) technique. This technique groups the dataset into k different clusters. Each of the clusters has a centroid point which represents the mean of the data points lying in that cluster.The idea of the K-Means algorithm is to find k-centroid points and every point in the dataset will belong to either of the k-sets having minimum Euclidean distance.

Rubric

Learners should mention terms like centroid-based and distance function.

Que 2) Explain different steps involved in K-means clustering algorithm.

Answer

1. Clusters the data into *k* groups where *k* is predefined.
2. Select *k* points at *random* as cluster centers.
3. Assign objects to their closest cluster center according to the *Euclidean distance* function.
4. Calculate the *centroid* or *mean* of all objects in each cluster.
5. Repeat steps 2, 3 and 4 until the same points are assigned to each cluster in consecutive rounds.

Rubric

Learners should clearly explain all the steps.

Que 3) How to determine optimum no of K? Explain the different methods briefly.

Answer

- Elbow Method – This method finds the point of inflection on a graph of the percentage of variance explained to the number of K and finds the elbow point.
- Silhouette method – The silhouette method calculates similarity/dissimilarity score between their assigned cluster and the next best (i.e, nearest) cluster for each of the data points

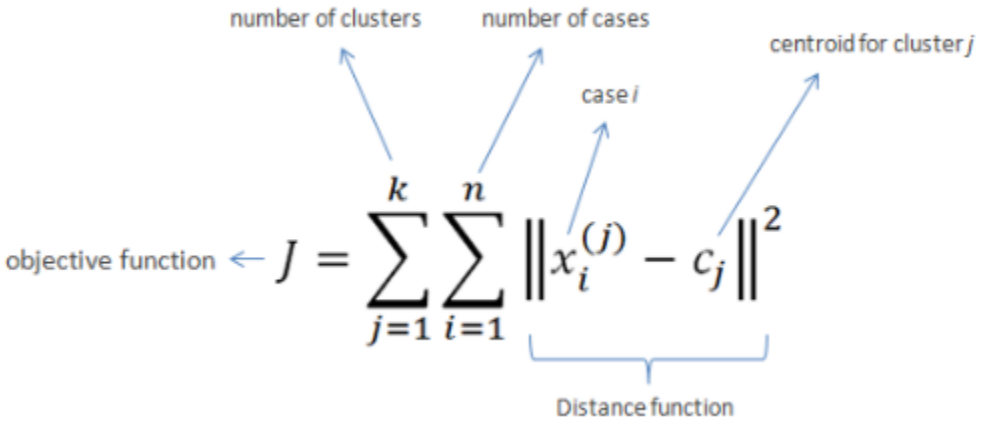
Rubric

Candidate should be able to clearly explain atleast 2 methods.

Que 4) What is the *Objective Function* of *k-Means*?

Answer

- The objective of K-Means clustering is to minimize total intra-cluster variance, or, *distance function*. The objective function of *k-means* depends on the proximities



of the data points to the *cluster centroids*. It is shown below:

- While the selection of distance function is optional, the squared Euclidean distance has been most widely used in both research and practice.

Rubric

Candiadate should be able to clearly explain / write the objective function and distances.

Que 5) What are some *Stopping Criteria* for *k-Means Clustering*?

Answer

1. Convergence. No further changes, points stay in the same cluster.
2. The maximum number of iterations. When the maximum number of iterations has been reached, the algorithm will be stopped. This is done to limit the runtime of the algorithm.
3. Variance did not improve by at least X
4. Variance did not improve by at least X* initial variance.

Rubric
Candidate should explain atleast 2 stopping criteria.

Que 6) Explain some cases where *k-Means clustering* fails to give good results

Answer

- *k-means* has trouble clustering data where clusters are of *various sizes* and *densities*.
- Outliers will cause the *centroids to be dragged*, or the outliers might get their own cluster instead of being ignored. Outliers should be clipped or removed before clustering.
- If the number of dimensions increase, a distance-based similarity measure converges to a constant value between any given examples. *Dimensions should be reduced* before clustering them.

Rubric
Candidate should explain atleast 2 cases.

Que 7) What are the ways to avoid the problem of initialization sensitivity in the K means Algorithm?

Answer
These are the two ways to avoid the problem of initialization sensitivity in sklearn library:

- Repeat K means: It basically repeats the algorithm again and again along with initializing the centroids followed by picking up the cluster which results in the small intracluster distance and large intercluster distance.
- K Means++: It is a smart centroid initialization technique.

Amongst the above two techniques, K-Means++ is the best approach.

Rubric
Learners should explain both methods.

Que 8) Why is the plot of the within-cluster sum of squares error (inertia) vs K in K means clustering algorithm elbow-shaped?

Answer

- Because for max value of k we have min sum of squares (inertia) . hence the total within the cluster sum of squares (inertia).for k=1, all data points are present in the one cluster, and due to more points in the same cluster gives more variance and hence more inertia value.And as K increases inertia score will come down.

Rubric

Learners should be able to explain relation between K and SSE values.

Week 12: Neural Networks

MLT

Question-1

What are some of the non-linear activation functions that are typically used in the hidden layers of neural networks?

Answer

(1) Sigmoid
(2) Tanh
(3) ReLU

Rubric

The student should be able to recall at least two out of the three.

Question-2

If a neural network is used for a single-output regression problem, how many neurons does it have in the output layer?

Answer

1

Rubric

No other answers are accepted.

Question-3

Consider a neural network for a multi-class classification problem with 10 classes. It has the following architecture: [20, 30, 15, 10].

Question-3.1

How many hidden layers does this network have?

Answer

2

Rubric

No other answers are accepted. If the student gives 3 as the answer, give him/her the hint that the first layer is the input.

Question-3.2

How many parameters (weights + biases) does the network have?

Answer

1255

Rubric

It is not enough if the student gives the correct answer. The student should be able to demonstrate the complete working of the solution. It should be something like this: (20 * 30 + 30) + (30 * 15 + 15) + (15 * 10 + 10). Ask the learner what each term represents.

Question-4

In the backpropagation algorithm, what is the sequence in which gradients are computed?

Answer

The gradient of the loss with respect to the weights in the final layer are computed first. We keep going back all the way up to the first layer. If there are n layers, then the sequence is n -> n - 1 -> n - 2 … -> 1

Rubric

If the learner is not able to grasp the question, you can give him/her this hint: “which layer comes first”?

Question-5

Describe the forward pass algorithm.

Answer

Initialize: $A_0 = X$

for l = 1 to l = L:

$$Z_l = A_{l-1} W_l + b_l$$

$$A_l = g(Z_l)$$

Assign: $\hat{Y} = A_L$

Return: $\hat{Y}$

Rubric

The student should be able to give an overview of the algorithm and all the equations correctly. Make sure that they use the terms: pre-activation, activation function and activation while describing various steps.

MLP

Que:
We want to train multi-layer perceptron classifier with three hidden layers with 10 neurons each. Instantiate the classifier object with all other parameters as default.

Ans:
from sklearn.neural_network import MLPClassifier
MLP_clf = MLPClassifier(hidden_layer_sizes=(10,10,10))

Rubric:
Student should able to write module name, API name correctly.
He/She should be able to set the parameter ‘hidden_layer_sizes’ correctly.

Que:
What is the loss function that is used in MLPClassifier?

Ans:
Cross-entropy function.

Rubric:
No other answer is acceptable.

Que:
Can we set different activation functions for different hidden layers in MLPClassifier/ MLPRegressor? Is it same as the activation function at the output layer?

Ans:
All hidden layers typically use the same activation function. The output layer will typically use a different activation function from the hidden layers and is dependent upon the type of prediction required by the model.

Rubric:
No other answer is acceptable.