Experiment : 1

Date _____

Page No. _____

Aim : To find the mean, median, mode and standard deviation, using their characteristic draw a box plot.

Software used : Google colab, python libraries

Theory :

→ Central Tendency : The measure of central tendency is a single value that attempts to describe a set of data by identifying the center pos^n within the set of data.
The mean, median, mode are all valid measures of central tendency.

→ Mean : It is the sum of all values in a dataset divided by the no. of values in the dataset.

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

→ Median : The median is the middle score for a set of data that has been arranged in order of magnitude.

CLASSTIME

Experiment : Date

Page No.

$$\text{median} = \begin{cases} \mu\left[\frac{n}{2}\right], & n = \text{odd} \\ \mu\left[\frac{\frac{n}{2} + \frac{(n+1)}{2}}{2}\right], & n = \text{even} \end{cases}$$

→ Mode: Most frequent score in a dataset of anything. In a histogram, the highest box is mode.

→ Variance: It is the measure of how far the set of data are dispersed out from the mean or avg. value (σ^2).

→ Standard deviation: It also represents the dispersion from mean is the square root of variance.

BOX PLOT

→ Min. score: lowest score excluding outlier
→ First Quartile (Q_1): 25% of score fall below the lower quartile.
→ Second Quartile (Q_2): Middle score of set of data by arranging in ascending order. It is also the median.
→ Third Quartile (Q_3): Around 75% data is below this point
→ Max score: The highest score excluding the outlier.

$[Q_1 - 1.5 * IQR, Q_3 + 1.5 * IQR]$ → range b/w Q_1 & Q_3.

CLASSTIME

Experiment : 2

Date

Page No.

Aim : Univariate, Bivariate, Multivariate analysis using histogram, violin plot, bar graph, scatter plot, heatmap.

Software used : Python compiler, google colab

Theory :

→ Univariable data : Data type consist of only 1 variable.

→ Bivariate data : It involves 2 different variables.

→ Multivariate data : It involves 3 or more variables.

→ Matplotlib : It is a comprehensive python library for creating static, animated, interactive visualisations in python.

→ Histogram : It provides a visual representation of grouped freq. distribution with continuous classes.

plt.hist(data['loan_amount_term'])

→ Violin plot : With the addition of a rotated kernel density plot on each side.

plt.violinplot(data['LoanApplicantIncome'])

CLASSTIME

Experiment :

Date

Page No.

→ Barplot : Bar chart represents category of data with rectangular bars with length and height is proportional to the values which they represent.

plt.bar(data.index, data)

→ Scatterplot : Diagram where each value in the dataset is represented by a dot.

plt.scatter(data['ApplicantIncome'], data['loan_amount']).

→ Seaborne : is a python data visualisation library based on matplotlib that enables statistical graphics in python.

→ Countplot : It is used to represent the occurance plot counts of the observations present in the categorical variable.

plt.countplot(x='education', hue=loanstatus, data=data)

→ Boxplot : also known as whisker plot, is used to display summary data values having properties like min, max, Q_1, Q_3, median and outliers.

plt.boxplot(data['ApplicantIncome'])

CLASSTIME

Experiment :

Date

Page No.

→ Heatmap : Matplotlib heatmap represents magnitude of phenomenon to represent data distribution by differences in datas.

0 : neutral
-ve : +ve relation
+ve : -ve relation
corr = df.corr()
sns.heatmap(corr, annot=true)

Observations and conclusions :

We have successfully plotted various plots of distinguishing visual representation of univariate, bivariate and multivariate analysis.

CLASSTIME

Experiment : 3

Date 29/01/24

Page No.

Aim: Perform EDA on a database, addressing missing values, to enhance modelling readiness.

Software used: Google colab

Theory:
Exploratory data analysis is a crucial step in the data analysis process that involves summary & visualising key characteristics of the data to understand its underlying structure.

Objective of EDA-
→ Understand the data
→ Explore relationship
→ Detect patterns
→ Prepare for modelling

- Handelling missing values
Missing values are a common challenge in a real world dataset. Dealing with missing value/data is essential to ensure the accuracy & reliability of any subsequent data analysis of modelling, several strategies can be employed to address missing value.

CLASSTIME

Experiment :

Date
Page No.

Common Strategies :

1) Removal of rows / columns :
PROS: Simple & straight-forward
CONS: May lead to loss of valuable info.
→ Suitable when missing values are random & not influential.

2) Imputation :
• Mean / Median / Mode imputation :
Filling missing values with the mean, median or mode of the variables.
PROS: Preserves the overall structure of data
CONS: May introduce bias, especially if missing data is not random.
• Interpolation or predictive imputation :
Predicting missing values based on the relationship with other variables.
PROS: Can capture more complex.
CONS: Requires a model & assumptions about relationship.

3) Advanced Techniques :
Machine learning algo. such as K-nearest neighbour or multiple imputations can be used for more sophisticated imputation strategies.

CLASSTIME

Experiment : 4

Date 05/02/24

Page No.

Aim : To understand & implement diverse data transformation techniques in Python, including Z-score, min-max scaling, mean normalisation, max absolute scaling and robust scaling.

Objectives:

→ Explore the theoretical concepts behind various data transformation techniques.

→ Implement these techniques using python & analyse their effects in data distribution.

→ Understand the importance of pre-processing in enhancing the performance of machine learning models.

Software used: Google Colab.

Theory:

Data transformation techniques:

1) Z-score : Z-score transforms the data to have a mean of 0 and a standard deviation of 1.

$$z = \frac{x - \mu}{\sigma}$$

, x = original value

μ = mean

σ = standard deviation

CLASSTIME

Experiment :

Date

Page No.

Impact:
→ centers the data around 0.
→ scales data to comparable range.

2) Min-max scaling: It transforms the data to a specific range, typically [0,1].

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Impact:
→ scales data to a specific range.
→ maintains the relative relationships b/w data points.

3) Mean normalisation: It transforms the data to have a mean of 0 and a range b/w -1 and 1. It is useful when the data distribution is skewed.

$$x_{new} = \frac{x - \mu}{x_{max} - x_{min}}$$

Impact:
→ centers the data around 0.
→ useful for data with a skewed distribution.

CLASSTIME

Experiment :

Date

Page No.

4) Max absolute scaling: It scales the data by dividing each value by the max. absolute value of the feature.

$$x_{new} = x / x_{max}.$$

Impact:
→ scales the data based on max. values.
→ preserves the sign of the data.

5) Robust scaling: It is resistant to outliers and scales the data based on the interquartile range (IQR).

$$x_{new} = \frac{x - x_{median}}{IQR \text{ or } (Q_3 - Q_1)}$$

Conclusion:
In this experiment, we have successfully explored & implemented diverse data transformation techniques.

Viva

CLASSTIME

Experiment : 5

Date 12/02/24

Page No.

Aim: To explore and implement the following distances & similarity techniques:

i) Euclidean distance
ii) Manhattan distance
iii) Co-sine similarity
iv) Minkowski distance

Software used: Google colab, python

Theory:

Distance metrics play a crucial role in various data science tasks, including clustering, classification and similarity search.

Understanding and implementing different distance metric enable data scientists to quantify the similarity or dissimilarity b/w data points, thereby facilitating effective analysis and decision making.

i) Euclidean distance:

- Straight-line distance b/w points in Euclidean space.
- $d(p,q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$

ii) Manhattan distance:

CLASSTIME

Experiment : Date

Page No.

- Defⁿ: $d(p,q) = \sum_{i=1}^{n} |p_i - q_i|$
- Sum of absolute differences b/w coordinates

iii) Cosine similarity:

- $\text{cosine-sim}(p,q) = \frac{p \cdot q}{\|p\| \|q\|}$
- measures the cosine of the ∠ b/w vectors

iv) Minkowski distance:

- $d(p,q) = \left(\sum_{i=1}^{n} |p_i - q_i|^p\right)^{1/p}$
- generalised metric including euclidean and Manhattan distances.

Conclusion: We have understood and implemented euclidean dist., manhatten dist., minkowski dist., and cosine similarity.

Jayita
4/3/24

CLASSTIME

Experiment : 6

Date 19/02/24

Page No.

Aim : Implementation of Apriori algorithm

Objective : To implement the apriori algo. using 2 different approaches.

a) Utilizing 'mlextend' library to apply the apriori using func".

b) Manually implementing the apriori from scratch.

Software used : Google colab

Theory :

The apriori is a classic algo. used for frequent itemset mining in association rule learning over transactional database. It aims to find frequent itemsets in a dataset and generate association rules based on these frequent itemsets. It operates in 2 main steps:

i) Generate candidate itemsets

Initially, it scans the database to find frequent items. Then, it generates candidate itemsets of length (K+1) by joining frequent itemsets of length K.

CLASSTIME

Experiment :

Date

Page No.

ii) Prune infrequent itemsets
After generating candidate itemsets, apriori prunes the candidates that contain any subset which is infrequent. This is based on the 'Apriori property', which states that if an itemset is infrequent all its supersets will also be infrequent.

Implementation

a) Using mlxtend library:

→ The mlxtend library provides an easy to use implementation of the apriori func^n algo.
→ We can implementation the apriori func^n from the library & use it directly on our dataset to find frequent itemsets and generate association rules.

b) Manual implementation
→ Initialize : Initialize frequent itemsets of length (single items) by counting their occurances in the dataset.
→ Generate candidates : Use the frequent itemsets of length K to generate candidate itemsets of length (K+1) by joining them

CLASSTIME

Experiment :

Date

Page No.

→ Prune candidates : Remove candidates that contain subsets of K length that are infrequent.

→ Repeat steps ② & ③ until no more frequent itemsets can be found.

→ Generate association rule : Once frequent itemsets are found, generate association rules based on these frequent itemsets.

Conclusion : We have successfully implemented apriori algo. using both the approaches.

CLASSTIME

Experiment : 7

Date 04/03/24

Page No.

Aim: Implementation of FP Growth algo.

Objective: We have to implement the FP growth algo. using:
a) 'mlxtend' library
b) mathematical programs w/o modules

Theory:

FP Growth: Its a popular method for frequent itemset mining in transactional databases. It constructs a compact data structure called FP tree to represent the database and efficiently mine frequent itemsets. The algo. involved foll. steps:

i) Construct FP tree:
Initially, it scans the transactional DB to construct the FP-tree, a compact representation of frequent itemsets.

ii) Mine frequent itemset:
After constructing the FP tree, it recursively mine frequent itemsets by performing condition pattern base construction and generating conditional FP tree.

CLASSTIME

Experiment : Date

Page No.

Implementation :

a) Using 'mlxtend' library
- import mlxtend
- import 'fp growth' func^n and apply it to the given dataset.

b) Manual implementation

Algorithm :

1) Construct FP-tree : Sort transactions, build tree
2) Mine frequent itemsets : Traverse tree, output frequent itemsets.
3) Insert transaction in tree :
   a. If transaction empty, return.
   b. Get first item, create child if not exist.
   c. Increment count, remove item, recurse.
4) Mine frequent itemsets
   a. If leaf node, return
   b. For each child, calculate support.
   c. If support >= min_support, output and recurse.

Conclusion : We have successfully implemented the FP growth algo. using both methods.

CLASSTIME

Experiment : 8

Date 01/04/23

Page No. 1

Aim : i) Perform logistic regression on diabetes dataset.

ii) Plot the confusion matrix in the heatmap form from the model generated and print the accuracy, precision, recall and F-1 score.

Software used : Google colab

Theory :

→ Logistic regression is a supervised machine learning algo. used for classification tasks where the goal is to predict the prob. that an instance belongs to a given class or not.
Logistic regression is a statistical algo. which analyse the relationship b/w 2 data factors.

→ Logistic func^n - Sigmoid func^n

It is used to map the predicted values to probabilities. It maps any real value into another value within a range of 0 and 1. The value of the logistic regression must be b/w 0 & 1, which

CLASSTIME

Experiment :

Date

Page No.

cannot go beyond this limit, so it forms a curve like the 's' form. It is called the sigmoid func^n or the logistic func^n.

→ Types of logistic regression:

1) Binomial: there can be only 2 possible types of the dependent variable, such as 0 or 1, pass or fail, etc.

2) Multinomial: there can be 3 or more possible unordered types of the dependent variable, such as 'low', 'high' and 'medium'.

3) Ordinal: there can be 3 or more possible ordered types of dependent variables such as 'low', 'high' or 'medium'.

→ Confusion matrix: It is a matrix that summarizes the performance of a machine learning model on a set of test data. It is a means of displaying the no. of accurate and inaccurate instances based on the model's predictions.

Experiment :

Date _____

Page No. _____

- True positives (TP) : occurs when the model accurately predicts a +ve datapoint.
- True negatives (TN) : occurs when the model accurately predicts a -ve data point.
- False positives (FP) : occurs when the model predicts a +ve data point incorrectly.
- False negatives (FN) : occurs when the model mispredicts a -ve data point.

PREDICTED \ ACTUAL		
	True positive	False positive
	False negative	True negative

→ Metrics based on confusion matrix data:

1) Accuracy : It is used to measure the performance of the model. It is the ratio of total correct instances to the total instances.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

2) Precision : It is a measure of how accurate a model's +ve predictions are.

$$\text{Precision} = \frac{TP}{TP+FP}$$

CLASSTIME

Experiment :

Date

Page No.

3) Recall : It measures the effectiveness of a classification model in identifying all relevant instances from a dataset.

$$\text{Recall} = \frac{TP}{TP+FN}$$

4) F-1 score : It is used to evaluate the overall performance of a classification model. It is the harmonic mean of precision and recall.

$$\text{F-1 score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Conclusion : Hence, we implemented logistic regression on diabetes dataset and plotted the corresponding confusion matrix along with accuracy, precision, recall & F-1 score.

CLASSTIME

Experiment : 9

Date 08/04/24

Page No.

Aim : To compare various boosting methods
i) Gradient Boosting
ii) XGboost
iii) Adaboost
iv) CAT Boost

Software used : Google colab.

Theory :

→ Boosting is an ensemble modeling technique that attempts to build a strong classifier from the no. of weak classifiers.

• Advantages :
1) Improved accuracy
2) Robustness to overfitting
3) Better handling of imbalanced data
4) Better interpretability.

→ Gradient boosting - It is a boosting technique that builds a final model from the sum of several weak learning algorithms that were trained on the same dataset.

CLASSTIME

Experiment :

Date

Page No.

→ XGboost – The full form of XGboost is eXtreme Gradient Boosting algo. which is an extreme variation of previous technique.
XGBoost applies a regularisation approach and outperforms a standard gradient boosting method.

→ Adaboost – It is a boosting algo. that also works on the principle of stagewise addition method where multiple weak learners are used for getting strong learner.
The value of alpha parameter, in this case, will be indirectly proportional to error of the weak learner.

→ CatBoost – The growth of decision trees include CatBoost is primary distinction that sets it apart from & improves upon competitors.

Conclusion : We have successfully compared various boosting methods :
i) Gradient boosting ii) XGBoost iii) Adaboost iv) CATBoost.
In all above methods, CATBoost performs the best.

Harsh 13/4/24

CLASSTIME

Experiment : 10

Date 15/04/24

Page No.

Aim : i) Perform classification using a Naive Bayes classifier on the given dataset.

ii) Perform Regression using a regression tree on the given dataset.

Software used : Google colab

Theory :

i) Naive Bayes Classification (Diabetes dataset) : It is a probabilistic classifier based on Baye's theorem, which is expressed as :

$$P(C_K/X) = \frac{P(X/C_K) \cdot P(C_K)}{P(X)}$$

where, $P(C_K/X)$ = posterior prob. of class C_K given features X.

$P(X/C_K)$ = likelihood of features X given class C_K.

$P(C_K)$ = prior probability of class C_K

$P(X)$ = probability of features X.

These classifiers are simple, yet effective for classification tasks, especially when the features are assumed to be independent given the class.

CLASSTIME

Experiment :

Date

Page No.

Its called 'naive' because it assumes that the presence of a particular feature in a class is unrelated to the presence of any other features.

2) Regression tree (Walmart Sales dataset):
Regression tree is a decision tree based model used for regression tasks, where the target variable is continuous. Each internal node in the tree represents a feature (or an attribute), each branch represents a decision rule, each leaf node represents the outcome.

→ In regression trees, the splitting of nodes is done based on a criterion that minimizes the variance of the target variable within each split. The most common criteria include mean squared error (MSE), mean absolute error (MAE), etc.

→ Recursive partitioning : The process of constructing a regression tree involves recursively partitioning the data into subsets, based on the values of the

CLASSTIME

Experiment :

Date

Page No.

features. At each step, the algo. selects the feature and the split point that minimizes the chosen criterion.

→ Once the tree is constructed, each instance falling into a leaf node is assigned the mean (or median) of the target variable within that leaf node as its predicted value.

→ After construction, trees may be pruned to avoid overfitting, where nodes that provide little predictive power are removed.

CONCLUSION: We have successfully performed classification using Naive Bayes classifier and regression using regression tree on the following given datasets.

Varsha
2104

CLASSTIME

Experiment : 11

Date 22/04/24

Page No.

Aim : To apply different clustering methods for classification on given dataset.

Software used : Google Colab

Theory :

1) K-means clustering :

→ It is a popular positioning clustering algorithm that divides data points into K-clusters.

→ It minimizes the sum of squared dist. b/w data points and their respective cluster centroids.

→ The algo. iteratively assigns data points to the nearest centroid and updates centroids until it converges.

2) DB-SCAN (Density Based Spatial Clustering of Application with Noise) :

Unlike K-means, DB scan does not require the no. of clusters to be specified in advance. It groups together points that are closely packed, defining clusters as areas of high density separated

CLASSTIME

Experiment :

Date

Page No.

By areas of low density. It labels points as core, bords or noise-based on their density and proximity to other points.

⇒ KNN :

KNN is a non-parametric instance-based learning algo. used for classification & regression tasks. In clustering, KNN assigns each point to the most common class among its K nearest neighbours. It relies on the dist. metric to determine similarity b/w data points.

CONCLUSION: We have applied K-means, DB-SCAN and KNN on given unlabelled dataset.

CLASSTIME