**Python:**

Total Marks: 20

Each question 10 marks

Question: 1

Write an efficient algorithm that searches for a value target in an m x n integer matrix. This matrix has the following properties:

Integers in each row are sorted from right to left.

The first integer of each row is greater than the last integer of the previous row.

Example-:

Input: matrix = [[1,3,5,7],[10,11,16,20],[23,30,34,60]], target = 3

Output: True

Program:

public class Solution {

public int searchMatrix(ArrayList<ArrayList<Integer>> a, int b) {

int r = a.size();

int c = a.get(0).size();

int start = 0;

int end = r - 1;

// default value is last row for edge case

int biRow = r -1; // row to search column

//binary search 1st value of rows

int mid = 0;

while (start <= end) {

mid = (start + end) / 2;

if ( b >= a.get(mid).get(0) && b <= a.get(mid).get(c-1)) {

break;

}

if (b < a.get(mid).get(0)) {

end = mid-1;

} else {

start = mid+1;

}

}

biRow = mid;

//binary search column of biRow

start = 0;

end = c-1;

while (start <= end) {

mid = (start + end) / 2;

if (b == a.get(biRow).get(mid)) {

return 1;

}

if (b < a.get(biRow).get(mid)) {

end = mid - 1;

} else {

start = mid + 1;

}

}

return 0;

}

}

Question: 2

2. Write a program that takes a string as input, and counts the frequency of each word in the string, there might be repeated characters in the string. Your task is to find the highest frequency and returns the length of the highest-frequency word.

Note - You have to write at least 2 additional test cases in which your program will run successfully and provide an explanation for the same.

Example input - string = “write write write all the number from from from 1 to 100”

Example output - 5

Explanation - From the given string we can note that the most frequent words are “write” and “from” and the maximum value of both the values is “write” and its corresponding length is 5

Program:

# Find frequency of each word in a string in Python

# using dictionary.

def count(elements):

# check if each word has '.' at its last. If so then ignore '.'

if elements[-1] == '.':

elements = elements[0:len(elements) - 1]

# if there exists a key as "elements" then simply

# increase its value.

if elements in dictionary:

dictionary[elements] += 1

# if the dictionary does not have the key as "elements"

# then create a key "elements" and assign its value to 1.

else:

dictionary.update({elements: 1})

# driver input to check the program.

Sentence = " write write write all the number from from from 1 to 100"

# Declare a dictionary

dictionary = {}

# split all the word of the string.

lst = Sentence.split()

# take each word from lst and pass it to the method count.

for elements in lst:

count(elements)

# print the keys and its corresponding values.

for allKeys in dictionary:

print ("Frequency of ", allKeys, end = " ")

print (":", end = " ")

print (dictionary[allKeys], end = " ")

print()

**Machine learning:**

Question: 1

1. Imagine you have a dataset where you have different Instagram features like u sername , Caption , Hashtag , Followers , Time\_Since\_posted , and likes , now your task is to predict the number of likes and Time Since posted and the rest of the features are your input features. Now you have to build a model which can predict the number of likes and Time Since posted.

Dataset This is the Dataset You can use this dataset for this question.

## Import the necessary libraries:-

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

from sklearn.preprocessing import OneHotEncoder

import re

import matplotlib.pyplot as plt

import seaborn as sns

import plotly.express as px

from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator

import warnings

warnings.filterwarnings('ignore')

## Load the dataset using pandas:

data = pd.read\_csv("instagram\_reach.csv", encoding = 'latin1')

## Checking top 5 rows

data.head()

## Checking Rows & Columns Availabale in Dataset

data.shape

## Checking Details Information related with Dataset

data.info()

## Checking Null Values

data.isnull().sum()

## Droping Null Value

data = data.dropna()

## Checking Again Null Values whether Null Value Drop or Not

data.isnull().sum()

## Checking All Columns name present in dataset

data.columns

## checking top 2 rows of dataset

data.head(2)

# Remove unnecessary columns

data= data.drop(['Likes: 0', 'S.No'], axis=1)

## Checking All Columns name present in Dataset

data.columns

## Distribution of Impressions From Followers

plt.figure(figsize=(10, 8))

plt.style.use('fivethirtyeight')

plt.title("Distribution of Impressions From Followers")

sns.distplot(data['Followers'])

plt.show()

## Distribution of Impressions From Likes

plt.figure(figsize=(10, 8))

plt.title("Distribution of Impressions From Likes")

sns.distplot(data['Likes'])

plt.show()

## Relation between Likes and Followers

followers = data["Followers"].sum()

likes = data["Likes"].sum()

labels = ['Followers', 'Likes']

values = [followers, likes]

fig = px.pie(data, values=values, names=labels,

title='Impressions on Instagram Posts From Various Sources', hole=0.5)

fig.show()

## Plotting Word-Cloud for Hashtag Related Data

text = " ".join(i for i in data.Hashtags)

stopwords = set(STOPWORDS)

wordcloud = WordCloud(stopwords=stopwords, background\_color="white").generate(text)

plt.style.use('classic')

plt.figure( figsize=(12,10))

plt.imshow(wordcloud, interpolation='bilinear')

plt.axis("off")

plt.show()

## Plotting Scatter-plot for showing Relationship Between Likes and Followers

figure = px.scatter(data\_frame = data, x="Likes",

y="Followers", trendline="ols",

title = "Relationship Between Likes and Followers")

figure.show()

# Select the relevant features and target variables

features = ['USERNAME', 'Caption', 'Hashtags', 'Followers']

target\_likes = 'Likes'

target\_time\_since\_posted = 'Time since posted'

# Split the data into training and testing sets

X = data[features]

y\_likes = data[target\_likes]

y\_time\_since\_posted = data[target\_time\_since\_posted]

X\_train, X\_test, y\_likes\_train, y\_likes\_test, y\_time\_since\_posted\_train, y\_time\_since\_posted\_test = train\_test\_split(X, y\_likes, y\_time\_since\_posted, test\_size=0.2, random\_state=42)

# Preprocess the text features using one-hot encoding

encoder = OneHotEncoder(sparse=False, handle\_unknown='ignore')

X\_train\_encoded = encoder.fit\_transform(X\_train)

X\_test\_encoded = encoder.transform(X\_test)

Train a model to predict the number of likes:

# Train a model to predict the number of likes

likes\_model = LinearRegression()

likes\_model.fit(X\_train\_encoded, y\_likes\_train)

likes\_predictions = likes\_model.predict(X\_test\_encoded)

likes\_mse = mean\_squared\_error(y\_likes\_test, likes\_predictions)

print("Mean Squared Error (Likes):", likes\_mse)

# Preprocess the time since posted variable

def extract\_numerical\_value(time\_string):

numerical\_value = re.findall(r'\d+', time\_string)[0]

return int(numerical\_value)

y\_time\_since\_posted\_train = y\_time\_since\_posted\_train.apply(extract\_numerical\_value)

y\_time\_since\_posted\_test = y\_time\_since\_posted\_test.apply(extract\_numerical\_value)

# Train a model to predict the time since posted

time\_since\_posted\_model = LinearRegression()

time\_since\_posted\_model.fit(X\_train\_encoded, y\_time\_since\_posted\_train)

time\_since\_posted\_predictions = time\_since\_posted\_model.predict(X\_test\_encoded)

time\_since\_posted\_mse = mean\_squared\_error(y\_time\_since\_posted\_test, time\_since\_posted\_predictions)

print("Mean Squared Error (Time Since Posted):", time\_since\_posted\_mse)

Question: 2

1.Explain how you can implement ML in a real world application.

1. Facial recognition

Facial recognition is one of the more obvious applications of machine learning. People previously received name suggestions for their mobile photos and Facebook tagging, but now someone is immediately tagged and verified by comparing and analyzing patterns through facial contours. And facial recognition paired with deep learning has become highly useful in healthcare to help detect genetic diseases or track a patient’s use of medication more accurately. It’s also used to combat important social issues such as child sex trafficking or sexual exploitation of children. The list of applications and industries influenced by it is steadily on the rise.

2. Product recommendations

Do you wonder how Amazon or other retailers frequently know what you might like to purchase? Or, have they gotten it wildly wrong and you wonder how they came up with the recommendation? Thank machine learning. Targeted marketing with retail uses machine learning to group customers based on buying habits or demographic similarities, and by extrapolating what one person may want from someone else’s purchases. While some suggested purchase pairings are obvious, machine learning can get eerily accurate by finding hidden relationships in data and predicting what you want before you know you want it. If the data is incomplete, sometimes you may end up with an offbase recommendation—but don’t worry, because not buying it is another data point to learn from.

3. Email automation and spam filtering

While your inbox seems relatively boring, machine learning influences its function behind the scenes. Email automation is a direct result of successful machine learning, and one function that goes most unnoticed is spam filtering. Successful spam filtering adapts and finds patterns in email content that is undesirable. This includes data from email domains, a sender’s physical location message text and structure, and IP addresses. It also requires help from users as they mark emails when they’re mistakenly filed. With each marked email, a new data reference is added that helps with future accuracy.

4. Financial accuracy

Machine learning has created a boon for the financial industry as most systems go digital. Abundant financial transactions that can’t be monitored by human eyes are easily analyzed thanks to machine learning, which helps find fraudulent transactions. One of the newest banking features is the ability to deposit a check straight from your phone by using handwriting and image recognition to “read” checks and convert them to digital text. Credit scores and lending decisions are also powered by machine learning as it both influences a score and analyzes financial risk. Additionally, combining data analytics with artificial intelligence, machine learning, and natural language processing is changing the customer experience in banking.

5. Social media optimization

Platforms from Facebook to Instagram and Twitter are using big data and artificial intelligence to enhance their functionality and strengthen the user experience. Machine learning has become helpful in fighting inappropriate content and cyberbullying, which pose a risk to platforms in losing users and weakening brand loyalty. Processing data through deep neural networks also allows social platforms to learn their users’ preferences as they offer content suggestions and target advertising.

6. Healthcare advancement

Every day, we’re getting closer to a full transition to electronic medical records. That means healthcare information for clinicians can be enhanced with analytics and machine learning to gain insights that support better planning and patient care, improved diagnoses, and lower treatment costs. Healthcare brands such as Pfizer and Providence have begun to benefit from analytics enhanced by human and artificial intelligence. There are some processes that are better suited to leverage machine learning; machine learning integration with radiology, cardiology, and pathology, for example, is leading to earlier detection of abnormalities or heightened attention on concerning areas. In the long run, machine learning will also benefit family practitioners or internists when treating patients bedside because data trends will predict health risks like heart disease. As an example, wearables generate mass amounts of data on the wearer’s health and many use AI and machine learning to alert them or their doctors of issues to support preventative measures and respond to emergencies.

7. Mobile voice to text and predictive text

Machines are also capable of learning language in other formats. Like Siri and Cortana, voice-to-text applications learn words and language then transcribe audio into writing. Predictive text also deals with language. Simple, supervised learning trains the process to recognize and predict what common, contextual words or phrases will be used based on what’s written. Unsupervised learning goes further, adjusting predictions based on data. You may start noticing that predictive text will recommend personalized words. For instance, if you have a hobby with unique terminology that falls outside of a dictionary, predictive text will learn and suggest them instead of standard words. It’s working when autocorrect starts trying to predict them in normal conversation.

8. Predictive analytics

Predictive analytics is an area of advanced analytics that uses data to make predictions about the future. Techniques such as data mining, statistics, and modeling employ machine learning and artificial intelligence to analyze current and historical data for any patterns or anomalies that can help identify risks and opportunities, minimize the chance for human errors, and increase speed and thoroughness of analysis. With closer investigation of what happened and what could happen using data, people and organizations are becoming more proactive and forward looking. Florida International University is one example. By integrating predictive models with data analysis from Tableau, they’re communicating critical insights about academic performance before students are at risk and supporting their individual needs to help them successfully complete all courses and graduate.

Believe it or not, the list of machine learning applications will grow so it’s almost too long to count. However, the benefits and improvements to our lives—and for data analysts sitting in global organizations—that come from enhancing human knowledge with machine power will be worth it, even though it feels daunting. Embrace the benefits, practicality, and future possibilities. Learn more ways that AI and machine learning are being used in augmented analytics and to augment human decision-making through smart analytics—whether for mundane or complex tasks.

Exploratory Data Analysis (EDA) refers to the method of studying and exploring record sets to apprehend their predominant traits, discover patterns, locate outliers, and identify relationships between variables. EDA is normally carried out as a preliminary step before undertaking extra formal statistical analyses or modeling.

The Foremost Goals of EDA

1. Data Cleaning: EDA involves examining the information for errors, lacking values, and inconsistencies. It includes techniques including records imputation, managing missing statistics, and figuring out and getting rid of outliers.

2. Descriptive Statistics: EDA utilizes precise records to recognize the important tendency, variability, and distribution of variables. Measures like suggest, median, mode, preferred deviation, range, and percentiles are usually used.

3. Data Visualization: EDA employs visual techniques to represent the statistics graphically. Visualizations consisting of histograms, box plots, scatter plots, line plots, heatmaps, and bar charts assist in identifying styles, trends, and relationships within the facts.

4. Feature Engineering: EDA allows for the exploration of various variables and their adjustments to create new functions or derive meaningful insights. Feature engineering can contain scaling, normalization, binning, encoding express variables, and creating interplay or derived variables.

5. Correlation and Relationships: EDA allows discover relationships and dependencies between variables. Techniques such as correlation analysis, scatter plots, and pass-tabulations offer insights into the power and direction of relationships between variables.

6. Data Segmentation: EDA can contain dividing the information into significant segments based totally on sure standards or traits. This segmentation allows advantage insights into unique subgroups inside the information and might cause extra focused analysis.

7. Hypothesis Generation: EDA aids in generating hypotheses or studies questions based totally on the preliminary exploration of the data. It facilitates form the inspiration for in addition evaluation and model building.

8. Data Quality Assessment: EDA permits for assessing the nice and reliability of the information. It involves checking for records integrity, consistency, and accuracy to make certain the information is suitable for analysis.

Types of EDA

Depending on the number of columns we are analyzing we can divide EDA into two types.

EDA, or Exploratory Data Analysis, refers back to the method of analyzing and analyzing information units to uncover styles, pick out relationships, and gain insights. There are various sorts of EDA strategies that can be hired relying on the nature of the records and the desires of the evaluation. Here are some not unusual kinds of EDA:

1. Univariate Analysis: This sort of evaluation makes a speciality of analyzing character variables inside the records set. It involves summarizing and visualizing a unmarried variable at a time to understand its distribution, relevant tendency, unfold, and different applicable records. Techniques like histograms, field plots, bar charts, and precis information are generally used in univariate analysis.

2. Bivariate Analysis: Bivariate evaluation involves exploring the connection between variables. It enables find associations, correlations, and dependencies between pairs of variables. Scatter plots, line plots, correlation matrices, and move-tabulation are generally used strategies in bivariate analysis.

3. Multivariate Analysis: Multivariate analysis extends bivariate evaluation to encompass greater than variables. It ambitions to apprehend the complex interactions and dependencies among more than one variables in a records set. Techniques inclusive of heatmaps, parallel coordinates, aspect analysis, and primary component analysis (PCA) are used for multivariate analysis.

4. Time Series Analysis: This type of analysis is mainly applied to statistics sets that have a temporal component. Time collection evaluation entails inspecting and modeling styles, traits, and seasonality inside the statistics through the years. Techniques like line plots, autocorrelation analysis, transferring averages, and ARIMA (AutoRegressive Integrated Moving Average) fashions are generally utilized in time series analysis.

5. Missing Data Analysis: Missing information is a not unusual issue in datasets, and it may impact the reliability and validity of the evaluation. Missing statistics analysis includes figuring out missing values, know-how the patterns of missingness, and using suitable techniques to deal with missing data. Techniques along with lacking facts styles, imputation strategies, and sensitivity evaluation are employed in lacking facts evaluation.

6. Outlier Analysis: Outliers are statistics factors that drastically deviate from the general sample of the facts. Outlier analysis includes identifying and knowledge the presence of outliers, their capability reasons, and their impact at the analysis. Techniques along with box plots, scatter plots, z-rankings, and clustering algorithms are used for outlier evaluation.

7. Data Visualization: Data visualization is a critical factor of EDA that entails creating visible representations of the statistics to facilitate understanding and exploration. Various visualization techniques, inclusive of bar charts, histograms, scatter plots, line plots, heatmaps, and interactive dashboards, are used to represent exclusive kinds of statistics.

These are just a few examples of the types of EDA techniques that can be employed at some stage in information evaluation. The choice of strategies relies upon on the information traits, research questions, and the insights sought from the analysis.

Program:

#Exploratory Data Analysis (EDA) Using Python Libraries

import pandas as pd

import numpy as np

# read datasdet using pandas

df = pd.read\_csv('employees.csv')

df.head()

#Getting Insights About The Dataset

df.shape

df.describe()

# information about the dataset

df.info()

#Changing Dtype from Object to Datetime

# convert "Start Date" column to datetime data type

df['Start Date'] = pd.to\_datetime(df['Start Date'])

df.nunique()

#Handling Missing Values

df.isnull().sum()

df = df.dropna(axis = 0, how ='any')

print(df.isnull().sum())

df.shape

from sklearn.preprocessing import LabelEncoder

# create an instance of LabelEncoder

le = LabelEncoder()

# fit and transform the "Senior Management"

# column with LabelEncoder

df['size'] = le.fit\_transform\

(df['size'])

#Histogram

# importing packages

import seaborn as sns

import matplotlib.pyplot as plt

sns.histplot(x='bath', data=df, )

plt.show()

#Boxplot

# importing packages

import seaborn as sns

import matplotlib.pyplot as plt

sns.boxplot( x="society", y='Team', data=df, )

plt.show()

#Handling Outliers

# importing packages

import seaborn as sns

import matplotlib.pyplot as plt

# Load the dataset

df = pd.read\_csv('Bengaluru\_House\_Data.csv')

sns.boxplot(x='price', data=df)

2. Feature engineering

Feature engineering is the process of transforming raw data into features that are suitable for machine learning models. In other words, it is the process of selecting, extracting, and transforming the most relevant features from the available data to build more accurate and efficient machine learning models.

The success of machine learning models heavily depends on the quality of the features used to train them. Feature engineering involves a set of techniques that enable us to create new features by combining or transforming the existing ones. These techniques help to highlight the most important patterns and relationships in the data, which in turn helps the machine learning model to learn from the data more effectively.

Need for Feature Engineering in Machine Learning-

We engineer features for various reasons, and some of the main reasons include:

Improve User Experience: The primary reason we engineer features is to enhance the user experience of a product or service. By adding new features, we can make the product more intuitive, efficient, and user-friendly, which can increase user satisfaction and engagement.

Competitive Advantage: Another reason we engineer features is to gain a competitive advantage in the marketplace. By offering unique and innovative features, we can differentiate our product from competitors and attract more customers.

Meet Customer Needs: We engineer features to meet the evolving needs of customers. By analyzing user feedback, market trends, and customer behavior, we can identify areas where new features could enhance the product’s value and meet customer needs.

Increase Revenue: Features can also be engineered to generate more revenue. For example, a new feature that streamlines the checkout process can increase sales, or a feature that provides additional functionality could lead to more upsells or cross-sells.

Future-Proofing: Engineering features can also be done to future-proof a product or service. By anticipating future trends and potential customer needs, we can develop features that ensure the product remains relevant and useful in the long term.

feature engineering in EDA-

Applying feature engineering techniques during exploratory data analysis (EDA) to uncover hidden patterns, identify relationships between features, and understand the data distribution. This helps in selecting relevant features and building better models.

Steps in Feature Engineering-

The steps for feature engineering vary per different Ml engineers and data scientists. Some of the common steps that are involved in most machine-learning algorithms are:

Data Cleansing

Data cleansing (also known as data cleaning or data scrubbing) involves identifying and removing or correcting any errors or inconsistencies in the dataset. This step is important to ensure that the data is accurate and reliable.

Data Transformation

Feature Extraction

Feature Selection

Feature selection involves selecting the most relevant features from the dataset for use in machine learning. This can include techniques like correlation analysis, mutual information, and stepwise regression.

Feature Iteration

Feature iteration involves refining and improving the features based on the performance of the machine learning model. This can include techniques like adding new features, removing redundant features and transforming features in different ways.

Overall, the goal of feature engineering is to create a set of informative and relevant features that can be used to train a machine learning model and improve its accuracy and performance. The specific steps involved in the process may vary depending on the type of data and the specific machine-learning problem at hand.

Techniques Used in Feature Engineering

Feature engineering is the process of transforming raw data into features that are suitable for machine learning models. There are various techniques that can be used in feature engineering to create new features by combining or transforming the existing ones. The following are some of the commonly used feature engineering techniques:

One-Hot Encoding

One-hot encoding is a technique used to transform categorical variables into numerical values that can be used by machine learning models. In this technique, each category is transformed into a binary value indicating its presence or absence. For example, consider a categorical variable “Colour” with three categories: Red, Green, and Blue. One-hot encoding would transform this variable into three binary variables: Colour\_Red, Colour\_Green, and Colour\_Blue, where the value of each variable would be 1 if the corresponding category is present and 0 otherwise.

Binning

Binning is a technique used to transform continuous variables into categorical variables. In this technique, the range of values of the continuous variable is divided into several bins, and each bin is assigned a categorical value. For example, consider a continuous variable “Age” with values ranging from 18 to 80. Binning would divide this variable into several age groups such as 18-25, 26-35, 36-50, and 51-80, and assign a categorical value to each age group.

Scaling

The most common scaling techniques are standardization and normalization. Standardization scales the variable so that it has zero mean and unit variance. Normalization scales the variable so that it has a range of values between 0 and 1.

Feature Split

Feature splitting is a powerful technique used in feature engineering to improve the performance of machine learning models. It involves dividing single features into multiple sub-features or groups based on specific criteria. This process unlocks valuable insights and enhances the model’s ability to capture complex relationships and patterns within the data.

Text Data Preprocessing

Text data requires special preprocessing techniques before it can be used by machine learning models. Text preprocessing involves removing stop words, stemming, lemmatization, and vectorization. Stop words are common words that do not add much meaning to the text, such as “the” and “and”. Stemming involves reducing words to their root form, such as converting “running” to “run”. Lemmatization is similar to stemming, but it reduces words to their base form, such as converting “running” to “run”. Vectorization involves transforming text data into numerical vectors that can be used by machine learning models.

Feature Engineering Tools-

There are several tools available for feature engineering. Here are some popular ones:

1. Featuretools

Featuretools is a Python library that enables automatic feature engineering for structured data. It can extract features from multiple tables, including relational databases and CSV files, and generate new features based on user-defined primitives. Some of its features include:

Automated feature engineering using machine learning algorithms.

Support for handling time-dependent data.

Integration with popular Python libraries, such as pandas and scikit-learn.

Visualization tools for exploring and analyzing the generated features.

Extensive documentation and tutorials for getting started.

2. TPOT

TPOT (Tree-based Pipeline Optimization Tool) is an automated machine learning tool that includes feature engineering as one of its components. It uses genetic programming to search for the best combination of features and machine learning algorithms for a given dataset. Some of its features include:

Automatic feature selection and transformation.

Support for multiple types of machine learning models, including regression, classification, and clustering.

Ability to handle missing data and categorical variables.

Integration with popular Python libraries, such as scikit-learn and pandas.

Interactive visualization of the generated pipelines.

3. DataRobot

DataRobot is a machine learning automation platform that includes feature engineering as one of its capabilities. It uses automated machine learning techniques to generate new features and select the best combination of features and models for a given dataset. Some of its features include:

* Automatic feature engineering using machine learning algorithms.
* Support for handling time-dependent and text data.
* Integration with popular Python libraries, such as pandas and scikit-learn.
* Interactive visualization of the generated models and features.
* Collaboration tools for teams working on machine learning projects.

4. Alteryx

Alteryx is a data preparation and automation tool that includes feature engineering as one of its features. It provides a visual interface for creating data pipelines that can extract, transform, and generate features from multiple data sources. Some of its features include:

Support for handling structured and unstructured data.

Integration with popular data sources, such as Excel and databases.

Pre-built tools for feature extraction and transformation.

Support for custom scripting and code integration.

Collaboration and sharing tools for teams working on data projects.

5. H2O.ai

H2O.ai is an open-source machine learning platform that includes feature engineering as one of its capabilities. It provides a range of automated feature engineering techniques, such as feature scaling, imputation, and encoding, as well as manual feature engineering capabilities for more advanced users. Some of its features include:

Automatic and manual feature engineering options.

Support for structured and unstructured data, including text and image data.

Integration with popular data sources, such as CSV files and databases.

Interactive visualization of the generated features and models.

Collaboration and sharing tools for teams working on machine learning projects.

Program:

import pandas as pd

import numpy as np

# Complete Case Analysis for Missing Data Imputation

import numpy as np

import pandas as pd

titanic = pd.read\_csv('titanic/train.csv')

# make a copy of titanic dataset

data1 = titanic.copy()

data1.isnull().mean()

#Mean/ Median/ Mode for Missing Data Imputation

impute missing values in age in train and test set

median = X\_train.Age.median()

for df in [X\_train, X\_test]:

df['Age'].fillna(median, inplace=True)

X\_train['Age'].isnull().sum()

Output:  
0

#Missing Value Indicator For Missing Value Indication

X\_train['Age\_NA'] = np.where(X\_train['Age'].isnull(), 1, 0)

X\_test['Age\_NA'] = np.where(X\_test['Age'].isnull(), 1, 0)

X\_train.head()

**Deep Learning :**

Question: 2

1.Explain how you can implement DL in a real-world application.

Here are ten ways deep learning is already being used in diverse industries.

1. Computer vision

High-end gamers interact with deep learning modules on a very frequent basis. Deep neural networks power bleeding-edge object detection, image classification, image restoration, and image segmentation. So much so, they even power the recognition of hand-written digits on a computer system. To wit, deep learning is riding on an extraordinary neural network to empower machines to replicate the mechanism of the human visual agency.

2. Sentiment based news aggregation

Carolyn Gregorie writes in her Huffington Post piece: “the world isn’t falling apart, but it can sure feel like it.” And we couldn’t agree more. I am not naming names here, but you cannot scroll down any of your social media feed without stumbling across a couple of global disasters – with the exception of Instagram perhaps.

News aggregators are now using deep learning modules to filter out negative news and show you only the positive stuff happening around. This is especially helpful given how blatantly sensationalist a section of our media has been of late.

3. Bots based on deep learning

Take a moment to digest this – Nvidia researchers have developed an AI system that helps robots learn from human demonstrative actions. Housekeeping robots that perform actions based on artificial intelligence inputs from several sources are rather common. Like human brains process actions based on past experiences and sensory inputs, deep-learning infrastructures help robots execute tasks depending on varying AI opinions.

4. Automated translations

Automated translations did exist before the addition of deep learning. But deep learning is helping machines make enhanced translations with the guaranteed accuracy that was missing in the past. Plus, deep learning also helps in translation derived from images – something totally new that could not have been possible using traditional text-based interpretation.

5. Customer experience

Many businesses already make use of machine learning to work on customer experience. Viable examples include online self-service platforms. Plus, many organizations now depend on deep learning to create reliable workflows. Most of us are already familiar with the use of chatbots by organizations. As this application of deep leering matures, we can expect to see further enhancements in this field.

6. Autonomous vehicles

The next time you are lucky enough to witness an autonomous vehicle driving down, understand that there are several AI models working simultaneously. While some models pin-point pedestrians, others are adept at identifying street signs. A single car can be informed by millions of AI models while driving down the road. Many have considered AI-powered car drives safer than human riding.

7. Coloring illustrations

At one point, adding colors to black and white videos used to be one of the most time-consuming jobs in media production. But thanks to deep learning models and artificial intelligence, adding color to b/w photos and videos is now easier than ever. As you read, hundreds of black and white illustrations are being recreated in colored form.

8. Image analysis and caption generation

One of the greatest feats of deep learning is the ability to identify images and generate intelligent captions for them. In fact, image caption generation powered by AI is so accurate that many online publications are already making use of such techniques to save time and cost.

9. Text generation

Machines now have the power to generate new text from the scratch. They can learn the punctuation, grammar, and style of a piece of text and pen down effective news pieces. Robo-journalists riding on deep learning models have been producing accurate match reports for at least three years now. And the skill isn’t limited to match report writing exclusively.

AI-based text generation is fully equipped to handle the complexity of opinion pieces on issues concerning you and myself. As of now, text generation has helped create entries on just about everything from children’s rhymes to scholarly topics.

10. Language identification

At this point, we are looking at a preliminary stage where deep learning machines can differentiate between different dialects. For example, a machine will make the decision that someone is speaking in English. It will then make a distinction based on the dialect. Once the dialect has been established, further processing will be handled by another AI that specializes in the particular language. Not to mention, there is no human intervention in any of these steps.

These were just a few applications of deep learning that exist already. The further growth of deep learning models will bring to us many more uses of artificial intelligence around us. At Futran Solutions, we work with top-of-the-line AI resources that make the above industry applications of AI come to life. Contact us today to find out more about our RPA, AI, and deep learning solutions.