1. A) GridSearchCV()

2. A) Random forest

3. B) The regularization will decrease

4. A) It regularizes the decision tree by limiting the maximum depth up to which a tree can be grown.

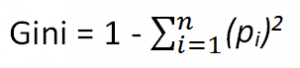
5. In case of classification problem, the prediction is made by taking mode of the class labels predicted by the component trees.

6. A) Gradient Descent algorithm can diverge from the optimal solution

7. B) Bias will decrease, Variance increase

8. B) model is overfitting

9. Gini index is equal to,



where pi  is the probability of an object being classified to a particular class.

So,1-(0.4)^2 + (0.6)^2

=0.48

entropy = -(class0 \* log2(class0) + class1 \* log2(class1))

=0.971

**10**. 1.Random forest leverages the power of multiple decision trees. It does not rely on the feature importance given by a single decision tree, decision tree model gives high importance to a particular set of features. But the random forest chooses features randomly during the training process. Therefore, it does not depend highly on any specific set of features. This is a special characteristic of random forest over bagging trees.

2. **random forests are a strong modeling technique and much more robust than a single decision tree**. They aggregate many decision trees to limit overfitting as well as error due to bias and therefore yield useful results

3.In case of random forest, **Trees are unpruned**. While a single decision tree like CART is often pruned, a random forest tree is fully grown and unpruned, and so, naturally, the feature space is split into more and smaller regions.trees are diverse. Each random forest tree is learned on a random sample, and at each node, a random set of features are considered for splitting. Both mechanisms create diversity among the trees.

**11**. Feature preprocessing is one of the most crucial steps in building a Machine learning model. Too few features and your model won’t have much to learn from. Too many features and we might be feeding unnecessary information to the model. Not only this, **but the values in each of the features need to be considered as well**.

We know that there are some set rules of dealing with categorical data, as in, encoding them in different ways. However, a large chunk of the process involves dealing with continuous variables. There are various methods of dealing with continuous variables. Some of them include converting them to a normal distribution

**Oftentimes, we have datasets in which different columns have different units – like one column can be in kilograms, while another column can be in centimeters**. Furthermore, we can have columns like income which can range from 20,000 to 100,000, and even more; while an age column which can range from 0 to 100(at the most). Thus, Income is about 1,000 times larger than age.**Thereby there is a need for scaling so that data can be evaluated in a uniform manner.**

**Two techniques** under this are listed below-

1.**MinMax Scaler**

The MinMax scaler is one of the simplest scalers to understand.  It just scales all the data between 0 and 1. The formula for calculating the scaled value is-

x\_scaled = (x – x\_min)/(x\_max – x\_min)

2. **Standard Scaler**

Just like the MinMax Scaler, the Standard Scaler is another popular scaler that is very easy to understand and implement.

For each feature, the Standard Scaler scales the values such that the mean is 0 and the standard deviation is 1(or the variance).

x\_scaled = x – mean/std\_dev

**12**. Gradient descent which is an optimization algorithm often used in Logistic Regression, SVM, Neural Networks etc. is another prominent example where if features are on different scale, certain weights are updated faster than others. However, feature **scaling helps in causing Gradient Descent to converge much faster as standardizing all the variables on to the same scale**, for example, for a linear regression makes it easy to calculate the slope ( y = mx + c) (where we normalize the M parameter to converge faster). wo key benefits of Stochastic Gradient Descent are efficiency and the ease of implementation. In a situation when data is less, classifiers in the module are scaled to problems with more than 10^5 training examples and more than 10^5 features.

Stochastic gradient descent is best suited for unconstrained optimisation problems. In contrast to BGD, SGD approximates the true gradient of E(w,b) by considering a single training example at a time.

**13**.No accuracy is not a reliable evaluation matrix in case of imbalanced dataset as they are standard metrics that are widely used for evaluating classification predictive models, such as classification accuracy or classification error.

Standard metrics work well on most problems, which is why they are widely adopted. **But all metrics make assumptions about the problem or about what is important in the problem. Therefore an evaluation metric must be chosen that best captures what you or your project stakeholders believe is important about the model or predictions, which makes choosing model evaluation metrics challenging**.

**This challenge is made even more difficult when there is a skew in the class distribution**. The reason for this is that many of the standard metrics become unreliable or even misleading **when classes are imbalanced, or severely imbalanced, such as 1:100 or 1:1000 ratio** between a minority and majority class.

For example, reporting classification accuracy for a severely imbalanced classification problem could be dangerously misleading. This is the case if project stakeholders use the results to draw conclusions or plan new projects.

Thereby Threshold Metrics play an important role for Imbalanced Classification

Threshold metrics are those that quantify the classification prediction errors.

That is, they are designed to summarize the fraction, ratio, or rate of when a predicted class does not match the expected class in a holdout dataset. sensitivity-Specificity Metrics

Sensitivity refers to the true positive rate and summarizes how well the positive class was predicted.

Sensitivity = TruePositive / (TruePositive + FalseNegative)

Specificity is the complement to sensitivity, or the true negative rate, and summarises how well the negative class was predicted.

Specificity = TrueNegative / (FalsePositive + TrueNegative)

Precision-Recall Metrics

Precision summarizes the fraction of examples assigned the positive class that belong to the positive class.

Precision = TruePositive / (TruePositive + FalsePositive)

Recall summarizes how well the positive class was predicted and is the same calculation as sensitivity.

Recall = TruePositive / (TruePositive + FalseNegative)

Precision and recall can be combined into a single score that seeks to balance both concerns, called the F-score or the F-measure.

F-Measure = (2 \* Precision \* Recall) / (Precision + Recall)

The F-Measure is a popular metric for imbalanced classification.

Another approach is Ranking Metrics for Imbalanced Classification

Rank metrics are more concerned with evaluating classifiers based on how effective they are at separating classes.

The most commonly used ranking metric is the ROC Curve or ROC Analysis.

ROC is an acronym that means Receiver Operating Characteristic and summarizes a field of study for analyzing binary classifiers based on their ability to discriminate classes.

A ROC curve is a diagnostic plot for summarizing the behavior of a model by calculating the false positive rate and true positive rate for a set of predictions by the model under different thresholds.

The true positive rate is the recall or sensitivity.

TruePositiveRate = TruePositive / (TruePositive + FalseNegative)

The false positive rate is calculated as:

FalsePositiveRate = FalsePositive / (FalsePositive + TrueNegative)

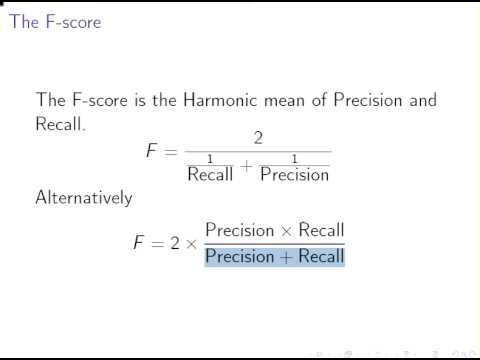
And like the ROC AUC, we can calculate the area under the curve as a score and use that score to compare classifiers. In this case, the focus on the minority class makes the Precision-Recall AUC more useful for imbalanced classification problems.

**14**. **The F-score, also called the F1-score, is a measure of a model’s accuracy on a dataset. It is used to evaluate binary classification systems, which classify examples into ‘positive’ or ‘negative’**.

The **F-score is a way of combining the precision and recall** of the model, and it is defined as the harmonic mean of the model’s precision and recall.

The F-score is commonly used for evaluating information retrieval systems such as search engines, and also for many kinds of machine learning models, in particular in natural language processing.

The formula for the standard F1-score is the harmonic mean of the precision and recall. A perfect model has an F-score of 1.



**15**.fit() is used to "train model" - it means to calculate parameters for transformation.

Only after training model we can transform() data.

If we want then you can do both things in one step using fit\_transform()

So all depends on our problem statement and our approach that whether we want to use fit() + other calculations + transform() or at once fit\_transform().