ENSEM IMP DATA SCIENCE AND BIG DATA ANALYTICS

UNIT – 5

Q.1] Write short note on
i) Time series Analysis
ii) TF - IDF.
ANS: here's a short note on both topics:

i) Time Series Analysis:

1. **Definition: Time series analysis is a statistical technique used to analyze data points collected over time to understand patterns, trends, and behaviors within the data.**
2. **Components: Time series data typically consists of four main components - trend, seasonality, cyclicality, and random variation.**
3. **Applications: Time series analysis is widely used in various fields such as finance for predicting stock prices, economics for analyzing economic indicators, meteorology for weather forecasting, and more.**
4. **Techniques: Common techniques used in time series analysis include moving averages, exponential smoothing, autoregression, and ARIMA (AutoRegressive Integrated Moving Average) models.**
5. **Forecasting: One of the primary objectives of time series analysis is forecasting future values based on historical data, allowing businesses and researchers to make informed decisions and plan effectively.**

ii) TF-IDF (Term Frequency-Inverse Document Frequency):

1. **Definition: TF-IDF is a numerical statistic used in natural language processing and information retrieval to evaluate the importance of a word in a document relative to a collection of documents.**
2. **Calculation: TF-IDF is calculated by multiplying two metrics: term frequency (TF), which measures how often a term appears in a document, and inverse document frequency (IDF), which measures how rare a term is across all documents in the collection.**
3. **Importance: Words with high TF-IDF scores are those that appear frequently within a specific document but are rare in the overall document collection, indicating their importance in distinguishing the document from others.**
4. **Applications: TF-IDF is widely used in text mining, document classification, information retrieval systems (such as search engines), and text summarization.**
5. **Variations: Various variations of TF-IDF exist, including adjusted TF-IDF to mitigate the impact of document length and other normalization techniques to enhance its effectiveness in different contexts.**

Q.2] What is clustering? With suitable example explain the steps involved in k - means algorithm.

ANS: Clustering is a machine learning technique used to group similar data points together based on certain features or characteristics. The goal of clustering is to partition a dataset into distinct groups, or clusters, where data points within the same cluster are more similar to each other than to those in other clusters. One popular clustering algorithm is the k-means algorithm. Here's an explanation of the steps involved in the k-means algorithm with a suitable example:

1. **Initialization:**
   - **Choose the number of clusters, k, that you want to identify in the dataset.**
   - **Randomly initialize k cluster centroids. These centroids represent the initial guesses for the centers of the clusters.**
2. **Assignment:**
   - **For each data point in the dataset, calculate the distance between the data point and each centroid.**
   - **Assign the data point to the cluster whose centroid is closest to it. This step effectively groups data points into k clusters.**
3. **Update Centroids:**
   - **Once all data points are assigned to clusters, recalculate the centroids of the clusters.**
   - **The new centroid of each cluster is the mean of all the data points assigned to that cluster. This step moves the centroid to the center of its respective cluster.**
4. **Repeat:**
   - **Repeat steps 2 and 3 until convergence is reached. Convergence occurs when the centroids no longer change significantly or when a predefined number of iterations is reached.**
5. **Convergence Criteria:**
   - **The algorithm stops when either the centroids do not change significantly between iterations or a maximum number of iterations is reached.**

Example:

Let's consider a dataset containing the annual income and spending score of customers in a mall. We want to segment these customers into different groups based on their income and spending behavior.

1. **Initialization:**
   - **Choose the number of clusters, say k=3.**
   - **Randomly initialize three cluster centroids.**
2. **Assignment:**
   - **Calculate the distance between each data point and the centroids.**
   - **Assign each data point to the cluster with the closest centroid.**
3. **Update Centroids:**
   - **Recalculate the centroids based on the mean of data points in each cluster.**
4. **Repeat:**
   - **Iterate steps 2 and 3 until convergence.**
5. **Convergence Criteria:**

- Stop when the centroids no longer change significantly or after a fixed number of iterations.

Q.3] Write short note on
i) Confusion matrix
ii) AVC - ROC curve
ANS: here's a short note on both topics:

i) Confusion Matrix:

1. **Definition: A confusion matrix is a table used in classification to present a summary of the performance of a machine learning model. It's a matrix of actual classes vs. predicted classes.**
2. **Components: It consists of four main components:**
   - **True Positive (TP): The number of correctly predicted positive instances.**
   - **True Negative (TN): The number of correctly predicted negative instances.**
   - **False Positive (FP): The number of incorrectly predicted positive instances (Type I error).**
   - **False Negative (FN): The number of incorrectly predicted negative instances (Type II error).**
3. **Interpretation: The diagonal elements of the confusion matrix represent correct predictions, while off-diagonal elements represent incorrect predictions.**
4. **Evaluation Metrics: Various evaluation metrics can be derived from the confusion matrix, such as accuracy, precision, recall (sensitivity), specificity, and F1-score.**
5. **Application: Confusion matrices are widely used in binary and multi-class classification problems to assess the performance of classification models and identify areas for improvement.**

ii) AUC-ROC Curve (Area Under the Receiver Operating Characteristic Curve):

1. **Definition: The AUC-ROC curve is a graphical plot that illustrates the performance of a binary classification model at various threshold settings.**
2. **Components: The curve plots the true positive rate (TPR) against the false positive rate (FPR) at different classification thresholds.**
3. **Interpretation: A model with perfect classification performance would have an AUC-ROC value of 1, indicating that it achieves a TPR of 1 and an FPR of 0 across all thresholds. A random classifier would have an AUC-ROC value of 0.5, forming a diagonal line from (0,0) to (1,1).**
4. **Evaluation: The AUC-ROC curve provides a single scalar value, the AUC (Area Under the Curve), which quantifies the model's ability to discriminate between positive and negative classes across all thresholds.**
5. **Application: The AUC-ROC curve is commonly used to compare the performance of different classification models, select the optimal threshold for a given model, and assess the trade-off between sensitivity and specificity.**

Q.4] Discuss Holdout method and Random Sub Sampling methods.
ANS: here's a simple explanation of the Holdout method and Random Subsampling method:

Holdout Method:

1. **Definition: The holdout method is a simple technique used for evaluating the performance of a machine learning model by splitting the dataset into two subsets: a training set and a testing set.**
2. **Splitting: The dataset is randomly divided into two portions - typically, the training set contains a larger proportion of the data (e.g., 70-80%), while the testing set contains the remaining portion (e.g., 20-30%).**
3. **Training: The model is trained using the training set, which involves learning the patterns and relationships within the data.**
4. **Testing: The trained model is then evaluated using the testing set to assess its performance on unseen data. The testing set acts as a proxy for real-world data, helping to gauge how well the model generalizes to new observations.**
5. **Evaluation: Performance metrics such as accuracy, precision, recall, and F1-score are calculated based on the model's predictions on the testing set, providing insights into its effectiveness.**

Random Subsampling Method:

1. **Definition: The random subsampling method, also known as cross-validation, is a technique used for model evaluation and validation by partitioning the dataset into multiple subsets, or folds.**
2. **Partitioning: The dataset is randomly divided into k non-overlapping folds of approximately equal size.**
3. **Training and Testing: The model is trained k times, each time using k-1 folds as the training set and the remaining fold as the testing set. This process ensures that each data point is used for both training and testing, reducing bias in the evaluation.**
4. **Evaluation: After training and testing the model k times, the performance metrics are averaged across all folds to obtain a robust estimate of the model's performance.**
5. **Variations: Common variations of the random subsampling method include k-fold cross-validation, leave-one-out cross-validation (LOOCV), and stratified cross-validation, which ensures that each fold maintains the same class distribution as the original dataset.**

Q.5] Suppose that the given data the taste is to cluster points (With (x.y) representing location) into three cluster, where the points are.
A1(2,10), A2(2,5), A3(8,4), B1 (5,8) B2(7,5) B3(6,4), C1(1,2), C2(4,9)
The distance function is Euclidean distance suppose initially we assign A1, B1 and C1 as the center of each cluster, respectively. use the k- means algorithm to show only the three cluster centers after the first round of execution with steps.
ANS: let's apply the k-means algorithm to the given data with the initial cluster centers A1, B1, and C1.

1. Initialization:
   - Cluster centers: A1(2,10), B1(5,8), C1(1,2)
   - Data points: A2(2,5), A3(8,4), B2(7,5), B3(6,4), C2(4,9)
2. Assignment Step:
   - Calculate the Euclidean distance between each data point and each cluster center.
   - Assign each data point to the nearest cluster center.

For each data point:

- Distance to A1:
  - A2: √((2-2)^2 + (5-10)^2) = √25 = 5
  - A3: √((8-2)^2 + (4-10)^2) = √52 ≈ 7.21
  - B2: √((7-2)^2 + (5-8)^2) = √18 ≈ 4.24
  - B3: √((6-2)^2 + (4-8)^2) = √20 ≈ 4.47
  - C2: √((4-2)^2 + (9-10)^2) = √2 ≈ 1.41
- Distance to B1:
  - A2: √((2-5)^2 + (5-8)^2) = √18 ≈ 4.24
  - A3: √((8-5)^2 + (4-8)^2) = √18 ≈ 4.24
  - B2: √((7-5)^2 + (5-8)^2) = √5 ≈ 2.24
  - B3: √((6-5)^2 + (4-8)^2) = √5 ≈ 2.24
  - C2: √((4-5)^2 + (9-8)^2) = √2 ≈ 1.41
- Distance to C1:
  - A2: √((2-1)^2 + (5-2)^2) = √5 ≈ 2.24
  - A3: √((8-1)^2 + (4-2)^2) = √65 ≈ 8.06
  - B2: √((7-1)^2 + (5-2)^2) = √53 ≈ 7.28
  - B3: √((6-1)^2 + (4-2)^2) = √41 ≈ 6.40
  - C2: √((4-1)^2 + (9-2)^2) = √58 ≈ 7.62

Assign each point to the nearest cluster center:

- A2, C2 to A1
- A3, B2, B3 to B1
- C1 to C1

3. Update Step:
   - Calculate the mean of the points assigned to each cluster and update the cluster centers.

New cluster centers:

- A1: Mean of (2,5) and (1,9) = ((2+1)/2, (5+9)/2) = (1.5, 7)
- B1: Mean of (8,4), (7,5), and (6,4) = ((8+7+6)/3, (4+5+4)/3) = (7, 4.33) (rounded to two decimal places)
- C1: Mean of (1,2) = (1, 2)

So, after the first round of execution, the updated cluster centers are:

- A1(1.5, 7)

- **B1(7, 4.33)**
- **C1(1, 2)**

Q.6] Explain the following text analysis steps with suitable example. i) Part of speech (POS) tagging ii) Lemmatization iii) Stemming

ANS: here's an explanation of the text analysis steps with suitable examples:

i) Part of Speech (POS) Tagging:

1. **Definition: POS tagging is the process of assigning grammatical categories (such as noun, verb, adjective, etc.) to each word in a sentence.**
2. **Example:**
   - **Sentence: "The quick brown fox jumps over the lazy dog."**
   - **POS Tagging:**
     - **"The": Determiner**
     - **"quick": Adjective**
     - **"brown": Adjective**
     - **"fox": Noun**
     - **"jumps": Verb**
     - **"over": Preposition**
     - **"the": Determiner**
     - **"lazy": Adjective**
     - **"dog": Noun**
3. **Importance: Helps in understanding the syntactic structure of the sentence and aids in tasks like named entity recognition, sentiment analysis, and word sense disambiguation.**

ii) Lemmatization:

1. **Definition: Lemmatization is the process of reducing words to their base or dictionary form (lemma), considering the context and meaning of the word.**
2. **Example:**
   - **Word: "running"**
   - **Lemma: "run"**
3. **Importance: Helps in normalization of words, reducing inflectional forms to a common base, which improves the accuracy of text analysis tasks like information retrieval and text classification.**

iii) Stemming:

1. **Definition: Stemming is the process of removing suffixes or prefixes from words to obtain their root or base form, often using simple rules without considering the context.**
2. **Example:**
   - **Word: "running"**
   - **Stem: "run"**
3. **Importance: Useful for reducing words to their simplest form, but may result in non-real words (e.g., "runn" instead of "run"), which could affect the accuracy in certain text analysis tasks like sentiment analysis and topic modeling.**

These text analysis techniques are essential for various natural language processing tasks, aiding in better understanding and processing of textual data.

Q.7] Given the confusion matrix, calculate accuracy. precision, Recall, Error rate with description on heart attact risk.

		Predicted classes	
	Classes	**Heart-Attack Risk-yes**	**Heart Attack Risk-No**
Actual Classes	**Heart Attack Risk-yes**	**80**	**220**
	Heart Attack Risk-No	**150**	**9,500**

ANS: To calculate accuracy, precision, recall, and error rate, we'll first define these metrics:

- **Accuracy: The proportion of correctly classified instances out of the total instances. It's calculated as (TP + TN) / (TP + TN + FP + FN), where TP is the true positive, TN is the true negative, FP is the false positive, and FN is the false negative.**
- **Precision: The proportion of correctly predicted positive cases out of all predicted positive cases. It's calculated as TP / (TP + FP).**
- **Recall (Sensitivity): The proportion of correctly predicted positive cases out of all actual positive cases. It's calculated as TP / (TP + FN).**
- **Error Rate: The proportion of incorrectly classified instances out of the total instances. It's calculated as (FP + FN) / (TP + TN + FP + FN).**

Given the confusion matrix:

		Predicted classes	
		Risk-yes	**Risk-No**
Actual	**Risk-yes**	**80**	**220**
classes	**Risk-No**	**150**	**9,500**

We can calculate the metrics as follows:

- **TP (True Positive): 80 (Predicted Risk-yes, Actual Risk-yes)**
- **TN (True Negative): 9,500 (Predicted Risk-No, Actual Risk-No)**
- **FP (False Positive): 220 (Predicted Risk-yes, Actual Risk-No)**
- **FN (False Negative): 150 (Predicted Risk-No, Actual Risk-yes)**

Now, we can calculate the metrics:

1. **Accuracy:**
    - **Accuracy = (TP + TN) / (TP + TN + FP + FN)**
    - **Accuracy = (80 + 9,500) / (80 + 9,500 + 220 + 150)**
    - **Accuracy ≈ (80 + 9,500) / 9,950**
    - **Accuracy ≈ 9580 / 9,950**
    - **Accuracy ≈ 0.9638 or 96.38%**
2. **Precision:**
    - **Precision = TP / (TP + FP)**
    - **Precision = 80 / (80 + 220)**
    - **Precision = 80 / 300**
    - **Precision ≈ 0.2667 or 26.67%**
3. **Recall (Sensitivity):**
    - **Recall = TP / (TP + FN)**
    - **Recall = 80 / (80 + 150)**
    - **Recall = 80 / 230**
    - **Recall ≈ 0.3478 or 34.78%**

4. **Error Rate:**
    - **Error Rate = (FP + FN) / (TP + TN + FP + FN)**
    - **Error Rate = (220 + 150) / (80 + 9,500 + 220 + 150)**
    - **Error Rate ≈ 370 / 9,950**
    - **Error Rate ≈ 0.0372 or 3.72%**

Description:

- **Accuracy: The accuracy of the model in predicting both positive and negative cases of heart attack risk is approximately 96.38%. This indicates that the model is highly accurate in classifying instances correctly.**
- **Precision: The precision of the model in predicting positive cases (heart attack risk-yes) is approximately 26.67%. This means that out of all instances predicted as having a heart attack risk, only about 26.67% actually have a heart attack risk.**
- **Recall: The recall (sensitivity) of the model in predicting positive cases (heart attack risk-yes) is approximately 34.78%. This means that out of all actual instances with a heart attack risk, the model correctly identifies around 34.78% of them.**
- **Error Rate: The error rate of the model is approximately 3.72%. This indicates that the model makes incorrect predictions for about 3.72% of instances.**

Q.8] Explain the TF/IDF (term frequency-inverse document frequency) terms in text analysis with suitable example.

ANS: here's an explanation of TF-IDF (Term Frequency-Inverse Document Frequency) in text analysis:

1. **Term Frequency (TF):**
   - **Definition: Term frequency measures the frequency of a term (word) in a document. It indicates how often a particular word occurs in a document.**
   - **Calculation: TF is calculated by dividing the number of times a term appears in a document by the total number of terms in the document.**
   - **Example: In a document containing 100 words, if the word "apple" appears 5 times, then the term frequency of "apple" in that document would be 5/100 = 0.05.**

2. **Inverse Document Frequency (IDF):**
   - **Definition: Inverse Document Frequency measures the importance of a term across a collection of documents (corpus). It indicates how rare or common a term is across all documents in the corpus.**
   - **Calculation: IDF is calculated by dividing the total number of documents in the corpus by the number of documents containing the term, and then taking the logarithm of that quotient.**
   - **Example: If there are 1,000 documents in the corpus and the term "apple" appears in 100 of them, then the IDF of "apple" would be log(1000/100) = log(10) = 1.**

3. **TF-IDF (Term Frequency-Inverse Document Frequency):**
   - **Definition: TF-IDF combines the TF and IDF scores to determine the importance of a term in a specific document relative to its importance in the entire corpus. It helps in identifying the significance of a term within a document while considering its rarity across all documents.**
   - **Calculation: TF-IDF is calculated by multiplying the TF score of a term in a document by the IDF score of the term across all documents.**
   - **Example: If the TF of "apple" in a document is 0.05 and the IDF of "apple" across the corpus is 1, then the TF-IDF score of "apple" in that document would be 0.05 * 1 = 0.05.**

4. **Usage:**
   - **TF-IDF is commonly used in text mining and information retrieval tasks, such as document classification, document clustering, and search engine ranking.**
   - **It helps in identifying the most important words in a document by giving higher weights to terms that are frequent in the document but rare across the corpus.**
   - **Words with higher TF-IDF scores are considered more significant and informative for understanding the content of a document.**

Q.9] Define following terms with respect to confusion matrix :
i) Accuracy
ii) Precision
iii) Recall
iv) AUC-ROC
ANS: here are the definitions of the terms with respect to a confusion matrix:

i) Accuracy:

- **Definition: Accuracy measures the overall correctness of the classification model. It calculates the proportion of correctly classified instances out of the total instances.**
- **Calculation: Accuracy = (TP + TN) / (TP + TN + FP + FN), where TP is the true positive, TN is the true negative, FP is the false positive, and FN is the false negative.**
- **Interpretation: Accuracy tells us how often the classifier is correct across all classes.**

ii) Precision:

- **Definition: Precision measures the accuracy of positive predictions made by the model. It calculates the proportion of correctly predicted positive cases out of all predicted positive cases.**
- **Calculation: Precision = TP / (TP + FP), where TP is the true positive and FP is the false positive.**
- **Interpretation: Precision tells us how precise the classifier is when it predicts positive instances.**

iii) Recall:

- **Definition: Recall, also known as sensitivity or true positive rate, measures the ability of the model to find all positive instances. It calculates the proportion of correctly predicted positive cases out of all actual positive cases.**
- **Calculation: Recall = TP / (TP + FN), where TP is the true positive and FN is the false negative.**
- **Interpretation: Recall tells us how many of the actual positive instances were correctly classified by the model.**

iv) AUC-ROC (Area Under the Receiver Operating Characteristic Curve):

- **Definition: AUC-ROC measures the performance of a classification model at various thresholds. It represents the area under the ROC curve, which plots the true positive rate (sensitivity) against the false positive rate (1 - specificity) for different threshold values.**
- **Interpretation: AUC-ROC provides an aggregate measure of the model's ability to distinguish between different classes. A higher AUC-ROC value (closer to 1) indicates better performance, while a value of 0.5 suggests that the model performs no better than random guessing.**

Q.10] Explain k-fold Cross Validation & Random Subsampling.
ANS: here's an explanation of k-fold cross-validation and random subsampling:

1. **k-fold Cross-Validation:**
   - **Definition: K-fold cross-validation is a technique used to assess the performance of a predictive model by partitioning the dataset into k equal-sized subsets (folds). The model is trained and evaluated k times, using a different fold as the validation set each time, while the remaining folds are used for training.**
   - **Procedure:**
     1. **Divide the dataset into k equal-sized subsets (folds).**
     2. **Train the model k times, each time using a different fold as the validation set and the remaining k-1 folds as the training set.**
     3. **Calculate the performance metric (e.g., accuracy, precision, recall) for each fold.**
     4. **Average the performance metrics across all k folds to obtain the overall performance estimate of the model.**
   - **Benefits:**
     - **Provides a more reliable estimate of model performance compared to a single train-test split.**
     - **Reduces the risk of overfitting or underfitting by using multiple validation sets.**
     - **Maximizes the use of data for both training and validation purposes.**

2. **Random Subsampling:**
   - **Definition: Random subsampling, also known as random splitting or holdout validation, is a technique used to evaluate the performance of a predictive model by randomly partitioning the dataset into a training set and a validation set.**
   - **Procedure:**
     0. **Randomly divide the dataset into a training set and a validation set, typically using a specified ratio (e.g., 70% training, 30% validation).**
     1. **Train the model on the training set and evaluate its performance on the validation set.**
     2. **Repeat the process multiple times (usually 1 or more) to obtain multiple performance estimates.**
   - **Benefits:**
     - **Simple and easy to implement.**
     - **Provides a quick estimate of model performance.**
     - **Useful when computational resources are limited or when dealing with large datasets.**
   - **Drawbacks:**
     - **Results may vary depending on the random partitioning of the data.**
     - **May not utilize the entire dataset effectively, especially when dealing with limited data.**

Q.11] What is text processing? Explain TF-IDF with example.

ANS: Here's an explanation of text processing and TF-IDF with an example:

1. **Text Processing:**
    - **Definition: Text processing refers to the technique of transforming raw text data into a format suitable for analysis. It involves several steps, such as cleaning, tokenization, normalization, and feature extraction, to prepare the text data for machine learning or natural language processing tasks.**
    - **Steps:**
        1. **Cleaning: Removing unnecessary characters, punctuation, or symbols from the text data.**
        2. **Tokenization: Breaking down the text into individual words or tokens.**
        3. **Normalization: Converting all text to lowercase, removing stop words, and stemming or lemmatizing words to their base forms.**
        4. **Feature Extraction: Transforming the text into numerical or vector representations that machine learning algorithms can understand and process.**
2. **TF-IDF (Term Frequency-Inverse Document Frequency):**
    - **Definition: TF-IDF is a statistical measure used to evaluate the importance of a word in a document relative to a collection of documents (corpus). It combines two metrics: term frequency (TF) and inverse document frequency (IDF).**
    - **Calculation:**
        - **Term Frequency (TF): Measures the frequency of a term in a document. It is calculated by dividing the number of times a term appears in a document by the total number of terms in the document.**
        - **Inverse Document Frequency (IDF): Measures the rarity of a term across all documents in the corpus. It is calculated by dividing the total number of documents in the corpus by the number of documents containing the term, and then taking the logarithm of that quotient.**
        - **TF-IDF Score: It is obtained by multiplying the TF score of a term in a document by the IDF score of the term across all documents.**
    - **Purpose: TF-IDF helps identify the significance of a term in a document by considering both its frequency in the document and its rarity across the entire corpus.**
    - **Example:**
        - **Consider a corpus containing three documents:**
            - **Document 1: "The cat sat on the mat."**
            - **Document 2: "The dog played with the ball."**
            - **Document 3: "The cat and the dog are friends."**
        - **Let's calculate the TF-IDF score for the term "cat" in Document 1:**
            - **Term Frequency (TF) = Number of times "cat" appears in Document 1 / Total number of terms in Document 1 = 1/6**

        - **Inverse Document Frequency (IDF) = log(Number of documents in the corpus / Number of documents containing "cat") = log(3/2) ≈ 0.176**
        - **TF-IDF Score = TF * IDF ≈ (1/6) * 0.176 ≈ 0.0293**

Q.12] With suitable example ,explain the steps involved in k-means algorithm.
ANS: Here's an explanation of the steps involved in the k-means algorithm with a suitable example:

1. **Initialization:**
    - **Randomly initialize k cluster centroids (points) in the feature space.**
    - **Example: Suppose we have a dataset of 100 points in 2D space and want to cluster them into 3 clusters. We randomly choose 3 points as the initial centroids.**
2. **Assignment Step:**
    - **Assign each data point to the nearest cluster centroid.**
    - **Example: For each data point, calculate the distance to each centroid and assign it to the cluster whose centroid is closest.**
3. **Update Step:**
    - **Recalculate the centroids of the clusters based on the mean of the data points assigned to each cluster.**
    - **Example: Calculate the mean of the data points assigned to each cluster and update the centroid to the calculated mean.**
4. **Convergence Check:**
    - **Repeat steps 2 and 3 until convergence criteria are met, such as:**
        - **Centroids do not change significantly between iterations.**
        - **Maximum number of iterations is reached.**
    - **Example: After updating the centroids, check if they have converged. If not, repeat steps 2 and 3 until convergence is achieved.**
5. **Finalization:**
    - **Once convergence is achieved, the algorithm assigns each data point to its final cluster.**
    - **Example: After the final iteration, each data point will be assigned to one of the clusters based on the nearest centroid.**
6. **Evaluation:**
    - **Evaluate the quality of the clusters using metrics such as inertia, silhouette score, or external validation measures.**
    - **Example: Calculate the inertia (sum of squared distances of samples to their closest cluster center) to assess how compact the clusters are.**
7. **Visualization:**
    - **Visualize the clusters and centroids in the feature space to interpret the results.**
    - **Example: Plot the data points and centroids on a 2D graph to visualize how the algorithm has clustered the data.**

Example:

- **Suppose we have the following data points in 2D space: [(1, 2), (2, 3), (3, 4), (8, 7), (9, 8), (10, 9)].**
- **We want to cluster these points into 2 clusters using the k-means algorithm.**
- **Randomly initialize two centroids: C1(2, 3) and C2(9, 8).**

- **Assign each data point to the nearest centroid based on Euclidean distance.**
- **Update the centroids based on the mean of the data points assigned to each cluster.**
- **Repeat steps 2 and 3 until convergence is achieved.**
- **Finally, visualize the clusters and centroids to interpret the results.**

Q.13] Given the confusion matrix, Calculate Accuracy, Precision, Recall, Error rate with description on Diabetic Risk.

		Predicted classes	
	Classes	**Diabetic Risk -Yes**	**Diabetic Risk -No**
Actual classes	**Diabetic Risk-Yes**	**90**	**210**
	Diabetic Risk-NO	**140**	**9560**

ANS: To calculate Accuracy, Precision, Recall, and Error Rate based on the provided confusion matrix for Diabetic Risk, we first define these metrics:

- **Accuracy: The proportion of correctly classified instances out of the total instances.**
- **Precision: The proportion of correctly predicted positive cases out of all predicted positive cases.**
- **Recall: The proportion of correctly predicted positive cases out of all actual positive cases.**
- **Error Rate: The proportion of incorrectly classified instances out of the total instances.**

Given the confusion matrix:

		Predicted classes	
		Diabetic Risk-Yes	**Diabetic Risk-No**
Actual classes	**Diabetic Risk-Yes**	**90**	**210**
	Diabetic Risk-No	**140**	**9560**

We can calculate the metrics as follows:

- **True Positive (TP): 90 (Predicted Diabetic Risk-Yes, Actual Diabetic Risk-Yes)**
- **True Negative (TN): 9560 (Predicted Diabetic Risk-No, Actual Diabetic Risk-No)**
- **False Positive (FP): 210 (Predicted Diabetic Risk-Yes, Actual Diabetic Risk-No)**
- **False Negative (FN): 140 (Predicted Diabetic Risk-No, Actual Diabetic Risk-Yes)**

1. **Accuracy:**
   - **Accuracy = (TP + TN) / (TP + TN + FP + FN)**
   - **Accuracy = (90 + 9560) / (90 + 9560 + 210 + 140)**
   - **Accuracy ≈ (90 + 9560) / 10000**
   - **Accuracy ≈ 9650 / 10000**
   - **Accuracy ≈ 0.965 or 96.5%**
2. **Precision:**
   - **Precision = TP / (TP + FP)**
   - **Precision = 90 / (90 + 210)**

- Precision = 90 / 300
- Precision = 0.3 or 30%

3. Recall:
    - Recall = TP / (TP + FN)
    - Recall = 90 / (90 + 140)
    - Recall = 90 / 230
    - Recall ≈ 0.391 or 39.1%
4. Error Rate:
    - Error Rate = (FP + FN) / (TP + TN + FP + FN)
    - Error Rate = (210 + 140) / (90 + 9560 + 210 + 140)
    - Error Rate = 350 / 10000
    - Error Rate = 0.035 or 3.5%

Description:

- **Accuracy: The accuracy of the model in predicting both positive and negative cases of diabetic risk is approximately 96.5%. This indicates that the model is highly accurate in classifying instances correctly.**
- **Precision: The precision of the model in predicting positive cases (diabetic risk-yes) is approximately 30%. This means that out of all instances predicted as having a diabetic risk, only about 30% actually have a diabetic risk.**
- **Recall: The recall of the model in predicting positive cases (diabetic risk-yes) is approximately 39.1%. This means that out of all actual instances with a diabetic risk, the model correctly identifies around 39.1% of them.**
- **Error Rate: The error rate of the model is approximately 3.5%. This indicates that the model makes incorrect predictions for about 3.5% of instances.**

Q.14] Explain the Text Preprocessing steps with suitable example.

ANS: Here's an explanation of text preprocessing steps with a suitable example:

1. **Text Lowercasing:**
    - **Definition: Convert all text to lowercase to ensure consistency and prevent the algorithm from treating words with different cases as different entities.**
    - **Example:**
        - **Original Text: "The quick Brown Fox Jumps Over The Lazy Dog."**
        - **Lowercased Text: "the quick brown fox jumps over the lazy dog."**
2. **Tokenization:**
    - **Definition: Break the text into individual words or tokens.**
    - **Example:**
        - **Text: "The quick brown fox jumps over the lazy dog."**
        - **Tokens: ["The", "quick", "brown", "fox", "jumps", "over", "the", "lazy", "dog."]**
3. **Removing Punctuation and Special Characters:**
    - **Definition: Remove punctuation marks and special characters from the text.**
    - **Example:**
        - **Text: "The quick brown fox jumps over the lazy dog."**
        - **After Removing Punctuation: "The quick brown fox jumps over the lazy dog"**
4. **Removing Stopwords:**

- Definition: Remove common words (stopwords) that do not carry significant meaning or contribute to the analysis.
- Example:
  - Original Text: "The quick brown fox jumps over the lazy dog."
  - After Removing Stopwords: "quick brown fox jumps lazy dog."

5. **Stemming or Lemmatization:**
   - Definition: Reduce words to their base or root form to normalize the text and reduce dimensionality.
   - Example:
     - Original Text: "running, ran, runs"
     - After Stemming: "run, run, run"
     - After Lemmatization: "run, run, run"
6. **Handling Numerical Values:**
   - Definition: Convert numerical values to a standard format or remove them if they do not contribute to the analysis.
   - Example:
     - Text: "The price of the product is $10.99."
     - After Handling Numerical Values: "The price of the product is."
7. **Handling HTML Tags and URLs:**
   - Definition: Remove HTML tags and URLs from the text as they do not contribute to the analysis.
   - Example:
     - Text: "<p>This is an example <a href="https://www.example.com">website</a>.</p>"
     - After Handling HTML Tags and URLs: "This is an example website."
8. **Spell Checking (Optional):**
   - Definition: Correct misspelled words to improve the accuracy of the analysis.
   - Example:
     - Text: "Ths is an example of text."
     - After Spell Checking: "This is an example of text."
9. **Text Vectorization:**
   - Definition: Convert text into numerical vectors for machine learning algorithms to process.
   - Example:
     - Text: "The quick brown fox jumps over the lazy dog."
     - After Text Vectorization: [1, 1, 1, 1, 1, 1, 1, 1, 1]