# Data Processing : Assignment-1 Presentation
# Predicting student performance analysis


Module Leader :
Dr. Pushphavathi T P
Department of CSE
FET
MSRUAS

Presented By :
Brindashree B V
20ETCS115003
MTech in
DataScience
MSRUAS


# Contents/overview of the presentation

- Introduction
- About Dataset
- Pre-Processing Techniques
- Machine Learning model- Random forest classification
- Implementation
- Results and Conclusion

# Introduction

- **Data science** is an inter-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from structured and unstructured data, and apply knowledge and actionable insights from data across a broad range of application domains.

- Data science is related to data mining, machine learning and big data. Data science is a "concept to unify statistics, data analysis, informatics, and their related methods" in order to "understand and analyze actual phenomena" with data.

- **Machine learning (ML)** is the study of computer algorithms that improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence.

- Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so.

# About the dataset

| Column Name | Description |
|---|---|
| gender | Male/ Female |
| race/ethnicity | Group division from A to E |
| parental level of education | Details of parental education varying from high school to master's degree |
| lunch | Type of lunch selected |
| test preparation course | Course details |
| DP score | Marks secured by a student in DP |
| reading score | Marks secured by a student in Reading |
| writing score | Marks secured by a student in Writing |

**Student Performance Analysis**

# Pre-Processing Steps

Step 1: Collected the dataset

Step 2: Describing the data

Step 3: Checking for missing values.

Step 4: Applying pre processing techniques

Data processing Assignment 1 - Jupyter Notebook

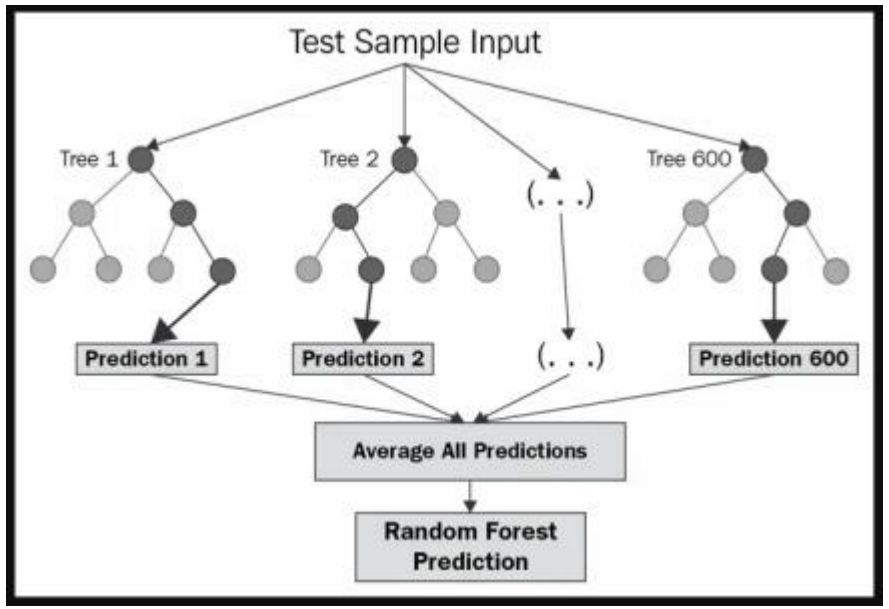
Cleaning the data . Checking for null values in data

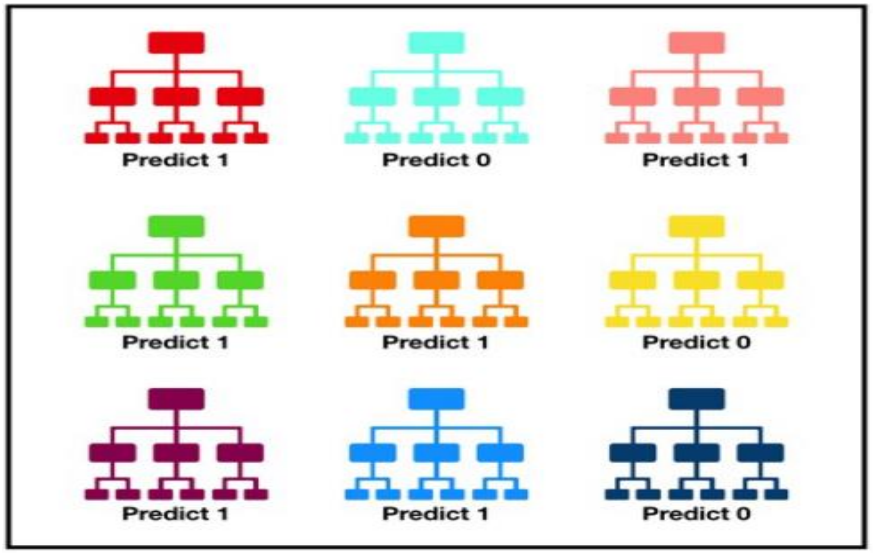
```
In [4]: df.isnull().sum()

Out[4]: gender                         0
        race/ethnicity                 0
        parental level of education    0
        lunch                          0
        test preparation course        0
        DP score                       0
        reading score                  0
        writing score                  0
        dtype: int64
```

# Machine Learning model- Random forest classification

- **Random forests** or **random decision forests** are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean/average prediction (regression) of the individual trees.

- Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.



# Implementation

The process of implementation involved:

1. Data was pre-processed and cleaned.
2. Exploratory Data analysis for the given dataset.
3. Data splitting – Train data and Test data
4. Model building using machine learning algorithm i.e Random forest classification.
5. Training the model by using the training data.
6. Testing the model and predicting the results.
7. Calculating the accuracy of the model.

Importing the Libraries

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
```

Reading the Data

```
In [2]: df = pd.read_csv(r'C:\Users\HP\Downloads\student_performance.csv')
        df
```

```
In [4]: df.isnull().sum()
```

```
Out[4]: gender                       0
        race/ethnicity               0
        parental level of education  0
        lunch                        0
        test preparation course      0
        DP score                     0
        reading score                0
        writing score                0
        dtype: int64
```

## Exploratory Data Analysis

```
In [16]: sns.set_style('darkgrid')
```

```
In [18]: # Heatmap
         sns.heatmap(df.corr(), annot = True, cmap='inferno')
         plt.show()
```


There is strong correlation between a student's reading score & writing score, reading score & DP score and writing score & DP score

```
In [37]: def getgrade(percentage, status):
           if status == 'Fail':
             return 'E'
           if(percentage >= 90):
             return 'O'
           if(percentage >= 80):
             return 'A'
           if(percentage >= 70):
             return 'B'
           if(percentage >= 60):
             return 'C'
           if(percentage >= 40):
             return 'D'
           else :
             return 'E'

         df['grades'] = df.apply(lambda x: getgrade(x['percentage'], x['status']), axis =

         df['grades'].value_counts()
```

```
Out[37]: B    260
         C    252
         D    223
         A    156
         O     58
         E     51
         Name: grades, dtype: int64
```

# Results and Conclusion

In [44]:
```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.25, rando

print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)
print(x_test.head())
```

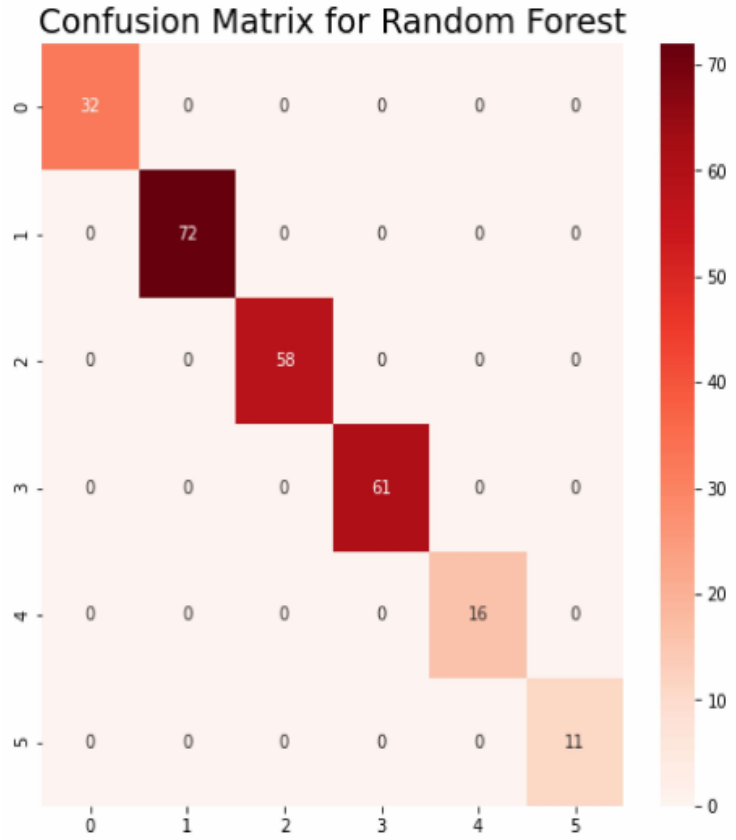
In [45]:
```
from sklearn.preprocessing import MinMaxScaler

# creating a scaler
mm = MinMaxScaler()

# feeding the independent variable into the scaler
x_train = mm.fit_transform(x_train)
x_test = mm.transform(x_test)
```

## Machine learning technique - Random Forest

In [46]:
```
from sklearn.ensemble import RandomForestClassifier

# creating a model
model = RandomForestClassifier()

# feeding the training data to the model
model.fit(x_train, y_train)

# predicting the x-test results
y_pred = model.predict(x_test)

# calculating the accuracies
print("Training Accuracy :", model.score(x_train, y_train))
print("Testing Accuracy :", model.score(x_test, y_test))
```

```
Training Accuracy : 1.0
Testing Accuracy : 1.0
```

In [47]:
```
from sklearn.metrics import confusion_matrix

# creating a confusion matrix
cm = confusion_matrix(y_test, y_pred)

# printing the confusion matrix
plt.rcParams['figure.figsize'] = (8, 8)
sns.heatmap(cm, annot = True, cmap = 'Reds')
plt.title('Confusion Matrix for Random Forest', fontweight = 30, fontsize = 20)
plt.show()
```



# Conclusion

This work bought in some of the most relevant work in performing the data analysis tasks, delivered graphical visualization for some input attributes and developed Random forest algorithm with the given Dataset. We conclude that it is important to choose a classification model that is suitable for various types and complexity of the dataset. Consequently , the topic of predicting students performance with data processing technique has inspired for working more with such methods.

Thank you