S.I.E.S. COLLEGE OF ARTS, SCIENCE AND COMMERCE

(AUTONOMOUS)

DEPARTMENT OF STATISTICS

SION (W), MUMBAI-400022.

"FINANCIAL LITERACY"

A PROJECT REPORT SUBMITTED TO THE DEPARTMENT OF STATISTICS OF S.I.E.S. COLLEGE OF ARTS, SCIENCE AND COMMERCE(AUTONOMOUS)

IN THE PARTIAL FULFILLMENT OF DEGREE OF BACHELOR OF SCIENCE IN STATISTICS

BY

TANVEER SINGH - TS2122108

MIHIKA LIKHITE - TS2122110

AMAAN MULLA - TS2122112

ADITH PERIGATHARA - TS2122118

INSHA SHAH - TS2122128

2021-2022

SIES
College of Arts, Science & Commerce (Autonomous)
RISE WITH EDUCATION
NAAC REACCREDITED - 'A' GRADE

DEPARTMENT OF STATISTICS

SION (W), MUMBAI-400022

CERTIFICATE

This is to certify that the project "Financial Literacy" carried out by a group of five students during the during the academic year 2021-2022

The team comprised of:

TANVEER SINGH	-	TS2122108
MIHIKA LIKHITE	-	TS2122110
AMAAN MULLA	-	TS2122112
ADITH PERIGATHARA	-	TS2122118
INSHA SHAH	-	TS2122128

This work is best of our knowledge and belief.

X

Mrs. Pallavi Rege
Head of Department, Statistics

X

Mr. Siddhant Marathe
Project Mentor

INDEX

ACKNOWLEDGEMENT

We want to extend our sincerest gratitude towards Mr. Siddhant Marathe (Mentor), Mrs. Pallavi Rege (H.O.D) and all the teachers of the Statistics Department of SIES College of Arts, Science and Commerce (Autonomous) who gave us the golden opportunity to present this project. The project gave us a deeper understanding of Statistics and Machine Learning techniques as well as helped us improve our interpersonal skills.

INTRODUCTION

What is Financial Literacy?

- It is the ability to confidently understand concepts such as budgeting, investing, saving and managing debt that results in financial well-being and self-confidence.
- The four points mentioned above are the fundamentals of Financial Literacy.
    - Budgeting - Budgets are based on four major uses of money: spending, investing, saving and donating. With the right balance across the primary uses of money, individuals can allocate their income more effectively, resulting in financial security and prosperity.
    - Investing - An individual who wishes to become financially literate must become familiar with key components of investing, such as interest rates, price levels, diversification, risk mitigation, and indexes.
    - Saving - Saving means securing one's financial future as well as the present. If you develop good saving habits, you will be able to accomplish a number of goals, such as achieving financial goals, achieving financial discipline, and establishing an emergency fund.
    - Debt - Debt is just borrowing and spending money that is not yours. Credit cards, bank loans, and other types of borrowing are all forms of debt. However, not every debt is a negative debt, and for this purpose you must first understand the difference between bad and good debt.

Why is Financial Literacy important?

- The importance of financial literacy lies in the ability to manage your money effectively.
- In the absence of such a foundation, your actions and decisions about savings and investments would be lacking.
- Financial literacy will allow you to understand financial concepts better as well as manage your finances effectively. As a result, you will be able to make more informed decisions related to your money and achieve financial stability.

Causality between Financial literacy and Economic Outcomes

A common question is whether financial literacy leads to economic behavior in a linear fashion. Financial literacy has been assessed through a study[1] that found people who are more financially literate plan better, save more, earn more on their investments, and manage their money better during retirement.

OBJECTIVES

1. To observe the impact of Socio-demographic factors on financial literacy of the population.

Socio-demographic factors:

- Age
- Gender
- Marital Status
- Working Status
- Modes of Income
- Income Range
- Working domain
- Education

2. Check for relationship between personal investment score (count of investment practices employed) and financial knowledge.

3. Assess and compare the contribution of family and peers on financial knowledge and investment practices.

4. Compare financial literacy between students and working professionals.

RESEARCH METHODOLOGY

- ESTABLISHING OBJECTIVES
  The first step of the project was to establish Objectives to be achieved through the project.

- PREPARING THE QUESTIONNAIRE
  We designed a questionnaire on google forms which covered all aspects of our objectives. To assess the financial literacy, a quiz designed by the Symbiosis Statistical Institute. We edited the survey according to our objectives with the help of a finance expert, Mr. Alan Aruldas. The quiz consisted of four sections: General Personal Finance, Savings and Borrowings, Insurance and Investment. Each question had a maximum of 1 mark for the correct answer. There were total of 25 questions in the quiz.

- DATA COLLECTION
  The prepared questionnaire was circulated and the method used for data collection was 'Snowball Sampling'. The population region chosen was Mumbai City, Mumbai Surburban, Navi Mumbai-Panvel and Thane.

- SAMPLE SIZE
  Using Cochran's formula for sample size, at 95% confidence interval and 5% marginal error, we required 206 data points at the least.

- DATA CLEANING
  A total of 275 people responded to our form. After cleaning the data for inconsistent entries (checked through adding Red Herring questions) and outliers, our sample size was reduced to 256 data points.

- DATA MANIPULATION
  Excel was used for data manipulation. Data manipulation refers to the process of adjusting the data to make it organised and easier to read. The data was divided into two sections: Working class and Student Class. The original data was still retained for the purpose of analysis.

- EXPLORATORY DATA ANALYSIS
  Python was used for the purpose of EDA. Graphs like histogram, bar plots, correlation plots, box plots, etc.

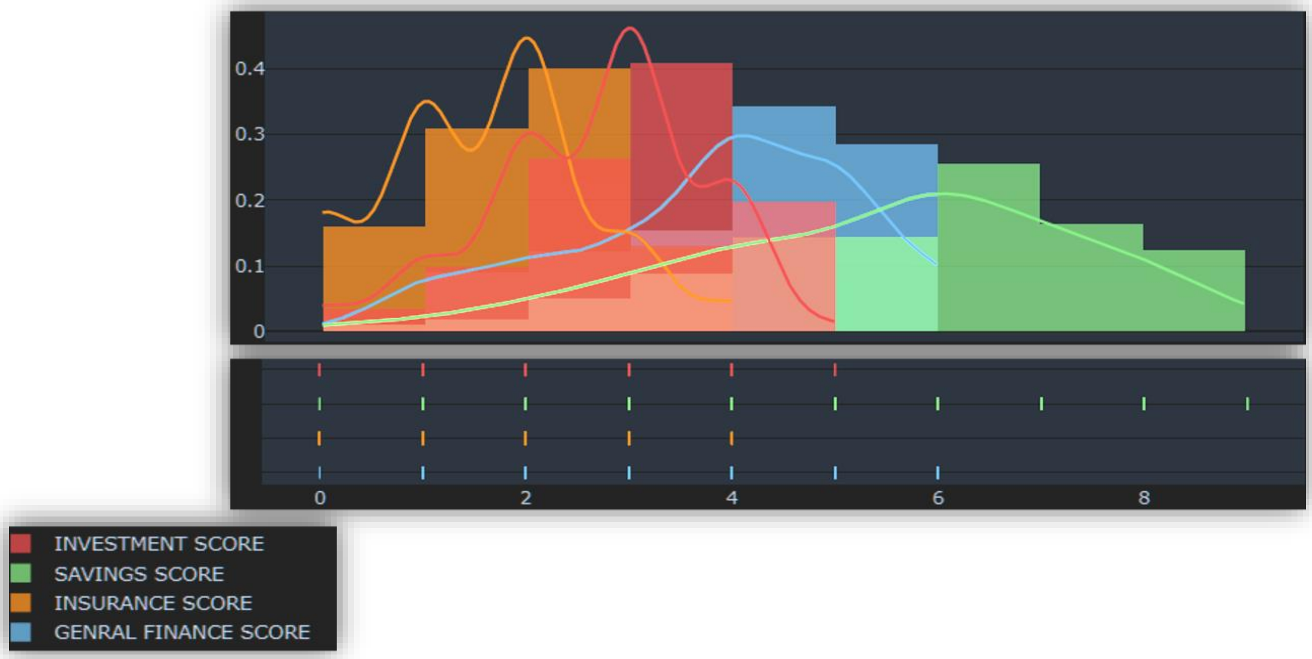
- HYPOTHESIS TESTING
  Python and R software were used for testing of hypothesis.

- LOGISTIC REGRESSION
  Python was used to build Logistic Regression model.

- RANDOM FOREST
  R software was used to build Random forests model.

EXPLORATORY DATA ANALYSIS

1. HISTOGRAM OF SCORES



The histogram of section-wise quiz scores suggests that:

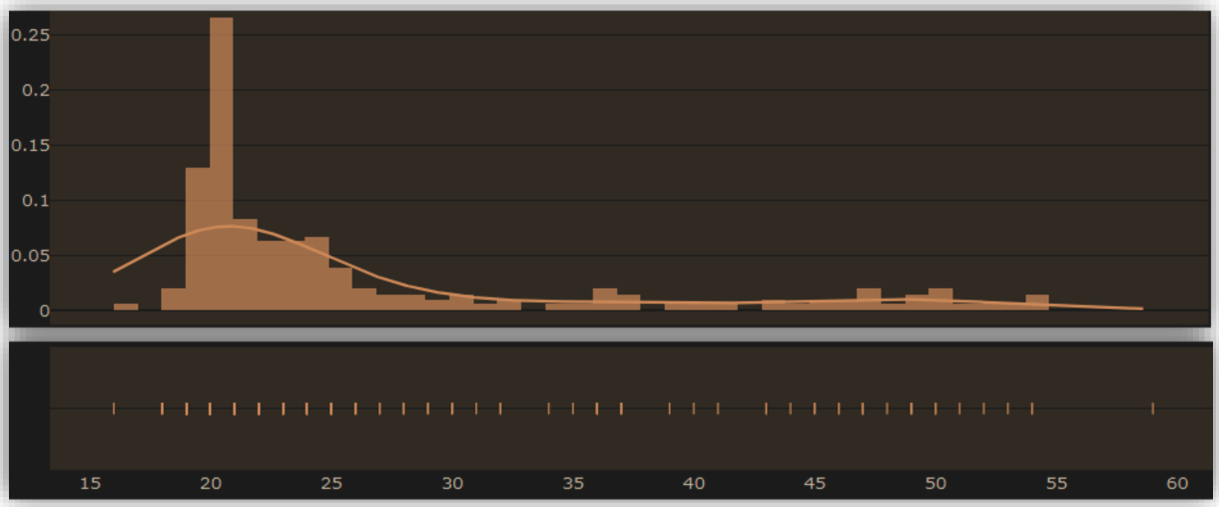
- The Investment score (red) is symmetric.
- The Saving score (green) is positively skewed suggesting the population is generally knowledgeable above Savings and borrowings.
- The General Personal Finance score (blue) is positively skewed suggesting the population is generally knowledgeable above General personal finance.
- The Insurance score (blue) is positively skewed but the kurtosis of the same is platykurtic suggesting the probability of the population being knowledgeable about Insurance is low.

The average total score(addition of the four scores mentioned above) of the population is 13.41.

The median total score of the population is 14.

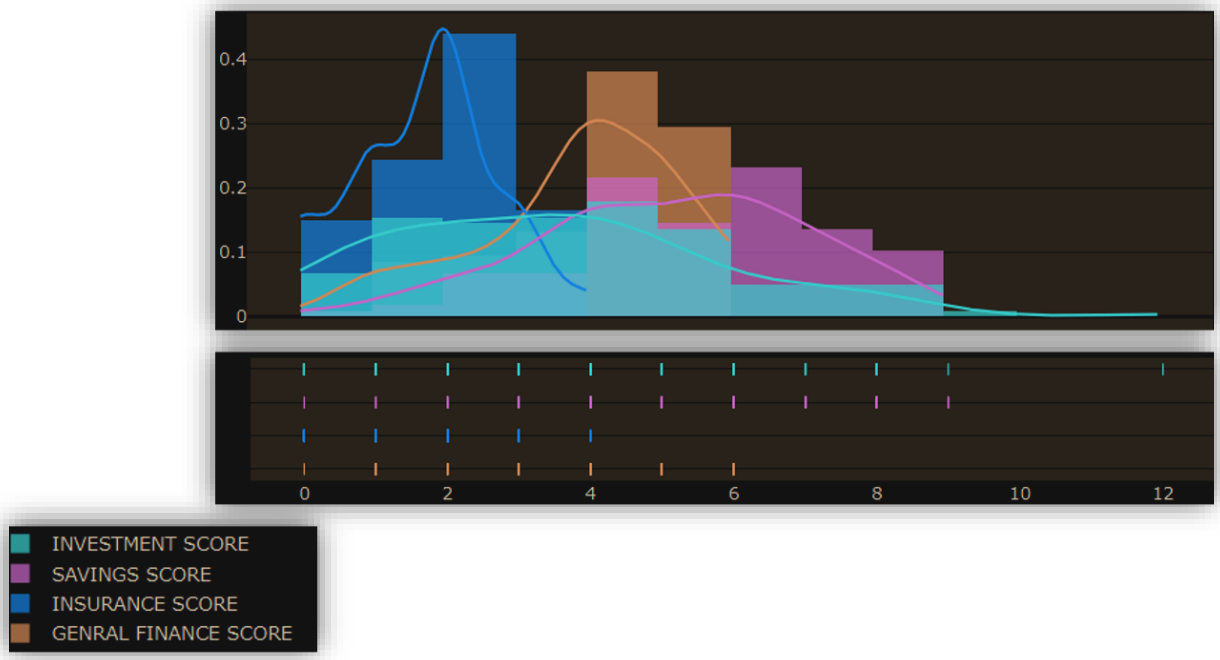
The total score ranges from 4-22.

2. HISTOGRAM OF AGE



- The histogram of Age shows that most of the population consists of the youth (17-30)

3. HISTOGRAM OF STUDENT FINANCIAL LITERACY SCORES

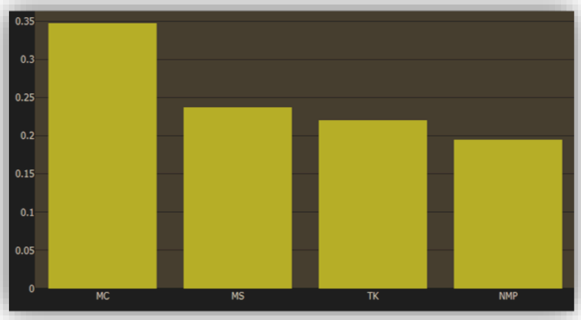


The histogram of section-wise quiz scores of **student class** suggests that:

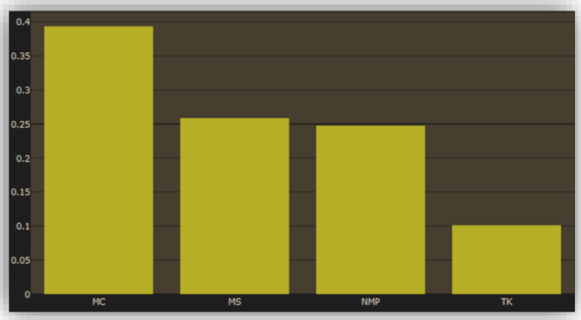
- The Investment score (dark blue) is symmetric.
- The Saving score (purple) is positively skewed suggesting the student population is generally knowledgeable above Savings and borrowings.
- The General Personal Finance score (brown) is slightly positively skewed suggesting the student population is generally knowledgeable above General personal finance.
- The Insurance score (blue) is symmetric but the kurtosis of the same is platykurtic suggesting the probability of the population being knowledgeable about Insurance is low.

4. DISTRICT

STUDENT



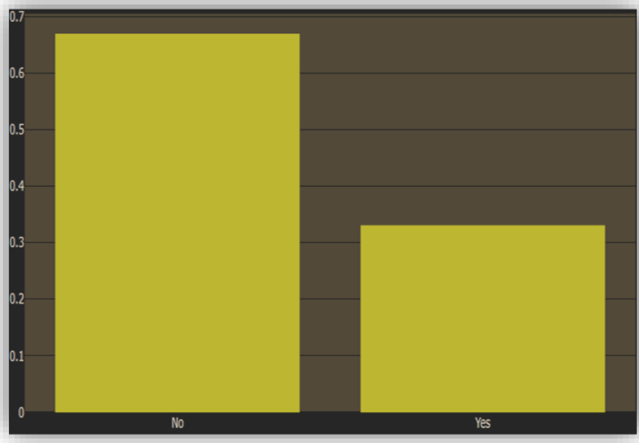
WORKING PROFESSIONAL



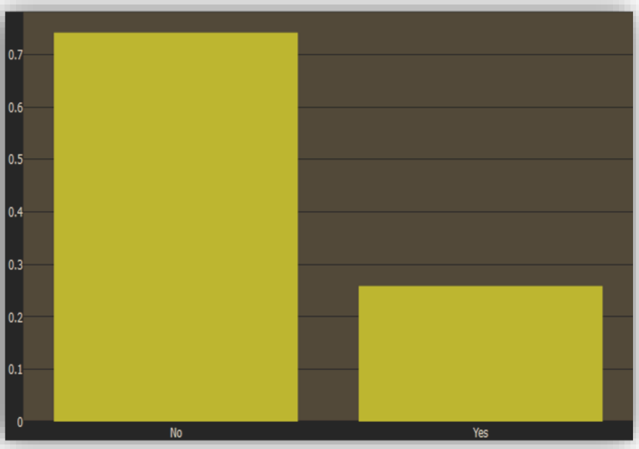
- From the graph we can observe, the student data consisted of more Navi Mumbai Panvel individuals than the working class.
- Most of the people from the population belonged to Mumbai City, followed by Mumbai-Surburb, Thane-Karjat and Navi Mumbai-Panvel

5. **QUESTION: HAVE YOU BEEN GUIDED ON MANAGING PERSONAL FINANCE?**

STUDENT



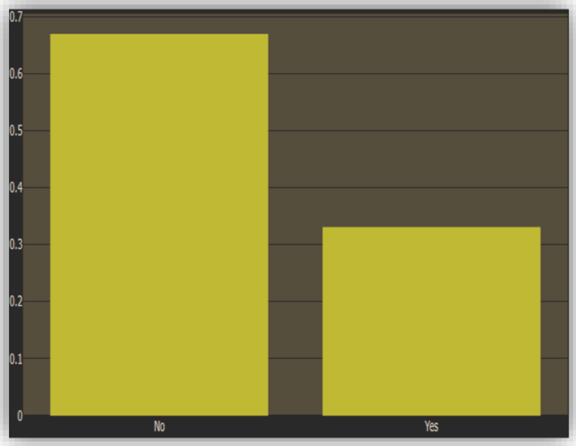
WORKING PROFESSIONAL

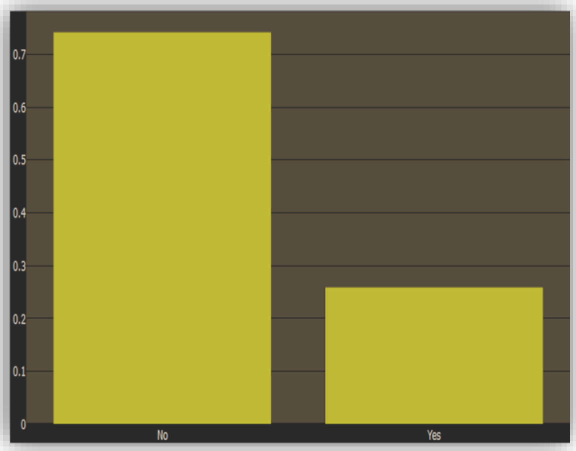


- Most of the population from both student and working professional class weren't guided about how to manage personal funds.

6. QUESTION: DO YOU KEEP EXCESS FUNDS IN CASE OF EMEGENCY?

STUDENT **WORKING PROFESSIONAL**

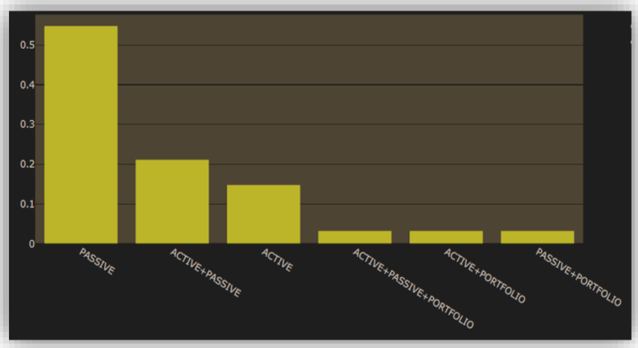




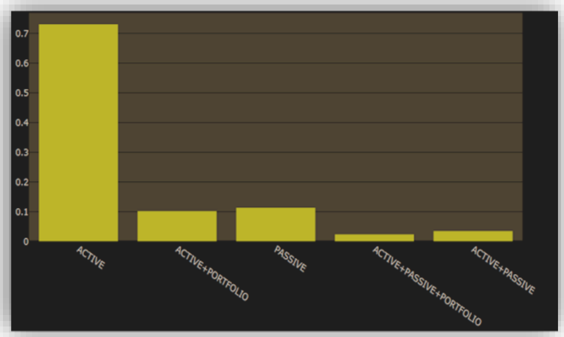
- It was seen that, most of the people do not keep emergency funds in both the population.
- However, in the case of student class, the proportion of students keeping emergency funds is more than that of working class.

7. INCOME TYPES: ACTIVE, PASSIVE, PORTFOLIO

STUDENT



WORKING PROFESSIONAL



The people were asked to state their modes of incomes and we classified them into the following categories:

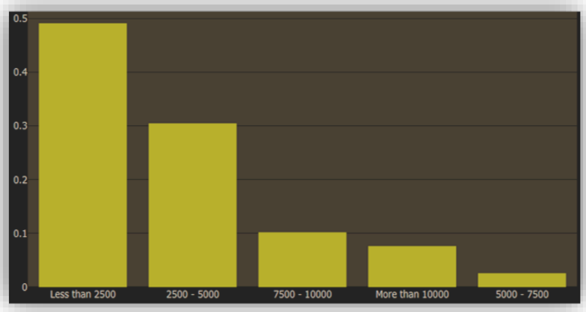
1. Active
2. Passive
3. Portfolio

People with more than one types of incomes were clubbed together with "+". Eg. "Active+Passive"

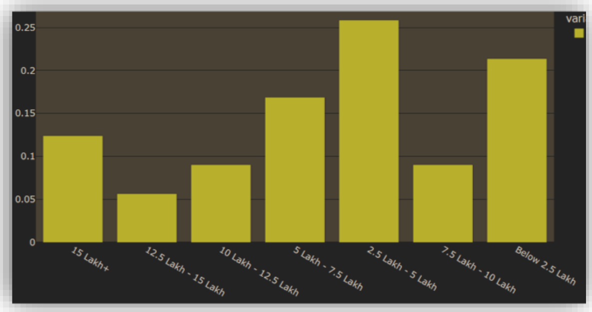
- Working class relief more on active income whereas student class relied on passive sources of income (eg. Pocket money).
- Working class tends to invest their money and hence, earn portfolio income whereas the same trend is not seen with students.

8. INCOME RANGE

STUDENT



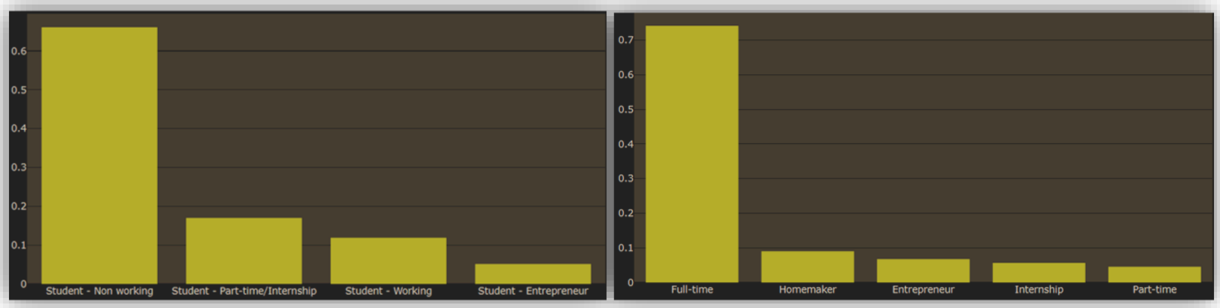
WORKING PROFESSIONAL



- Most of the students have income between 0-5000, whereas most of the working professionals have income 2-5 Lakhs.

9. WORKING STATUS

STUDENT **WORKING PROFESSIONAL**



- Most of the students are non-working and most of the working professionals are full-time workers.

ANALYSIS AND MODELS

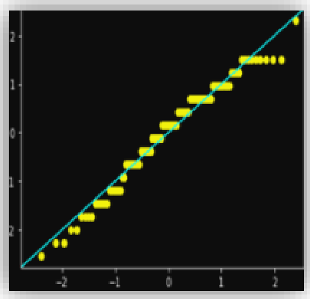
❖ PARAMETRIC ANALYSIS

- D'AGOSTINO K SQUARED TEST

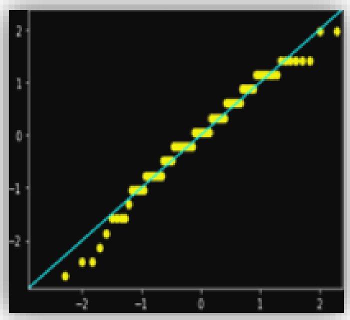
In statistics, **D'Agostino's K^2 test**, named for Ralph D'Agostino, is a goodness-of-fit measure of departure from normality, that is the test aims to establish whether or not the given sample comes from a normally distributed population. The test is based on transformations of the sample kurtosis and skewness

(Checking Normality for the Data)

a) Percentage score of working class
p-value : 0.12157
Level of Significance: 0.05
Conclusion: Percentage scores of **student** class was seen to be following normal distribution



b) Percentage score of working class
p-value: 0.2088
Level of Significance: 0.05
Conclusion: Percentage scores of **working** class was seen to be following normal distribution



▪ BARLETT'S TEST

Bartlett's test for homogeneity of variances is used to test that variances are equal for all samples. It checks that the assumption of equal variances is true before running certain statistical tests It's used when you're fairly certain your data comes from a normal distribution.

Assumptions:

1. Bartlett's test is sensitive to departures from normality
2. Checking if samples have equal variances
3. Two sample population variances

We used Barlett's Test:

- **TO CHECK EQUALITY OF POPULATION VARIANCE FOR PECENTAGE SCORE BETWEEN STUDENT MALES AND FEMALES**
  - **Test Statistic:**

    $$\chi^2 = \frac{(N-k)\ln(S_p^2) - \sum_{i=1}^{k}(n_i - 1)\ln(S_i^2)}{1 + \frac{1}{3(k-1)}\left(\sum_{i=1}^{k}(\frac{1}{n_i-1}) - \frac{1}{N-k}\right)}$$

    P-value: 0.4
  - **Level of significance:0.05**
  - **Conclusion:** Population Variances were found to be Equal.

- **TO CHECK EQUALITY OF POPULATION VARIANCE FOR PECENTAGE SCORE BETWEEN WORKING MALES AND FEMALES**
  - **Test Statistic:**

    $$\chi^2 = \frac{(N-k)\ln(S_p^2) - \sum_{i=1}^{k}(n_i - 1)\ln(S_i^2)}{1 + \frac{1}{3(k-1)}\left(\sum_{i=1}^{k}(\frac{1}{n_i-1}) - \frac{1}{N-k}\right)}$$

  - **P-value: 0.67**
  - **Level of significance:0.05**
  - **Conclusion:** Population Variances were found to be **Equal.**

- **T TEST**

A t-test is a type of inferential statistic used to determine if there is a significant difference between the means of two groups. Essentially, a t-test allows us to compare the average values of the two data sets.

Assumptions:

1. Data values must be independent.
2. Data should be normally distributed – confirmed via D'gostino test
3. Data values are continuous
4. The variances for the two independent groups are equal – confirmed via Barlett's test

We used Two Sample T - Test for mean:

- **TO CHECK EQUALITY OF POPULATION VARIANCE FOR PECENTAGE SCORE BETWEEN STUDENT MALES AND FEMALES**

  - **Test Statistic:**

    $$t = \frac{\bar{x} - \mu}{\frac{s}{\sqrt{n}}}$$

    p-value: 0.32
  - **Level of Significance: 0.05**
  - Conclusion: Means of scores for **student** males and female were found to be equal

- **TO CHECK EQUALITY OF POPULATION VARIANCE FOR PECENTAGE SCORE BETWEEN WORKING MALES AND FEMALES**

  - **Test Statistic:**

    $$t = \frac{\bar{x} - \mu}{\frac{s}{\sqrt{n}}}$$

  - **p-value:0.102**
  - **Level of Significance: 0.05**
  - **Conclusion:** The means of scores for **working** males and female were found to be equal

- ## PROPORTION TEST

Two sample proportion test

Two sample Z test of proportions is the test to determine whether the two populations differ significantly on specific characteristics. In other words, compare the proportion of two different populations that have some single characteristic.

Assumptions:

1) The samples are independent.
2) The assumptions for binomial distribution for both population are satisfied.
3) Values of x1 , n1−x1 , n2−x2 ,and are all more than 5.

We used Two sample proportion test:

- **TO CHECK EQUALITY OF POPULATION PROPORTION FOR HIGH PECENTAGE SCORE BETWEEN STUDENT MALES AND FEMALES**

➢ **Test Statistic:**

$$z = \frac{\hat{p} - p}{\sqrt{\frac{pq}{n}}}$$

➢ **p-value:0.04081**

➢ **Level of Significance: 0.05**

➢ **Conclusion:** The proportion of student high scorers' females was found to be less than male.

- **TO CHECK EQUALITY OF POPULATION PROPORTION FOR HIGH PECENTAGE SCORE BETWEEN WORKING MALES AND FEMALES**

➢ **Test Statistic:**

$$z = \frac{\hat{p} - p}{\sqrt{\frac{pq}{n}}}$$

➢ **p-value:0.02898**

➢ **Level of Significance: 0.05**

➢ **Conclusion:** The proportion of working high scorers females was found to be less than males.

❖ NON- PARAMETRIC TESTING

- **WILCOXON'S TEST (Mann Whitney U Test)**
  (To check for two population Median)

The Mann-Whitney U test is a non-parametric test that can be used in place of an unpaired t-test.

It allows two groups or conditions or treatments to be compared without making the assumption that values are normally distributed

It is used to test the null hypothesis that two samples come from the same population (i.e., have the same median) or, alternatively, whether observations in one sample tend to be larger than observations in the other and so on. Although it is a non-parametric test it does assume that the two distributions are similar in shape.

Assumptions:

1. Dependent variable should be measured on ordinal scale
2. Independent variable should consist of two categories.
3. Observations should be independent.
4. Distribution of scores for both groups of your independent variable have the same shape or a different shape.

We used WILCOXON'S TEST:

- **TO CHECK WHETHER MEDIAN OF PERSONAL INVESTMENT SCORES DIFFER BETWEEN STUDENT CLASS AND WORKING CLASS**

➢ **Test Statistic:**

$$W = \sum_{i=1}^{N_r} [\mathrm{sgn}(x_{2,i} - x_{1,i}) \cdot R_i]$$

➢ **p-value: 0.0002815**
➢ **Level of Significance: 0.05**
➢ **Conclusion:** Median (personal investment) scores for students was found to be less than that of Median (working class)

- **TO CHECK WHETHER MEDIAN OF PERSONAL INVESTMENT SCORES BETWEEN STUDENTS WITH HIGH FAMILY SCORE AND LOW FAMILY SCORE ARE EQUAL.**

➢ **Test Statistic:**

$$W = \sum_{i=1}^{N_r} [\mathrm{sgn}(x_{2,i} - x_{1,i}) \cdot R_i]$$

➢ **p-value: 0.002454**
➢ **Level of Significance: 0.05**
➢ **Conclusion:** Median (personal investment) scores for students with high and Median (family peer score) were found to be equal

- **TO CHECK WHETHER MEDIAN OF PERSONAL INVESTMENT SCORES BETWEEN STUDENTS WITH HIGH PEER SCORE AND LOW PEER SCORE ARE EQUAL**

➢ **Test Statistic:**

$$W = \sum_{i=1}^{N_r} [\mathrm{sgn}(x_{2,i} - x_{1,i}) \cdot R_i]$$

➢ **p-value: 0.2502**
➢ **Level of Significance: 0.05**
➢ **Conclusion:** Median (personal investment) scores for students with high and Median (low peer) score were found to be equal

- **TO CHECK WHETHER MEDIAN OF FINANCIAL SCORES BETWEEN THOSE WHO WERE GUIDED AND UNGUIDED ON MANAGING FINANCE ARE SAME**

➢ **Test Statistic:**

$$W = \sum_{i=1}^{N_r} [\mathrm{sgn}(x_{2,i} - x_{1,i}) \cdot R_i]$$

➢ **p-value: 2.072e-06**
➢ **Level of Significance: 0.05**
➢ **Conclusion:** Median of financial scores between guided and unguided students were found to be equal.

- ## KENDALL'S CORRELATION TEST

The **Kendall rank correlation coefficient**, commonly referred to as **Kendall's τ coefficient** (after the Greek letter τ, tau), is a statistic used to measure the ordinal association between two measured quantities.

It is a measure of rank correlation

. Kendall rank correlation (non-parametric) is an alternative to Pearson's correlation (parametric) when the data you're working with has failed one or more assumptions of the test.

Assumptions:

1. The variables are measured on an **ordinal scale.**

- ### CORRELATION 'KENDALL'TEST

Assumptions:

The assumptions for Kendall's Correlation include:

2. Continuous or ordinal Monotonicity

Correlation test using Kendall rank correlation coefficient.

Test Statistic:

$$\tau_B = \frac{n_c - n_d}{\sqrt{(n_0 - n_1)(n_0 - n_2)}}$$

$$n_0 = \frac{n(n-1)}{2}, \text{ where n is data size}$$

$$n_c = \text{number of concordant (x,y) pairs}$$

$$n_d = \text{discordant pairs}$$

$$n_1 = \sum_j \frac{t_j(t_j - 1)}{2} \ (t_j = \text{number x values tied at jth value})$$

$$n_2 = \sum_k \frac{u_k(u_k - 1)}{2} \ (u_k = \text{number y values tied at kth value})$$

- **TO CHECK WHETHER THERE IS SIGNIFICANT CORRELATION BETWEEN FAMILY SCORE AND FINANCIAL SCORE**
  - **p-value: 2.072e-06**
  - **Level of significance: 0.05**

  - **Conclusion:** There is no significant correlation between Family Score and Total Score.

- **TO CHECK WHETHER THERE IS SIGNIFICANT CORRELATION BETWEEN PEER SCORE AND FINANCIAL SCORE**
  - **p-value: 2.16e-06**
  - **Level of significance: 0.05**
  - **Conclusion:** There is no significant correlation between Peer Score and Total Score.

- **TO CHECK WHETHER THERE IS SIGNIFICANT CORRELATION BETWEEN PEER SCORE AND FAMILY SCORE**
- **p-value: 8.586e-15**
- **Level of significance: 0.05**
- **Conclusion:** There is no significant correlation between Peer Score and Family Score.

▪ KRUSKAL WALLIS TEST

1. **Kruskal Wallis Test**
2. **Pairwise Wilcoxon Test**

1. KRUSKAL WALLIS TEST

A **Kruskal-Wallis test** is used to determine whether or not there is a statistically significant difference between the medians of three or more independent groups. This test is the nonparametric equivalent of the one-way ANOVA and is typically used when the normality assumption is violated.

Assumptions:

One independent variable with two or more levels (independent groups).

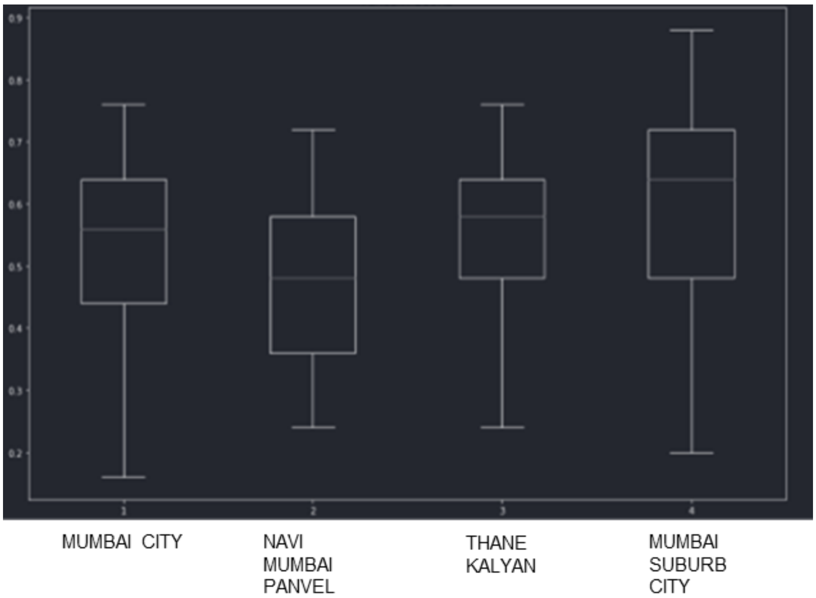
1. Ordinal scale, Ratio Scale or Interval scale dependent variables.
2. Your observations should be independent.
3. All groups should have the same shape distributions.

We used this test to compare mean percent score ranks of the Family Score vs Peer Score

- **Test Statistic:**

$$H = \frac{12}{N(N+1)} \sum_{i=1}^{g} n_i \left(\bar{r}_{i\cdot} - \frac{N+1}{2}\right)^2$$
$$= \frac{12}{N(N+1)} \sum_{i=1}^{g} n_i \bar{r}_{i\cdot}^2 - 3(N+1)$$

- **p-value: 0.02291**
- **Level of significance: 0.05**
- **Conclusion:** At least one of the mean ranks was found to be different.



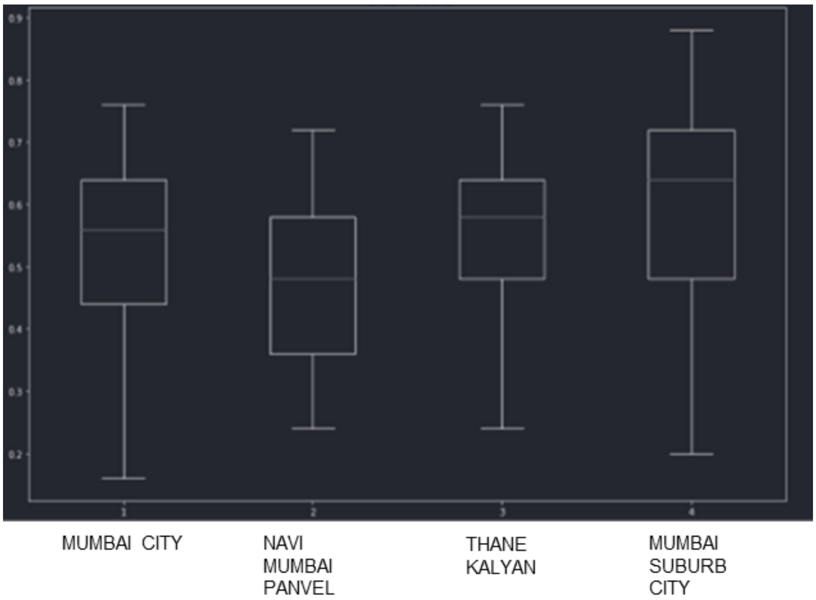
2. PAIRWISE WILCOXON TEST

The **paired samples Wilcoxon test** (also known as **Wilcoxon signed-rank test**) is a **non-parametric** alternative to paired t-test used to compare paired data. It's used when your data are not normally distributed.

We used this test to compare average percentage score difference between Family Score and Peer Score:

- **P-value < 0.005**
- **Level of significance: 0.05**
- **Conclusion:** The average per difference between district was found to be not zero

X / Y	Mumbai City	Mumbai Suburb	Navi Mumbai - Panvel
MS	**0.390**	**-**	**-**
NMP	**0.157**	**0.056**	**-**
TK	**0.157**	**0.360**	**0.035**

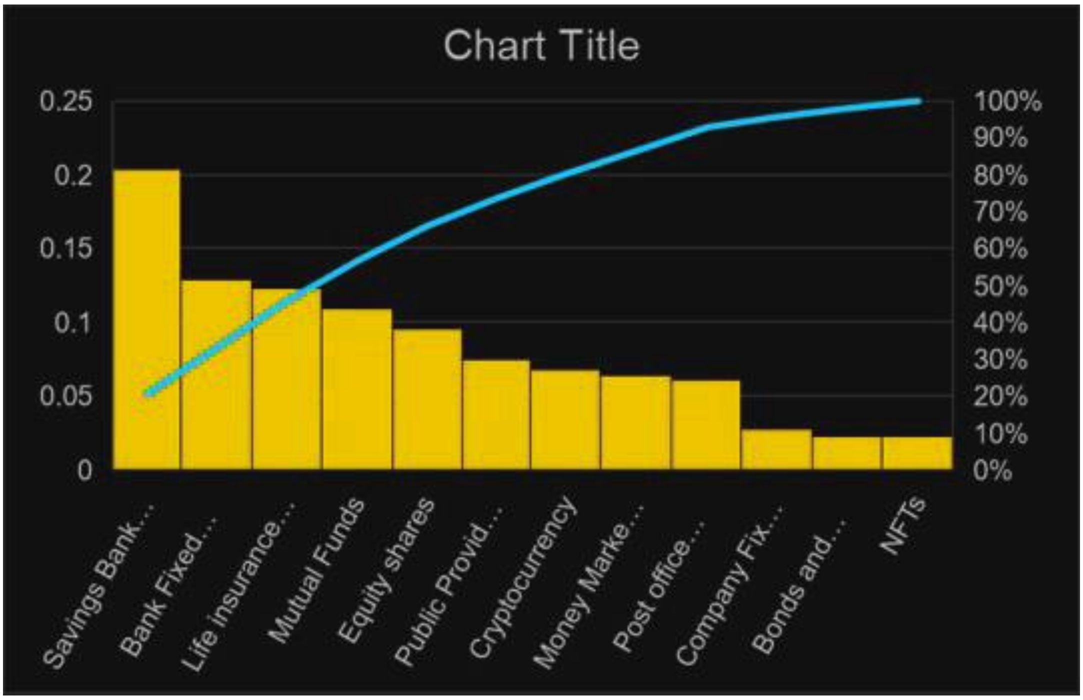


❖ PARETO ANALYSIS

Pareto analysis is a technique used for business decision-making, but which also has applications in several different fields from welfare economics to quality control. It is based largely on the "80-20 rule." As a decision-making technique, Pareto analysis statistically separates a limited number of input factors—either desirable or undesirable—which have the greatest impact on an outcome.

Pareto analysis is premised on the idea that 80% of a project's benefit can be achieved by doing 20% of the work—or, conversely, 80% of problems can be traced to 20% of the causes. Pareto analysis is a powerful quality and decision-making tool. In the most general sense, it is a technique for getting the necessary facts needed for setting priorities

Tendency of People Investing in assets:



❖ MAXIMUM ENTROPY CLASSIFIER

Maximum entropy classifier is a classifier that can be applied in a Single or multi-label classification set ups.

Maximum entropy classifier is a discriminative classifier.

It obtains probability of sample belonging to a specific Class by computing sigmoid (aka logistic function) Of linear combination of features. The weight vector for linear combination is learnt via Model training.

BINARY LOGISTIC REGRESSION

Maximum entropy classifier viz. Logistic regression is an extension of simple linear regression. The dependent variable is dichotomous or binary in nature, we cannot use simple linear regression.

Maximum entropy classifier is the statistical technique used to predict the relationship between predictors (our independent variables) and a predicted variable where the dependent variable is binary. There must be two or more independent variables, or predictors, for a logistic regression. All predictor variables are tested in one block to assess their predictive ability while controlling for the effects of other predictors in the model.

By using Maximum entropy classifier, **we predicted the Level of Percent Score**.

The Features were:

Family Score

Peer Score

Personal Investment Score

Label:

$$\left\{\begin{matrix} \text{High;} & X \geq \text{Median} \\ \text{Low;} & X < \text{Median} \end{matrix}\right\}$$

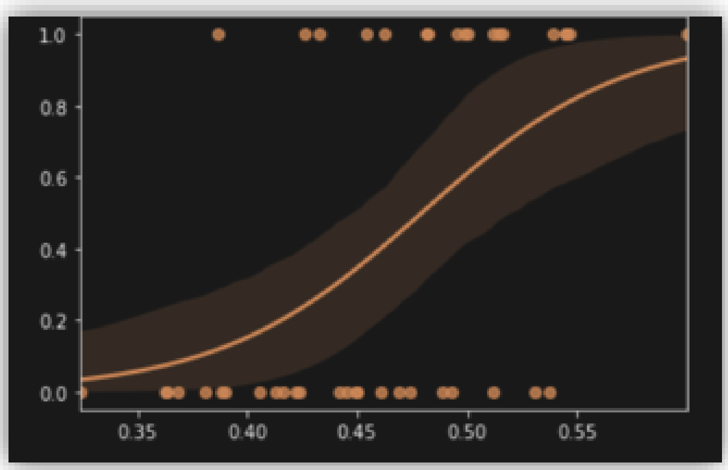
Where X= Percent Score

RESULTS:

Confusion matrix:

21	3
5	13

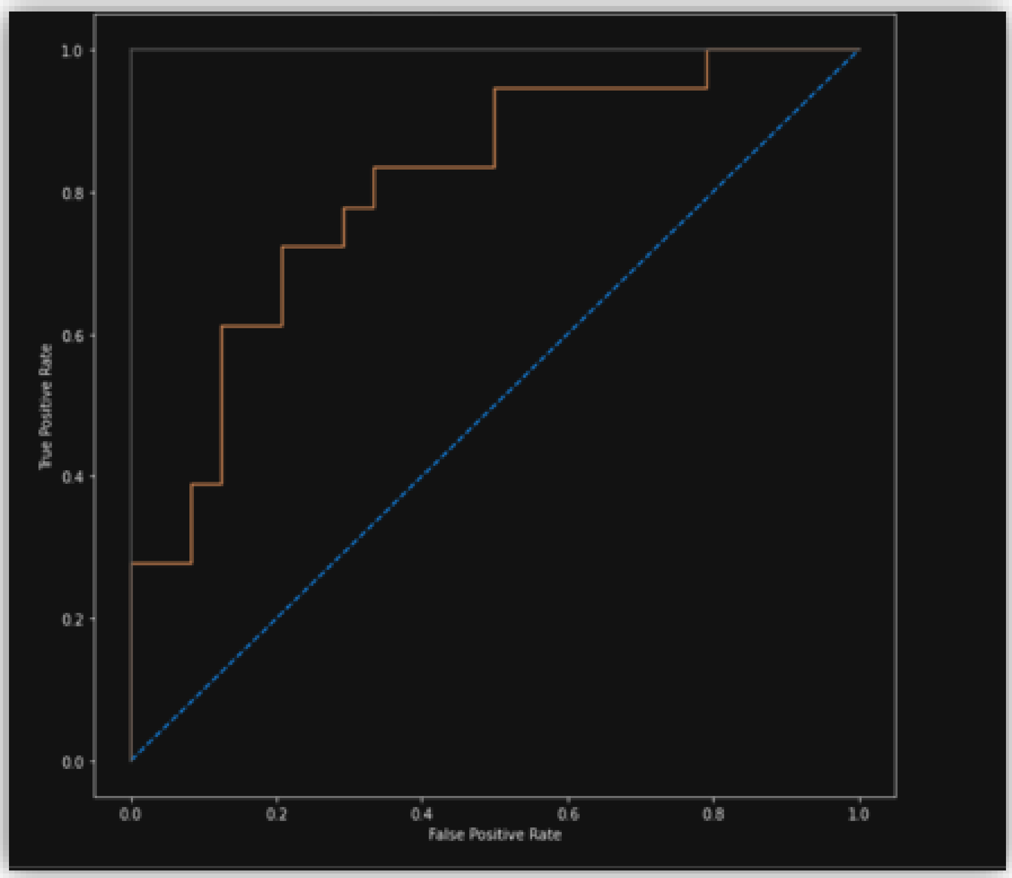
Precision	0.8125
Recall	0.72
F1 score	0.78
Accuracy	0.81



ROC Curve:

The ROC curve shows the trade-off between sensitivity (or TPR) and specificity (1 – FPR). Classifiers that give curves closer to the top-left corner indicate a better performance. The closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate the test.

ROC Curve: Maximum Entropy Classifier



ROC CURVE VALUE : 0.81

We see that the area under the curve is 0.81. This means that 81% variability in our dependent variable is explained by our model.

❖ RANDOM FORESTS

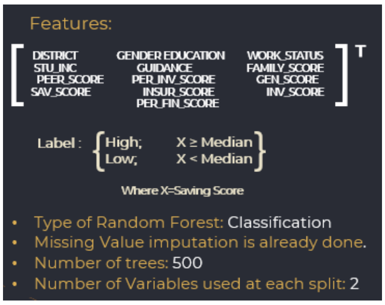
Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems. Random forest also offers features such as Diversity, Parallelization, Train test split and stability.

Random forest extensively uses Bagging, also known as Bootstrap Aggregation. It is the ensemble technique used by random forest. Bagging chooses a random sample from the data set. Hence each model is generated from the samples (Bootstrap Samples) provided by the Original Data with replacement known as row sampling. This step of row sampling with replacement is called bootstrap. Now each model is trained independently which generates results. The final output is based on majority voting after combining the results of all models. This step which involves combining all the results and generating output based on majority voting is known as aggregation.

Random forest is great when working with high dimensional data and is faster to train than decision trees as we are working only on a subset of features in the existing model.

1. Students – Predict Savings Score



Type of Random Forest: Classification

Number of Trees: 500

Number of Variables used at each split: 2

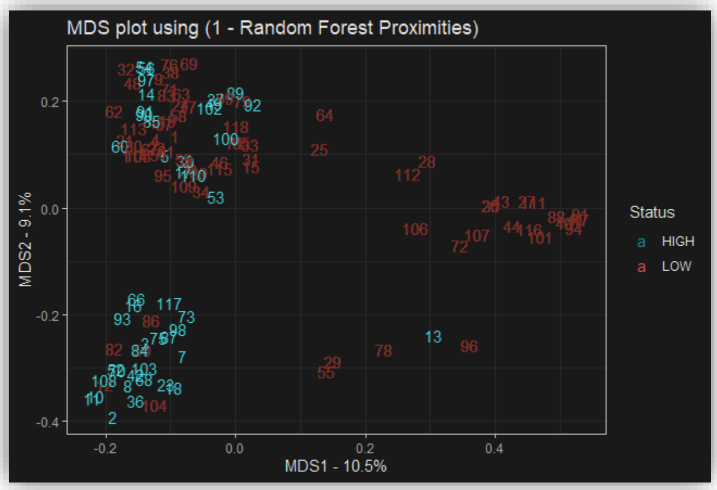
Confusion Matrix:

68	6
16	28

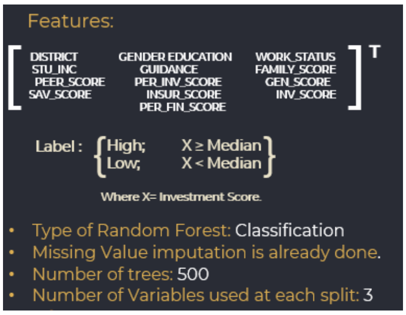
Precision	0.81
Recall	0.857
F1 score	0.83
Accuracy	0.813

Results:

- We can see that the error rates become stationary after 500 trees itself.



2. Working Professionals – Predict Investment Scores



Type of Random Forest: Classification

Number of Trees: 500

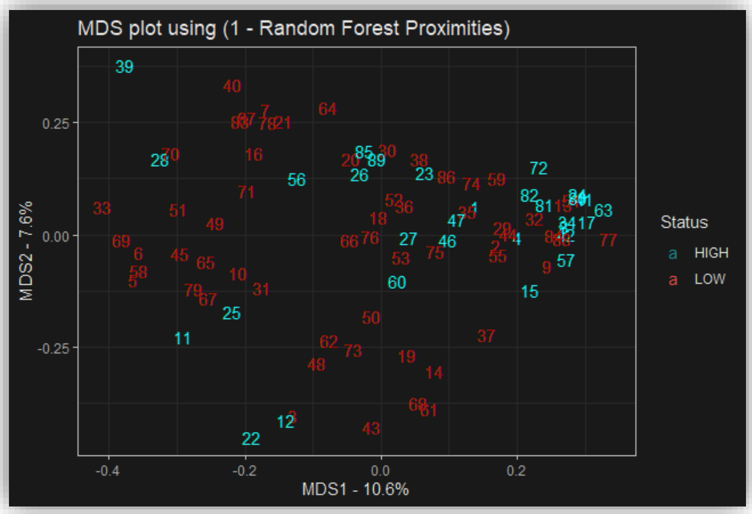
Number of Variables used at each split: 3

Confusion Matrix :

51	8
9	20

Precision	0.80
Recall	0.86
F1 score	0.82
Accuracy	0.80

- The error rates become stationary after 500 trees.



CONCLUSION

- **Financial Literacy scores are surprisingly not affected by your Age, Gender or Working Status (that is whether you are a student or working professional).**

- **Financial Literacy scores is highly affected by factors like your interaction with family and Peers, locality, how much time and resources that you spend on Self Learning etc.**

CODES

- Python Codes

Python libraries used:

Graph objs
Numpy
Matplotlib
Pandas
Plotly
Seaborn
Scipy
Sklearn

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from scipy import stats

from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [ ]:

```
workable_df=pd.read_excel('Workable_data.xlsx',sheet_name='WORKABLE DATA',usecols='A:
W')
```

In [ ]:

```
working_imp=workable_df.select_dtypes(include='number').drop(['GUIDANCE','INV_GUIDE_E
XPERTS','INSUR_SCORE','INV_SCORE','PERCENT_SCORE','EMERG_FUND'],axis=1)
```

In [ ]:

```
plt.figure(figsize=(15,10))
sns.heatmap(working_imp.corr(method='kendall'),annot=True)
plt.show()
```

In [ ]:

```
stats.binom_test(x=18,n=49,p=0.25)
```

In [ ]:

```
workable_df=pd.DataFrame(workable_df)
```

In [ ]:

```
plt.figure(figsize=(20,10))
sns.heatmap(workable_df.corr(method='kendall'),annot=True)
plt.show()
```

In [ ]:

```
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```

In [ ]:

```
plt.figure(figsize=(5,5))
sns.pairplot(workable_df,diag_kind='kde')
```

In [ ]:

```
workable_df['PERCENT_SCORE'].plot.kde()
plt.show()
```

In [ ]:

```
student_df=pd.read_excel('Workable_data.xlsx',sheet_name='STUDENT',usecols='A:W')
```

In [ ]:

```
plt.figure(figsize=(20,10))
sns.heatmap(student_df.corr(method='kendall'),annot=True)
plt.show()
```

In [ ]:

```
plt.figure(figsize=(30,10))
student_df.boxplot()
```

In [ ]:

```
sns.kdeplot(student_df['PERCENT_SCORE'])
```

In [ ]:

```
student_df['normal_percent_score']=(student_df['PERCENT_SCORE'] -student_df['PERCENT_
SCORE'].mean())/ student_df['PERCENT_SCORE'].std()
```

In [ ]:

```
sns.kdeplot(student_df['normal_percent_score'])
```

In [ ]:

```
student_df['normal_percent_score'].std()
```

In [ ]:

```
def normality(data,feature):
    plt.figure(figsize=(10,5))
    plt.subplot(1,2,1)
    sns.kdeplot(data[feature])
    plt.subplot(1,2,2)
    stats.probplot(data[feature],plot=pylab)
    plt.show()
```

In [ ]:

```
workable_df.skew()
```

In [ ]:

```
stats.shapiro(stats.boxcox(student_df['PERCENT_SCORE'],lmbda=1.5444,alpha=0.05))
```

In [ ]:

```
std=StandardScaler()
m=std.fit_transform(student_df.select_dtypes(include='number'))
```

In [ ]:

```
m
```

In [ ]:

```
new_numeric_data_frame=pd.DataFrame(m,columns=student_df.select_dtypes(include='number').columns)
new_numeric_data_frame.describe().T
```

In [ ]:

```
new_numeric_data_frame.skew()
```

In [ ]:

```
new_numeric_data_frame.kurtosis()
```

In [ ]:

```
workable_df.skew()
```

In [ ]:

```
workable_df.kurtosis()
```

In [ ]:

```
student_df.kurtosis()
```

In [ ]:

```
student_score=student_df['PERCENT_SCORE']
```

In [ ]:

```
working_df=pd.read_excel('Workable_data.xlsx',sheet_name='WORKING_PROF',usecols='A:V')
```

In [ ]:

```
stats.normaltest(working_df['PERCENT_SCORE'])
```

In [ ]:

```
stats.normaltest(student_df['PERCENT_SCORE'])
```

In [ ]:

```
stats.normaltest((new_numeric_data_frame['PERCENT_SCORE']))
```

In [ ]:

```
stats.normaltest((student_df['PEER_SCORE']))
```

In [ ]:

```
stats.normaltest(workable_df['PERCENT_SCORE'])
```

In [ ]:

```
stats.anderson(working_df['PERCENT_SCORE'])
```

In [ ]:

```
data =student_df['normal_percent_score']
result = stats.anderson(data)
print('stat=%.3f' % (result.statistic))
for i in range(len(result.critical_values)):
    sl, cv = result.significance_level[i], result.critical_values[i]
    if result.statistic < cv:
        print('Probably Gaussian at the %.1f%% level' % (sl))
    else:
        print('Probably not Gaussian at the %.1f%% level' % (sl))
```

In [ ]:

```
from sklearn.preprocessing import normalize
```

In [ ]:

```
stats.normaltest(student_df['PERCENT_SCORE'])
```

In [ ]:

```
stats.normaltest(workable_df['PERCENT_SCORE'])
```

In [ ]:

```
from scipy.stats import jarque_bera

result = (jarque_bera(stats.boxcox(workable_df['PERCENT_SCORE'],1.5444)))

print(f"JB statistic: {result[0]}")
print(f"p-value: {result[1]}")
```

In [ ]:

```
from scipy.stats import jarque_bera

result = (jarque_bera(student_df['PERCENT_SCORE']))

print(f"JB statistic: {result[0]}")
print(f"p-value: {result[1]}")
```

In [ ]:

```
from scipy.stats import jarque_bera

result = (jarque_bera(workable_df['PERCENT_SCORE']))

print(f"JB statistic: {result[0]}")
print(f"p-value: {result[1]}")
```

In [ ]:

```
from scipy.stats import kstest

result = (kstest(working_df['PERCENT_SCORE'], cdf='norm'))

print(f"K-S statistic: {result[0]}")
print(f"p-value: {result[1]}")
```

In [ ]:

```
from sklearn.preprocessing import OrdinalEncoder
multi=OrdinalEncoder()
k=multi.fit_transform(np.array(workable_df['DISTRICT']).reshape(-1,1))
```

In [ ]:

```
n=multi.fit_transform(np.array(workable_df['EDUCATION']).reshape(-1,1))
```

In [ ]:

```
y=workable_df['PERCENT_SCORE']
```

In [ ]:

```
#EDA
#seed 42
#split
#regression model overall
#anova
#ftest
#cv
#SGD
#polynomial regression
```

#FEATURE_SELECTION
#VALIDATION CURVE
#HYPER TUNNING
#MODEL
#TESTING
#LOGISTIC ON GUIDANCE USING DEMOGRAPHICS

In [ ]:
```
workable_df.describe().T
```

In [ ]:
```
student_female_data=student_df[student_df['GENDER']=='F']
```

In [ ]:
```
student_female_data.shape
```

In [ ]:
```
plt.figure(figsize=(20,10))
sns.heatmap(student_female_data.corr(method='kendall'),annot=True)
```

In [ ]:
```
student_female_data.shape
```

In [ ]:
```
student_femaleguided_data=student_female_data[student_female_data['GUIDANCE']==1]
```

In [ ]:
```
plt.figure(figsize=(20,10))
plt.title('woman with guidance',fontdict={'fontsize':50})
sns.heatmap(student_femaleguided_data.corr(method='kendall'),annot=True)
plt.show()
print(student_femaleguided_data.shape)
```

In [ ]:
```
student_female_not_guided=student_female_data[student_female_data['GUIDANCE']==0]
plt.figure(figsize=(20,10))
plt.title('woman with no guidance',fontdict={'fontsize':50})
sns.heatmap(student_female_not_guided.corr(method='kendall'),annot=True)
plt.show()
print(student_female_not_guided.shape)
```

In [ ]:
```
student_male_data=student_df[student_df['GENDER']=='M']
plt.figure(figsize=(20,10))
plt.title('male student',fontdict={'fontsize':50})
sns.heatmap(student_male_data.corr(method='kendall'),annot=True,cmap='viridis')
plt.show()
print(student_male_data.shape)
```

In [ ]:
```
student_guided_data=student_df[student_df['GUIDANCE']=='1']
student_guided_data
```

In [ ]:
```
student_male_guided_df=student_male_data[student_male_data['GUIDANCE']==1]
plt.figure(figsize=(20,10))
plt.title('male student with guidence',fontdict={'fontsize':50})
sns.heatmap(student_male_guided_df.corr(method='spearman'),annot=True,cmap='viridis')
plt.show()
```

```
print(student_male_guided_df.shape)
```

In [ ]:

```
student_male_not_guided=student_male_data[student_male_data['GUIDANCE']==0]
plt.figure(figsize=(20,10))
plt.title('man with no guidance',fontdict={'fontsize':50})
sns.heatmap(student_male_not_guided.corr(method='kendall'),annot=True,cmap='viridis')
plt.show()
print(student_male_not_guided.shape)
```

In [ ]:

```
student_df['WORK_STATUS'].replace(to_replace='Student - Non working',value='non-working',inplace=True)
student_df['WORK_STATUS'].replace(['Student - Working','Student - Part-time/Internship','Student - Entrepreneur'],'working',inplace=True)
```

In [ ]:

```
student_female_data['WORK_STATUS'].replace(to_replace='Student - Non working',value='non-working',inplace=True)
student_female_data['WORK_STATUS'].replace(['Student - Working','Student - Part-time/Internship','Student - Entrepreneur'],'working',inplace=True)
student_female_non_working=student_female_data[student_female_data['WORK_STATUS']=='non-working']
plt.figure(figsize=(20,10))
plt.title('woman with no work',fontdict={'fontsize':50})
sns.heatmap(student_female_non_working.corr(method='kendall'),annot=True)
plt.show()
print(student_female_non_working.shape)
```

In [ ]:

```
student_female_working=student_female_data[student_female_data['WORK_STATUS']=='working']
plt.figure(figsize=(20,10))
plt.title('woman with  work',fontdict={'fontsize':50})
sns.heatmap(student_female_non_working.corr(method='spearman'),annot=True)
plt.show()
print(student_female_working.shape)
```

In [ ]:

```
student_femaleguided_data['WORK_STATUS'].replace(to_replace='Student - Non working',value='non-working',inplace=True)
student_femaleguided_data['WORK_STATUS'].replace(['Student - Working','Student - Part-time/Internship','Student - Entrepreneur'],'working',inplace=True)
student_female_non_working_guidence=student_femaleguided_data[student_femaleguided_data['WORK_STATUS']=='non-working']
plt.figure(figsize=(20,10))
plt.title('woman with no work no guidence',fontdict={'fontsize':50})
sns.heatmap(student_female_non_working.corr(method='kendall'),annot=True)
plt.show()
print(student_female_non_working_guidence.shape)
```

In [ ]:

```
student_male_data['WORK_STATUS'].replace(to_replace='Student - Non working',value='non-working',inplace=True)
student_male_data['WORK_STATUS'].replace(['Student - Working','Student - Part-time/Internship','Student - Entrepreneur'],'working',inplace=True)
```

```
student_male_non_working=student_male_data[student_male_data['WORK_STATUS']=='non-working']
plt.figure(figsize=(20,10))
plt.title('man with no work',fontdict={'fontsize':50})
sns.heatmap(student_male_non_working.corr(method='kendall'),annot=True,cmap='viridis')
plt.show()
print(student_male_non_working.shape)
```

In [ ]:

```
student_male_working=student_male_data[student_male_data['WORK_STATUS']=='working']
plt.figure(figsize=(20,10))
plt.title('man with work',fontdict={'fontsize':50})
sns.heatmap(student_male_non_working.corr(method='kendall'),annot=True,cmap='viridis')
plt.show()
print(student_male_working.shape)
```

In [ ]:

```
student_mumbai_central=student_df[student_df['DISTRICT']=='MC']
plt.figure(figsize=(20,10))
plt.title('Mumbai central',fontdict={'fontsize':'50'})
sns.heatmap(student_mumbai_central.corr(method='kendall'),annot=True,cmap='Greens')
plt.show()
print(student_mumbai_central.shape)
```

In [ ]:

```
student_mumbai_S=student_df[student_df['DISTRICT']=='MS']
plt.figure(figsize=(20,10))
plt.title('Mumbai S',fontdict={'fontsize':'50'})
sns.heatmap(student_mumbai_S.corr(method='kendall'),annot=True,cmap='BuPu')
plt.show()
print(student_mumbai_S.shape)
```

In [ ]:

```
student_mumbai_tk=student_df[student_df['DISTRICT']=='TK']
plt.figure(figsize=(20,10))
plt.title('Mumbai tk',fontdict={'fontsize':'50'})
sns.heatmap(student_mumbai_tk.corr(method='kendall'),annot=True,cmap='YlGnBu')
plt.show()
print(student_mumbai_tk.shape)
```

In [ ]:

```
student_mumbai_NMP=student_df[student_df['DISTRICT']=='NMP']
plt.figure(figsize=(20,10))
plt.title('Mumbai NMP',fontdict={'fontsize':'50'})
sns.heatmap(student_mumbai_NMP.corr(method='kendall'),annot=True,cmap='Blues')
plt.show()
print(student_mumbai_NMP.shape)
```

In [ ]:

```
student_female_mc=student_female_data[student_female_data['DISTRICT']=='MC']
plt.figure(figsize=(20,10))
plt.title('Mumbai FEMALE MC',fontdict={'fontsize':'50'})
sns.heatmap(student_female_mc.corr(method='kendall'),annot=True,cmap='BuPu')
plt.show()
print(student_female_mc.shape)
```

In [ ]:

```
plt.figure(figsize=(15,10))
plt.boxplot([student_mumbai_central['PERCENT_SCORE'],student_mumbai_NMP['PERCENT_
SCORE'],student_mumbai_S['PERCENT_SCORE'],student_mumbai_tk['PERCENT_SCORE']],)
plt.title('PERCENT SCORE')
plt.show()
```

In [ ]:

```
student_active_data=student_df[(student_df['INCOME_TYPE']=='ACTIVE')]
plt.figure(figsize=(20,10))
plt.title('ACTIVE',fontdict={'fontsize':'50'})
sns.heatmap(student_active_data.corr(method='kendall'),annot=True,cmap='BuPu')
plt.show()
print(student_active_data.shape)
```

In [ ]:

```
items=student_df['INCOME_TYPE'].value_counts()
```

In [ ]:

```
t_test=stats.ttest_ind_from_stats(mean1=student_df["PERCENT_SCORE"].mean(),std1=student
_df["PERCENT_SCORE"].std(),nobs1=student_df["PERCENT_SCORE"].count(),
                              mean2=working_df["PERCENT_SCORE"].mean(),std2=working_df["PE
RCENT_SCORE"].std(),nobs2=working_df["PERCENT_SCORE"].count()      )
```

In [ ]:

```
t_test
```

In [ ]:

```
t_test1=stats.ttest_ind_from_stats(mean1=student_df["FAMILY_SCORE"].mean(),std1=student
_df["FAMILY_SCORE"].std(),nobs1=student_df["FAMILY_SCORE"].count(),
                              mean2=working_df["FAMILY_SCORE"].mean(),std2=working_df["FAM
ILY_SCORE"].std(),nobs2=working_df["FAMILY_SCORE"].count()      )
```

In [ ]:

```
t_test1
```

In [ ]:

```
t_test2=stats.ttest_ind_from_stats(mean1=student_df["PEER_SCORE"].mean(),std1=student_d
f["PEER_SCORE"].std(),nobs1=student_df["PEER_SCORE"].count(),
                              mean2=working_df["PEER_SCORE"].mean(),std2=working_df["PEER_
SCORE"].std(),nobs2=working_df["PEER_SCORE"].count()      )
```

In [ ]:

```
t_test2
```

In [ ]:

```
t_test3=stats.ttest_ind_from_stats(mean1=student_df["PERSONAL_INVESTMENT_SCORE"].
mean(),std1=student_df["PERSONAL_INVESTMENT_SCORE"].std(),nobs1=student_df["PERS
ONAL_INVESTMENT_SCORE"].count(),
                              mean2=working_df["PERSONAL_INVESTMENT_SCORE"].mean(),std2
=working_df["PERSONAL_INVESTMENT_SCORE"].std(),nobs2=working_df["PERSONAL_INV
ESTMENT_SCORE"].count()      )
```

In [ ]:

```
t_test3
```

In [ ]:

```
working_df['PERSONAL_INVESTMENT_SCORE']
```

In [ ]:

```
t_test4=stats.ttest_ind_from_stats(mean1=student_df["PERSONAL_INVESTMENT_SCORE"].
mean(),std1=student_df['PERSONAL_INVESTMENT_SCORE'].std(),nobs1=student_df["PERSO
NAL_INVESTMENT_SCORE"].count(),
                    mean2=working_df["PERSONAL_INVESTMENT_SCORE"].mean(),std2=wor
king_df["PERSONAL_INVESTMENT_SCORE"].std(),nobs2=working_df["PERSONAL_INVEST
MENT_SCORE"].count())
t_test4
```

In [ ]:

```
t_test1=stats.ttest_ind_from_stats(mean1=student_df["SAV SCORE"].mean(),std1=student_df["
SAV SCORE"].std(),nobs1=student_df["SAV SCORE"].count(),
                         mean2=working_df["SAV SCORE"].mean(),std2=working_df["SAV SCO
RE"].std(),nobs2=working_df["SAV SCORE"].count() )
t_test1
```

In [ ]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from scipy.stats import loguniform
from scipy.stats import uniform

from sklearn.datasets import fetch_california_housing
from sklearn.dummy import DummyRegressor

from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Ridge
from sklearn.linear_model import RidgeCV
from sklearn.linear_model import Lasso
from sklearn.linear_model import LassoCV
from sklearn.linear_model import SGDRegressor

from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_absolute_percentage_error


from sklearn.model_selection import cross_validate
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import ShuffleSplit
from sklearn.model_selection import ShuffleSplit
from sklearn.model_selection import validation_curve
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import RandomizedSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
```

In [ ]:

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OrdinalEncoder
from sklearn.preprocessing import PolynomialFeatures
trainset,testset=train_test_split(student_df,test_size=0.2,random_state=42)
```

In [ ]:

```
trainY=trainset["PERCENT_SCORE"]
testY=testset["PERCENT_SCORE"]
trainX=trainset.select_dtypes(include='number').drop(['GOOGLE_SCORE','AGE','GUIDANCE','G
EN_SCORE',
                                    'SAV SCORE','INSUR_SCORE',
                                    'TOTAL SCORE','normal_percent_score','PER_FIN_SCORE','INV_
SCORE',
                                   'PERCENT_SCORE','GUIDANCE_EXPERTS','EMERG_FUND'],axis
=1)
testX=testset.select_dtypes(include='number').drop(['GOOGLE_SCORE','AGE','GUIDANCE','GE
N_SCORE',
                                    'SAV SCORE','INSUR_SCORE',
                                    'TOTAL SCORE','normal_percent_score','PER_FIN_SCORE','INV_
SCORE',
                                   'PERCENT_SCORE','GUIDANCE_EXPERTS','EMERG_FUND'],axis
=1)
```

In [ ]:

```
lin_reg_pipeline=Pipeline([("feature_scaling",StandardScaler()),
                  ("lin_reg",LinearRegression())
                  ])
lin_reg_cv_results=cross_validate(lin_reg_pipeline,
                       trainX,
                       trainY,
                       cv=ShuffleSplit(n_splits=10,test_size=0.2,random_state=360),
                       scoring="neg_mean_absolute_error",
                       return_train_score=True,
                       return_estimator=True)
lin_reg_train_error=-1*lin_reg_cv_results["train_score"]
lin_reg_test_error=-1*lin_reg_cv_results["test_score"]
print(f"mean absolute error of linear regression model on train set :\n"
f"{lin_reg_train_error.mean():} +/- {lin_reg_train_error.std(): }")
print(f"mean absolute error of linear regression model on test set :\n"
f"{lin_reg_test_error.mean():} +/- {lin_reg_test_error.std(): }")
```

In [ ]:

```
from sklearn.preprocessing import MaxAbsScaler
sgd_reg_pipeline=Pipeline([("feature_scaling",MaxAbsScaler()),
                  ("sgd_reg",SGDRegressor(max_iter=np.ceil(1e6/trainX.shape[0]),
                               early_stopping=True,
                               eta0=1e-4,
                               learning_rate="constant",
                               tol=1e-5,
                               validation_fraction=0.1,
                               n_iter_no_change=5,
                               average=10,
                               random_state=42))])
sgd_reg_cv_results=cross_validate(sgd_reg_pipeline,
                        trainX, trainY,
                     cv=ShuffleSplit(n_splits=10,test_size=0.2,random_state=360),
                       scoring="neg_mean_absolute_error",
                       return_train_score=True,
                       return_estimator=True)
```

```
sgd_reg_train_error=-1*sgd_reg_cv_results["train_score"]
sgd_reg_test_error=-1*sgd_reg_cv_results["test_score"]

print(f"mean absolute error of SGD regression model on train set :\n"
f"{sgd_reg_train_error.mean():.3f} +/- {sgd_reg_train_error.std(): .3f}")
print(f"mean absolute error of SGD regression model on test set :\n"
f"{sgd_reg_test_error.mean():.3f} +/- {sgd_reg_test_error.std(): .3f}")
```

In [ ]:

```
from sklearn.preprocessing import PolynomialFeatures
poly_reg_pipeline=Pipeline([("poly",PolynomialFeatures(degree=1)),
                    ("feature_scaling",StandardScaler()),
                    ("lin_reg",LinearRegression())

                    ])
poly_reg_cv_results=cross_validate(poly_reg_pipeline,
                         trainX, trainY,
                        cv=ShuffleSplit(n_splits=10,test_size=0.2,random_state=42),
                        scoring="neg_mean_absolute_error",
                        return_train_score=True,
                        return_estimator=True)

poly_reg_train_error=-1*poly_reg_cv_results["train_score"]
poly_reg_test_error=-1*poly_reg_cv_results['test_score']

print(f"mean absolute error of poly regression regression model on train set :\n"
f"{poly_reg_train_error.mean():} +/- {poly_reg_train_error.std(): }")
print(f"mean absolute error of poly regression regression model on test set :\n"#calculated on dev set and not on real test set
f"{poly_reg_test_error.mean():} +/- {poly_reg_test_error.std(): }")
poly_reg_pipeline.fit(trainX,trainY)
```

In [ ]:

```
from sklearn.metrics import r2_score
sgd_reg_pipeline.fit(trainX,trainY)
k=sgd_reg_pipeline.predict(testX)
t=np.array(testY)-(k)
t=t**2
t.sum()
m=(np.array(testY)-np.array(testY).mean())**2
1-t.sum()/m.sum()
```

In [ ]:

```
((np.array(testY)-poly_reg_pipeline.predict(testX))**2).sum()/m.sum()
```

In [ ]:

```
degree=[1,2,3,4,5,6,7,8,9]
train_scores,test_scores=validation_curve(poly_reg_pipeline,
                            trainX, trainY,
                            param_name="poly__degree",
                            param_range=degree,
                            cv=ShuffleSplit(n_splits=10,test_size=0.2,random_state=42),
                            scoring="neg_mean_absolute_error",
                            n_jobs=2)
train_errors,test_errors=-train_scores,-test_scores
```

```
plt.plot(degree,train_errors.mean(axis=1),"b-x",label="training error")
plt.plot(degree,test_errors.mean(axis=1),"r-x",label="test error")
plt.legend()
plt.xlabel("degree")
plt.ylabel("mean_absolute_error")
_=plt.title("validation curve for polynomial regression")
```

In [ ]:

```
alpha_list=np.logspace(-4,0,num=20)
alpha_list
```

In [ ]:

```
ridge_reg_pipeline=Pipeline([("poly",PolynomialFeatures(degree=2)),
                            ("feature_scaling",StandardScaler()),#alphaS
                            ("ridge_cv",RidgeCV(alphas=alpha_list,
                                            cv=ShuffleSplit(n_splits=10,test_size=0.2,random_state=42),
                                            scoring="neg_mean_absolute_error"))
])
ridge_reg_cv_results=ridge_reg_pipeline.fit(trainX,trainY)#DIFFERENT STEP
```

In [ ]:

```
print("the score with best alpha is ",
     f"{ridge_reg_cv_results[-1].best_score_:.3f}")#NEW DIFFERENT STEP
print("the error with best alpha is ",
     f"{-ridge_reg_cv_results[-1].best_score_:.3f}")#NEW DIFFERENT STEP
print("the best value of alpha is",ridge_reg_cv_results[-1].alpha_)
```

In [ ]:

```
ridge_grid_pipeline=Pipeline([("poly",PolynomialFeatures(degree=2)),
                             ("feature_scaling",StandardScaler()),
                             ("ridge",Ridge())
])

param_grid={"poly__degree":(1,2,3),
          "ridge__alpha":np.logspace(-4,0,num=20)}
ridge_grid_search=GridSearchCV(ridge_grid_pipeline,
                              param_grid=param_grid,
                              n_jobs=2,
                              cv=ShuffleSplit(n_splits=10,test_size=0.2,random_state=42),
                              scoring="neg_mean_absolute_error",
                              return_train_score=True,)
ridge_grid_search.fit(trainX,trainY)
```

In [ ]:

```
mean_train_error=-1*ridge_grid_search.cv_results_["mean_train_score"][ridge_grid_search.be
st_index_]
mean_test_error=-1*ridge_grid_search.cv_results_["mean_test_score"][ridge_grid_search.best
_index_]
std_train_error=ridge_grid_search.cv_results_["std_train_score"][ridge_grid_search.best_inde
x_]
std_test_error= ridge_grid_search.cv_results_["std_test_score"][ridge_grid_search.best_index_
]

print(f"Best mean absolute error of polynomial  ridge regression regression model on train set :\n"
```

```
f"{mean_train_error:} +/- {std_train_error: }")
print(f"Best mean absolute error of polynomial ridge regression regression model on test set :\n"# calculated on dev set and not on real test set
f"{mean_test_error:} +/- {std_test_error: }")
```

In [ ]:

```
print("mean cross validated score of the best estimator is ",ridge_grid_search.best_score_)
print("mean cross validated errorof the best estimator is ",-ridge_grid_search.best_score_)
```

In [ ]:

```
print("the best parameter value is ",ridge_grid_search.best_params_)
```

In [ ]:

```
lasso_reg_pipeline=Pipeline([("poly",PolynomialFeatures(degree=2)),
                    ("feature_Scaling",StandardScaler()),
                    ("lasso",Lasso(alpha=0.01))
])

lasso_reg_cv_results=cross_validate(lasso_reg_pipeline,trainX,trainY,

                        cv=ShuffleSplit(n_splits=10,test_size=0.2,random_state=42),
                        scoring="neg_mean_absolute_error",
                        return_train_score=True,
                        return_estimator=True)
lasso_reg_train_error=-1*lasso_reg_cv_results["train_score"]
lasso_reg_test_error=-1*lasso_reg_cv_results["test_score"]

print(f" mean absolute error oflinear regression model on train set :\n"
f"{lasso_reg_train_error.mean():} +/- {lasso_reg_train_error.std(): }")
print(f" mean absolute error of linear regression model on test set :\n"#calculated on dev set and not on real test set
f"{lasso_reg_test_error.mean():} +/- {lasso_reg_test_error.std(): }")
```

In [ ]:

```
lasso_grid_pipeline=Pipeline([("poly",PolynomialFeatures()),
                    ("feature_scaling",StandardScaler()),
                    ("lasso",Lasso())
])

param_grid={"poly__degree":(1,2,3,4,5,6,7,8,9),
       "lasso__alpha":np.logspace(-4,0,num=20)}
lasso_grid_search=GridSearchCV(lasso_grid_pipeline,
                    param_grid=param_grid,
                    n_jobs=2,
                    cv=ShuffleSplit(n_splits=10,test_size=0.2,random_state=42),
                    scoring="neg_mean_absolute_error",
                    return_train_score=True)
lasso_grid_search.fit(trainX,trainY)
```

In [ ]:

```
mean_train_error=-1*lasso_grid_search.cv_results_["mean_train_score"][lasso_grid_search.be
st_index_]
mean_test_error=-1*lasso_grid_search.cv_results_["mean_test_score"][lasso_grid_search.best_
index_]
std_train_error=lasso_grid_search.cv_results_["std_train_score"][lasso_grid_search.best_inde
x_]
```

```
std_test_error= lasso_grid_search.cv_results_["std_test_score"][lasso_grid_search.best_index_
]

print(f"Best mean absolute error of polynomial  lasso regression regression model on train set :\n"
f"{mean_train_error:} +/- {std_train_error: }")
print(f"Best mean absolute error of polynomial lasso regression regression model on test set :\n"#
calculated on dev set and not on real test set
f"{mean_test_error:} +/- {std_test_error: }")
```

In [ ]:

```
print("mean cross validated score of the best estimator is ",lasso_grid_search.best_score_)
print("mean cross validated errorof the best estimator is ",-lasso_grid_search.best_score_)
```

In [ ]:

```
print("the best parameter value is ",lasso_grid_search.best_params_)
```

In [ ]:

```
poly_sgd_pipeline=Pipeline([("poly",PolynomialFeatures()),
                ("feature_scaling",StandardScaler()),
                ("sgd_reg",SGDRegressor(penalty="elasticnet",random_state=42))
                ])
poly_sgd_cv_results=cross_validate(poly_sgd_pipeline,
                    trainX,trainY,

                     cv=ShuffleSplit(n_splits=10,test_size=0.2,random_state=42),
                     scoring="neg_mean_absolute_error",
                     return_train_score=True,
                     return_estimator=True)

poly_sgd_train_error=-1*poly_sgd_cv_results["train_score"]
poly_sgd_test_error=-1*poly_sgd_cv_results["test_score"]

print(f" mean absolute error oflinear regression model on train set :\n"
f"{poly_sgd_train_error.mean():} +/- {poly_sgd_train_error.std(): }")
print(f" mean absolute error of linear regression model on test set :\n"#calculated on dev set and
not on real test set
f"{poly_sgd_test_error.mean():} +/- {poly_sgd_test_error.std(): }")
```

In [ ]:

```
class unifrom_int:
  def __init__(self,a,b):
    self.distribution=uniform(a,b)
  def rvs(self,*args,**kwargs):
    return self._distribution.rvs(*args,**kwargs).astype(int)
```

In [ ]:

```
baseline_model_median=DummyRegressor(strategy="median")
baseline_model_median.fit(trainX,trainY)
mean_absolute_percentage_error(testY,baseline_model_median.predict(testX))
```

In [ ]:

```
mean_absolute_percentage_error(testY,lin_reg_cv_results["estimator"][0].predict(testX))
```

In [ ]:

```
poly_reg_pipeline.fit(trainX,trainY)
mean_absolute_percentage_error(testY,poly_reg_pipeline.predict(testX))
```

In [ ]:

mean_absolute_percentage_error(testY,ridge_grid_search.best_estimator_.predict(testX))

In [ ]:

mean_absolute_percentage_error(testY,lasso_grid_search.best_estimator_.predict(testX))

- R codes

R libraries:

readxl
randomForest
tidyverse
ggplot2

```
#Random forest Working professionals

library(randomForest)
library(ggplot2)
library(tidyverse)
library(cowplot)
library(readxl)

data = read_xlsx("D:/RFDATA.xlsx", sheet = "rfdata")

head(data)

str(data)
attach(data)
```

```
data$DISTRICT = as.factor(DISTRICT)
data$GENDER = as.factor(GENDER)
data$MARITAL_STATUS = as.factor(MARITAL_STATUS)
data$EDUCATION = as.factor(EDUCATION)
data$WORK_STATUS = as.factor(WORK_STATUS)
data$WORKING_SECTOR = as.factor(WORKING_SECTOR)
data$WORK_PROF_SALARY = as.factor(WORK_PROF_SALARY)
data$GUIDANCE_EXPERTS = as.factor(GUIDANCE_EXPERTS)
data$GEN_SCORE = as.factor(GEN_SCORE)
data$SAV_SCORE = as.factor(SAV_SCORE)
data$INSUR_SCORE = as.factor(INSUR_SCORE)
data$INV_SCORE = as.factor(INV_SCORE)

str(data)

set.seed(40)
rfmodel1 <- randomForest(INV_SCORE~., proximity = TRUE ,data=data)
rfmodel1

rfmodel2 <- randomForest(INV_SCORE~., ntree = 1000,data=data)
rfmodel2

par(mfrow = c(2,1))
plot(rfmodel1) ; plot(rfmodel2)
par(mfrow = c(1,1))
#Stabilizes at 500
#choose rfmodel1 with no of splits = 3

rfmodel1

distance.matrix <- as.dist(1-rfmodel1$proximity)
```

```
mds.stuff <- cmdscale(distance.matrix, eig=TRUE, x.ret=TRUE)
mds.var.per <- round(mds.stuff$eig/sum(mds.stuff$eig)*100, 1)

mds.values <- mds.stuff$points
mds.data <- data.frame(Sample=rownames(mds.values),
            X=mds.values[,1],
            Y=mds.values[,2],
            Status=data$INV_SCORE)

ggplot(data=mds.data, aes(x=X, y=Y, label=Sample)) +
 geom_text(aes(color=Status)) +
 theme_bw() +
 xlab(paste("MDS1 - ", mds.var.per[1], "%", sep="")) +
 ylab(paste("MDS2 - ", mds.var.per[2], "%", sep="")) +
 ggtitle("MDS plot using (1 - Random Forest Proximities)")
--
#Random forest

library(randomForest)
library(ggplot2)
library(tidyverse)
library(cowplot)
library(readxl)

data = read_xlsx("D:/RFDATA.xlsx", sheet = "rfdata2")

head(data)

str(data)
attach(data)
data$DISTRICT = as.factor(DISTRICT)
data$GENDER = as.factor(GENDER)
```

```
data$EDUCATION = as.factor(EDUCATION)
data$WORK_STATUS = as.factor(WORK_STATUS)
data$STU_INC = as.factor(STU_INC)
data$GUIDANCE = as.factor(GUIDANCE)
data$GEN_SCORE = as.factor(GEN_SCORE)
data$SAV_SCORE = as.factor(SAV_SCORE)
data$INSUR_SCORE = as.factor(INSUR_SCORE)
data$INV_SCORE = as.factor(INV_SCORE)
data$PER_FIN_SCORE = as.factor(PER_FIN_SCORE)

str(data)

set.seed(50)
rfmodel1 <- randomForest(SAV_SCORE~., data=data)
rfmodel1

rfmodel2 <- randomForest(SAV_SCORE~., ntree = 1000,data=data)
rfmodel2

par(mfrow = c(2,1))
plot(rfmodel1) ; plot(rfmodel2)
par(mfrow = c(1,1))

oob.value <- vector(length = 10)
for(i in 1:10){
  temp.model<-randomForest(SAV_SCORE~., data = data, mytri = i, ntree = 1000)
  oob.value[i]<- temp.model$err.rate[nrow(temp.model$err.rate),1]
}

which(oob.value == min(oob.value))

rfmodel_op <- randomForest(SAV_SCORE ~ .,
```

```
               data=data,
               ntree=1000,
               proximity=TRUE,
               mtry=2)

rfmodel_op

distance.matrix <- as.dist(1-rfmodel_op$proximity)

mds.stuff <- cmdscale(distance.matrix, eig=TRUE, x.ret=TRUE)
mds.var.per <- round(mds.stuff$eig/sum(mds.stuff$eig)*100, 1)

mds.values <- mds.stuff$points
mds.data <- data.frame(Sample=rownames(mds.values),
                X=mds.values[,1],
                Y=mds.values[,2],
                Status=data$SAV_SCORE)

ggplot(data=mds.data, aes(x=X, y=Y, label=Sample)) +
  geom_text(aes(color=Status)) +
  theme_bw() +
  xlab(paste("MDS1 - ", mds.var.per[1], "%", sep="")) +
  ylab(paste("MDS2 - ", mds.var.per[2], "%", sep="")) +
  ggtitle("MDS plot using (1 - Random Forest Proximities)")
```

QUESTIONNAIRE

FINANCE

Greetings!
The students of SIES College of Arts, Science and Commerce (Autonomous) are conducting this survey in order to observe the knowledge about Finances amongst people.
It is our humble request that you fill this survey with utmost honestly.
The data is collected for purely academic research purposes.

* Required

1. What is your age? *

   Eg. 28,37

   ____

2. Gender *

   Mark only one oval.

   ◯ Female
   ◯ Male
   ◯ Other

3. Which district do you reside in? *

   Mark only one oval.

   ◯ Mumbai City
   ◯ Mumbai Suburb
   ◯ Thane-Karjat
   ◯ Navi-Mumbai-Panvel
   ◯ Other: ______________________

4. What is your Marital Status? *

Mark only one oval.

- Single
- Married
- Divorced
- Widowed

5. What is your highest education qualification? *

Mark only one oval.

- Below SSC
- SSC
- HSC
- Diploma
- Undergraduate
- Post Graduate
- Phd
- Graduate
- Other: ______________________

6. Which course did you pursue during your undergraduate?(Eg. BAF, BSc IT, BCom, BMM,BTech IT etc) *

If you have not completed your undergraduate, please write NA.

____

7. What is/are your main modes of income? *

*MULTIPLE SELECTIONS ARE ALLOWED

Check all that apply.

- [ ] Pocket Money
- [ ] Internship stipend
- [ ] Job Salary (Earned Income)
- [ ] Business profits
- [ ] Dividend
- [ ] Rental
- [ ] Capital gains
- [ ] Royalties or Licensing Incomes
- [ ] No source of income
- [ ] Pension
- [ ] Spouse's Share of Income

8. What is your working status? *

Mark only one oval.

- Student - Non working *Skip to question 9*
- Student - Working *Skip to question 9*
- Student - Entrepreneur *Skip to question 9*
- Student - Part-time/Internship *Skip to question 9*
- Internship *Skip to question 11*
- Part-time *Skip to question 11*
- Full-time *Skip to question 11*
- Entrepreneur *Skip to question 11*
- Unemployed *Skip to question 13*
- Homemaker *Skip to question 11*
- Retired *Skip to question 13*

Student Section

9. What is your MONTHLY pocket money(in Rs)? (eg. 2500) *

Write "0" if you don't get pocket money.

10. What is your income per MONTH? (in Rs.) - FOR WORKING STUDENT *

If you are a working student, then select your MONTHLY income salary range. Click NA if you are a Student - Non working or receive no stipend

Mark only one oval.

- Less than 2500
- 2500 - 5000
- 5000 - 7500
- 7500 - 10000
- 10000+
- NA

Skip to question 13

Working Class Section

11. Choose your working sector from the options given below, *

Mark only one oval.

- Private
- Government
- Self Employed
- Dependent

12. What is your salary per annum?(in Rs.) *

Mark only one oval.

- Below 2.5 Lakh
- 2.5 Lakh - 5 Lakh
- 5 Lakh - 7.5 Lakh
- 7.5 Lakh - 10 Lakh
- 10 Lakh - 12.5 Lakh
- 12.5 Lakh - 15 Lakh
- 15 Lakh+

Skip to question 13

Modes of learning about finances

13. Where do you get the information you need about money matters (such as spending, savings, banking, investment)? *

*MULTIPLE SELECTIONS ARE ALLOWED

Check all that apply.

- Parents/Guardian
- Relatives
- Spouse
- Friends/Peers
- Television/Radio
- Financial Books
- Social Media
- Magazines/Newspaper
- Advertisements (Flyers, Billboards,etc)
- Forums/Webinars/Seminars
- Clubs and Social Circles (eg Rotary Club, Lions , etc)

Other:

14. Have you been guided/trained on managing personal finance? *

Mark only one oval.

- Yes
- No

15. Do you discuss managing personal finance with your family? *

Mark only one oval per row.

	Never	Rarely	Sometimes	Often	Always
Investment and risks (Stock market, MFs, FDs, etc)					
Investment in digital assets(Cryptocurrency, NFTs)					
Insurance - Life, Healthcare, General (Motor, accident, etc)					
Savings and Budgeting					
News related to economics or finance					
Guidance on spending decisions					

16. Do you discuss managing personal finance with your Friends/Peers? *

Mark only one oval per row.

	Never	Rarely	Sometimes	Often	Always
Investment and risks (Stock market, MFs, FDs, etc)	⬭	⬭	⬭	⬭	⬭
Investment in digital assets(Cryptocurrency, NFTs)	⬭	⬭	⬭	⬭	⬭
Insurance - Life, Healthcare, General (Motor, accident, etc)	⬭	⬭	⬭	⬭	⬭
Savings and Budgeting	⬭	⬭	⬭	⬭	⬭
News related to economics or finance	⬭	⬭	⬭	⬭	⬭
Guidance on spending decisions	⬭	⬭	⬭	⬭	⬭

Spending Patterns

17. What do you invest in? *

Mark only one oval per row.

	Yes	No
Savings Bank Account	◯	◯
Money Market Funds(also known as liquid funds)	◯	◯
Bank Fixed Deposits	◯	◯
Post office Savings Schemes	◯	◯
Public Provident fund	◯	◯
Company Fixed Deposits	◯	◯
Bonds and debentures	◯	◯
Mutual Funds	◯	◯
Life insurance policies	◯	◯
Equity shares	◯	◯
Cryptocurrency	◯	◯
NFTs	◯	◯

18. Do you take consultations for your above mentioned investments by experts. *

Mark only one oval.

◯ Yes

◯ No

What is the percentage of your monthly income that you allocate to the following three categories?

19. Savings and Investment *

Mark only one oval.

- Less than 20%
- Equal to 20%
- More than 20

20. Essentials *

Mark only one oval.

- Equal to 50
- Less than 50
- More than 50

21. Wants(Luxuries) *

Mark only one oval.

- Equal to 30
- Less than 30
- More than 30

22. Do you have excess funds incase of an emergency? *

Mark only one oval.

- Yes
- No

23. What are your reasons for investments? (ex : education ,dream house, etc)
*WRITE IN TWO OR THREE WORDS

Subsection 1: Personal Finance and Opinions

24. Do you maintain financial records? *

Mark only one oval.

- Maintain very detailed records
- Maintain minimal records
- Maintain no records

25. What do you do with your pocket money? *

Mark only one oval.

- Spend it fully
- Save a portion of it in the bank
- Save it in home
- NA(not applicable)

26. How often you ask for extra money from your parents *

Mark only one oval.

- Once in a month
- 2-3 times in a month
- Never
- NA(not applicable)

27. How often do you borrow money from your friends/family/peers/spouse? *

Mark only one oval.

- Never
- Once in a month
- 2-3 times in a month
- More than 3 times amonths

28. Do you inform your parents about the expenditures at college in advance? *

Mark only one oval.

- Yes
- No
- NA(not applicable)

Subsection 2: General Personal Finance Knowledge

29. Personal financial planning involves * 1 point

Mark only one oval.

- Monitoring Expenses
- Minimizing expenses
- Financial planning
- All of the above

30. What is an asset? * 1 point

Mark only one oval.

- Any item owned by a business or individual.
- An obligation to pay money to a third party.
- A financial obligation
- All of the above

31. Your net worth is the difference between your * 1 point

Mark only one oval.

- Expenditures and income
- Liabilities and assets
- Bank borrowings and savings
- None of the above

32. You are NOT overspending if * 1 point

Mark only one oval.

- Your rent or mortgage exceeds 30% of after-tax income
- Your monthly expenditure is less than your monthly income.
- You are missing due dates for making your monthly pre-payments(Bills)
- You don't have extra money from your income to support unexpected expenses

33. Cost of taking an apartment on lease/rent includes, * 1 point

A. Security deposit
B. Monthly rental payment
C. Expenses incurred for non-compliance of lease terms
D. Medical expenses of your friend who fell and broke his arm in the apartment

Mark only one oval.

- Option A and D
- Option B only
- Option A,B and C
- Option A, B and D

34. How does the 50-20-30 rule distribute your income? * 1 point

Mark only one oval.

- 50% expenses, 20% flexible spending, 30% saving
- 50% expenses, 20% saving, 30% flexible spending
- 50% flexible spending, 20% saving, 30% expenses
- 50% saving, 20% flexible spending, 30% expenses

Subsection 3: Savings and borrowings

35. To earn as much interest as possible, you should open a savings account that earns ______ interest and has the _____ interest rate. * 1 point

Mark only one oval.

- compound; lowest
- compound; highest
- simple; lowest
- simple; highest

36. The value of 2 crore in money today is less than the value of 2 crore after 5 years? * 1 point

Mark only one oval.

- True
- False

37. About how much should you save in an emergency fund? * 1 point

Mark only one oval.

- 1-3 months of living expenses
- 3-6 months of living expenses
- 6-9 months of living expenses
- 9-12 months of living expenses

38. Having a high credit score will allow lenders to give you lower interest rates. * 1 point

Mark only one oval.

- True
- False

39. Which of the following investments require that you keep your money invested for a specified period or face an early withdrawal penalty? * 1 point

Mark only one oval.

- Fixed deposit.
- Savings Accounts.
- National Saving Certificates.
- Current Accounts.
- None of these.

40. You may receive your financial report from * 1 point

Mark only one oval.

- Credit information bureau (Eg. CIBIL)
- A commercial bank
- Post Office
- University
- Retail Store

41. Which is FALSE about credit cards? * 1 point

Mark only one oval.

- You can use your credit card to receive a cash advance
- If your credit limit is Rs. 10,000, and you utilize a credit of Rs. 4,000, then interest would be charged on Rs. 10,000
- A credit card company will not charge you interest if you pay off the entire balance by the due date
- You cannot spend more than your credit limit.

42. You will improve your creditworthiness by * 1 point

Mark only one oval.

- Visiting your local commercial bank
- By showing good repayment history
- Paying cash for all goods and services
- Borrowing large amounts of money from your friends
- Donating money to charity

43. If you become a guarantor on a loan taken by your friend then * 1 point

Mark only one oval.

- You become responsible for the loan payments if your friend defaults
- It means that your friend cannot receive the loan by himself
- You are entitled to receive part of the loan
- All of the above

Subsection 4: Insurance

44. The main reason to purchase insurance is to * 1 point

Mark only one oval.

- Protect you from a loss recently incurred.
- Provide you with excellent investment returns.
- Compensates you for a potential loss in future .
- Decrease the chances of accidents.
- Improve your standard of living by filing fraudulent claims.

45. Under current law, until what age can a child stay on their parents' health insurance? * 1 point

Mark only one oval.

- 18
- 21
- 26
- 29

46. Which government agency would you go to resolve complaint against insurance company? * 1 point

Mark only one oval.

- Consumer Court.
- Grievance Redressal Officers, GRO, of all insurance companies.
- Grievance Redressal Cell of the Consumer Affairs Department of IRDA.
- All of the above.
- None of the above.

47. Term Insurance Means * 1 point

Mark only one oval.

- It is the policy wherein the insured gets death benefit if any contingency happens within the policy term.
- The insured is, however, not entitled to receive any survival benefit if he outlives the policy term.
- These plans are relatively cheaper than endowment policies, money back policies and ULIPs.
- All of the above.
- None of the above.

48. Microinsurance is meant for * 1 point

Mark only one oval.

- Protecting lower income sections
- Urban Area
- Rural area
- Involves Small amount

Subsection 5: Investments

49. If you own a company's stock then * 1 point

Mark only one oval.

- You own a part of the company
- You have lent money to the company
- You are liable to the company's debt
- The company will return your original investment with interest

50. Assume you're in your early twenties and you would like to build up for a secure retirement in the next 30 years. Which of the following approaches should not be in your plan? * 1 point

Mark only one oval.

- Start to build up your savings account at a commercial bank
- Save money in fixed deposit accounts
- Put monthly savings in a diversified growth mutual fund
- Invest in Pension Schemes
- None of the above

51. In general, investments that are riskier tend to provide higher returns over time than investment with less risk * 1 point

Mark only one oval.

- True
- False

52. Which of the following is a benefit of 'Systematic Investment Plans (SIPs)? * 1 point

Mark only one oval.

- Brings Discipline by automated investments
- Rupee Cost Averaging
- Affordable Investment Plans
- All of these

53. If US dollar value increase, the value of Indian rupee * 1 point

Mark only one oval.

- Decreases
- Increases
- Remains same
- Cannot be determined based on the information given

REFERENCES

- https://towardsdatascience.com/understanding-random-forest-58381e0602d2

- https://drive.google.com/file/d/1Yv1RPwCUon548Z5fZ72djRuMaJSxfubO/view?usp=sharing

- https://en.wikipedia.org/wiki/Random_forest#:~:text=Random%20forests%20or%20random%20decision,decision%20trees%20at%20training%20time.&text=Random%20forests%20generally%20outperform%20decision,lower%20than%20gradient%20boosted%20trees.

- https://www.sciencedirect.com/topics/computer-science/binary-logistic-regression#:~:text=Binary%20logistic%20regression%20(LR)%20is,or%20not%20readmitted%20(0).

- https://www.statisticshowto.com/wilcoxon-signed-rank-test/#:~:text=Wilcoxon%20Signed%20Ranks%20Test%20Statistic,W%E2%80%93%20for%20the%20test%20statistic.&text=Compare%20your%20test%20statistic%20to,a%205%25%20alpha%20level).

- https://en.wikipedia.org/wiki/Kendall_rank_correlation_coefficient#:~:text=In%20statistics%2C%20the%20Kendall%20rank,association%20between%20two%20measured%20quantities.

- https://www.statisticshowto.com/bartletts-test/