NTUC CAPSTONE PROJECT: FINANCE (PREDICTIVE ANALYSIS)

BY: NIRVAN JOTHI UTHAYAJOTHI

MODULE: (SCTP) ASSOCIATE DATA ANALYST

MODULE BATCH: VLC-SCTP211-24-0658

PROJECT & OBJECTIVE

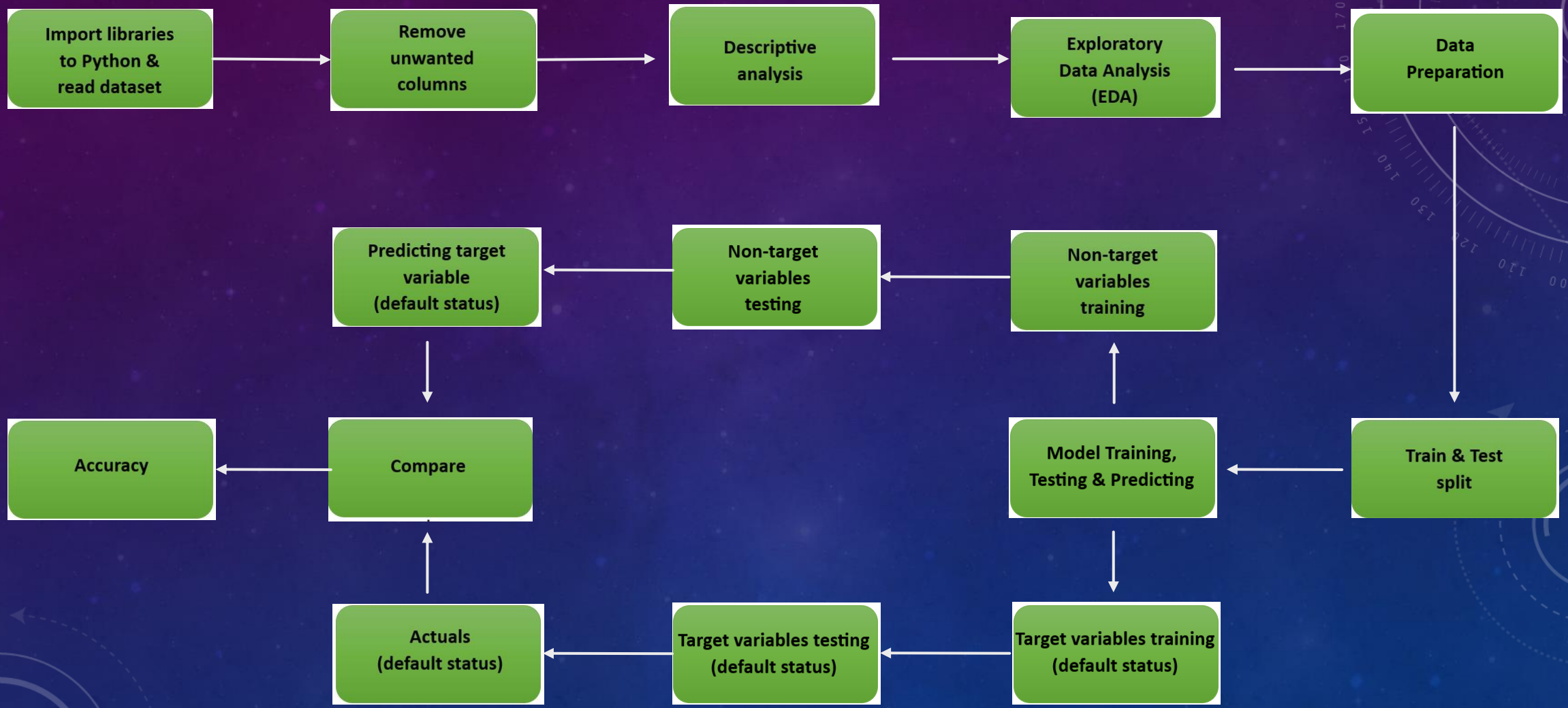
PROBLEM STATEMENT

In the financial industry, it is crucial for lenders to assess the credit worthiness of borrowers before granting loans or credit. Identifying potential defaulters, who are at higher risk of failing to repay their debts, can help mitigate financial losses and maintain a healthy lending portfolio. The goal of this project is to develop a predictive model that can accurately classify borrowers as defaulters or non-defaulters based on various financial and demographic factors.

RESEARCH OBJECTIVE

- To create a machine learning model to predict the defaulter and Non-defaulter by analyzing historical data

MIND-MAP

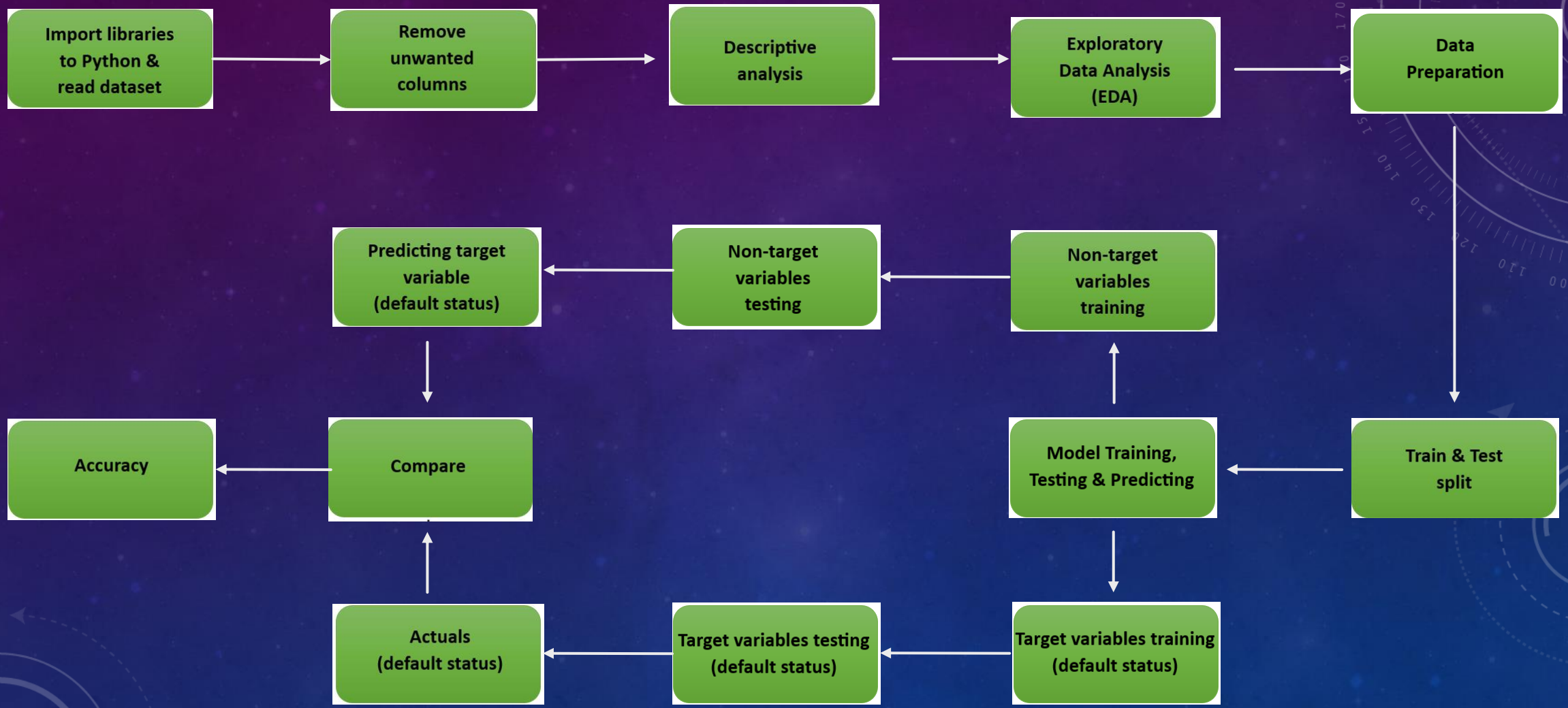

Import libraries to Python & read dataset
Remove unwanted columns
Descriptive analysis
Exploratory Data Analysis (EDA)
Data Preparation
Train & Test split
Model Training, Testing & Predicting
Target variables training (default status)
Target variables testing (default status)
Actuals (default status)
Compare
Accuracy
Predicting target variable (default status)
Non-target variables testing
Non-target variables training

DATA UNDERSTANDING AND PREPARATION

- Python was used to conduct the analysis and predictions
- All the necessary libraries are imported, along the way more libraries imported as needed. These libraries are needed in order to perform the analysis, predictions, calculations, plot graphs and more
- The loan excel csv file was uploaded into python and viewed
- Columns that were not needed for the analysis were removed such as dates and ids
- Descriptive analysis was done to check the number of rows, columns, data types, any missing values, etc…
- There are no missing values, the data type for all columns are in correct format (int & float for numbers, etc…)

#	Column	Non-Null Count	Dtype
0	loan_type	5000 non-null	object
1	loan_amount	5000 non-null	int64
2	interest_rate	5000 non-null	float64
3	loan_term	5000 non-null	int64
4	employment_type	5000 non-null	object
5	income_level	5000 non-null	object
6	credit_score	5000 non-null	int64
7	gender	5000 non-null	object
8	marital_status	5000 non-null	object
9	education_level	5000 non-null	object
10	default_status	5000 non-null	bool

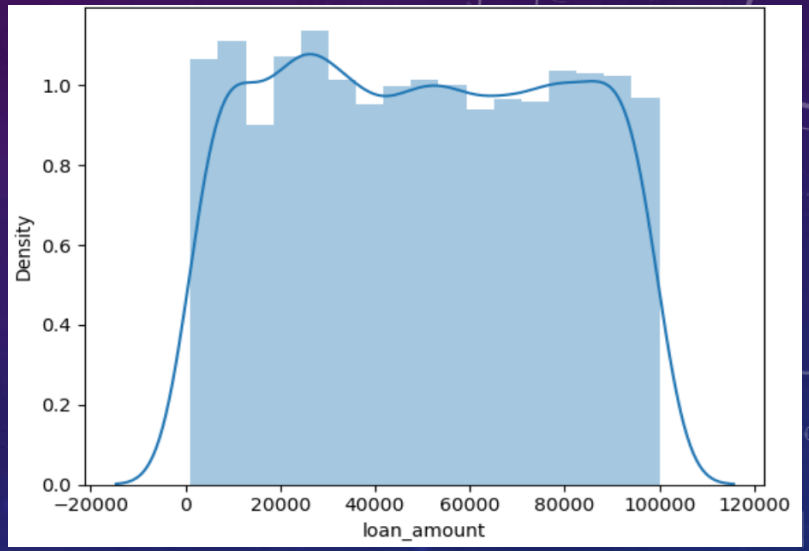
MIND-MAP

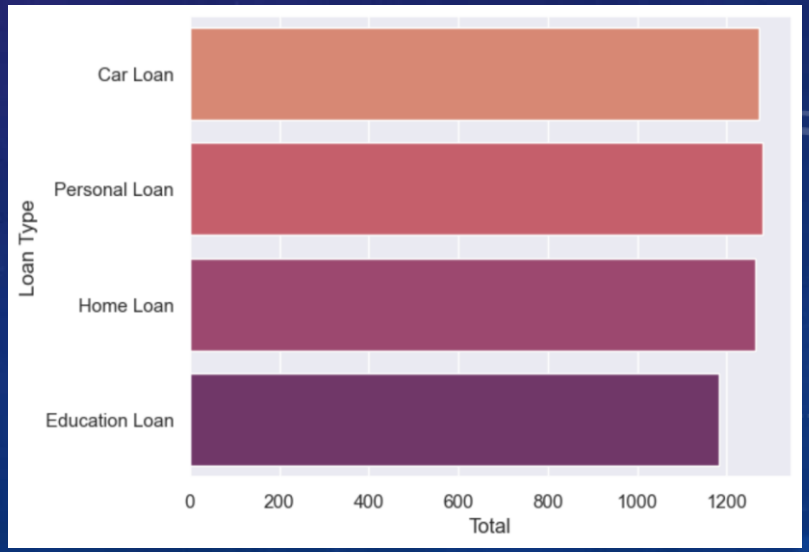

Import libraries to Python & read dataset
Remove unwanted columns
Descriptive analysis
Exploratory Data Analysis (EDA)
Data Preparation
Train & Test split
Model Training, Testing & Predicting
Target variables training (default status)
Target variables testing (default status)
Actuals (default status)
Compare
Accuracy
Predicting target variable (default status)
Non-target variables testing
Non-target variables training

DATA UNDERSTANDING AND PREPARATION

EXPLORATORY DATA ANALYSIS (EDA)

- To explore the data, methods were used to find the statistics, value counts, relationship between categories, spread of data in each category and to plot graphs
- All the numerical columns/variables tend to have values which are spread centrally within a certain range making them close to or being symmetric
- All categorical columns/variables tend to have category totals that are close to one another
- There is no over or under sampling: no category of any categorical column/variable is exceptionally above or below the rest of the categories under the variable, and there are no values that are exceptionally above or below the rest of the values for any numerical column/variable

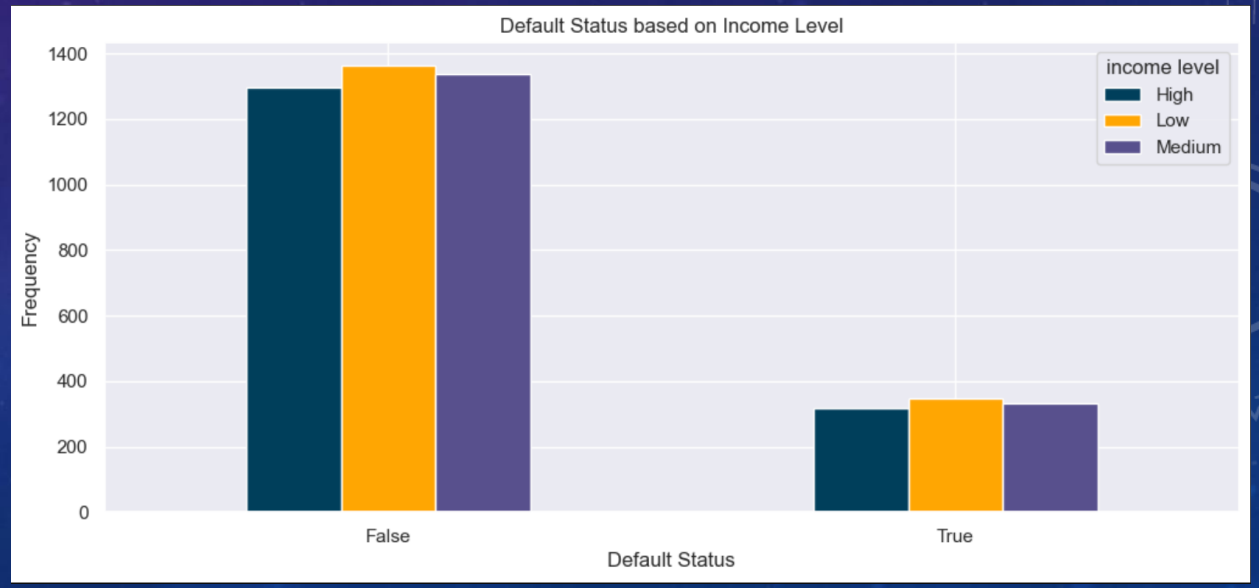




DATA UNDERSTANDING AND PREPARATION

- Exploring further, one example is plotting the default status based on income level to find out the frequency of which income level defaulted and did not default based on the data provided
- There are 4001 false default status and 999 true defaults status, reaching 5000 records totally as per dataset provided. False default status meaning the customer is able to pay back the loan while if the customer defaults (true), meaning the customer is not able to pay back the loan
- Non-defaulters (false) are higher in all income level categories, with low income level being the highest, followed by medium then high income levels
- For the defaulters (true), low income level is the highest again, followed by medium then high income levels
- We would expect for the non-defaulters (false) that high income level will be the most frequent and low income level the least frequent, however it is the other way around. May be influenced by other variables

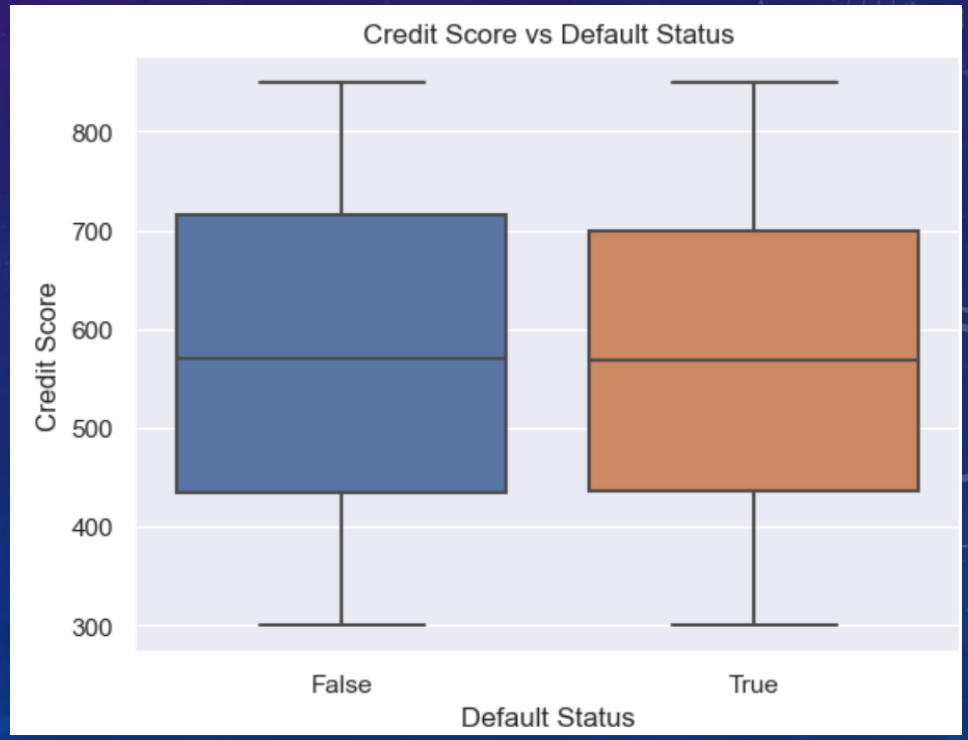
```
False    4001
True      999
Name: default_status
```



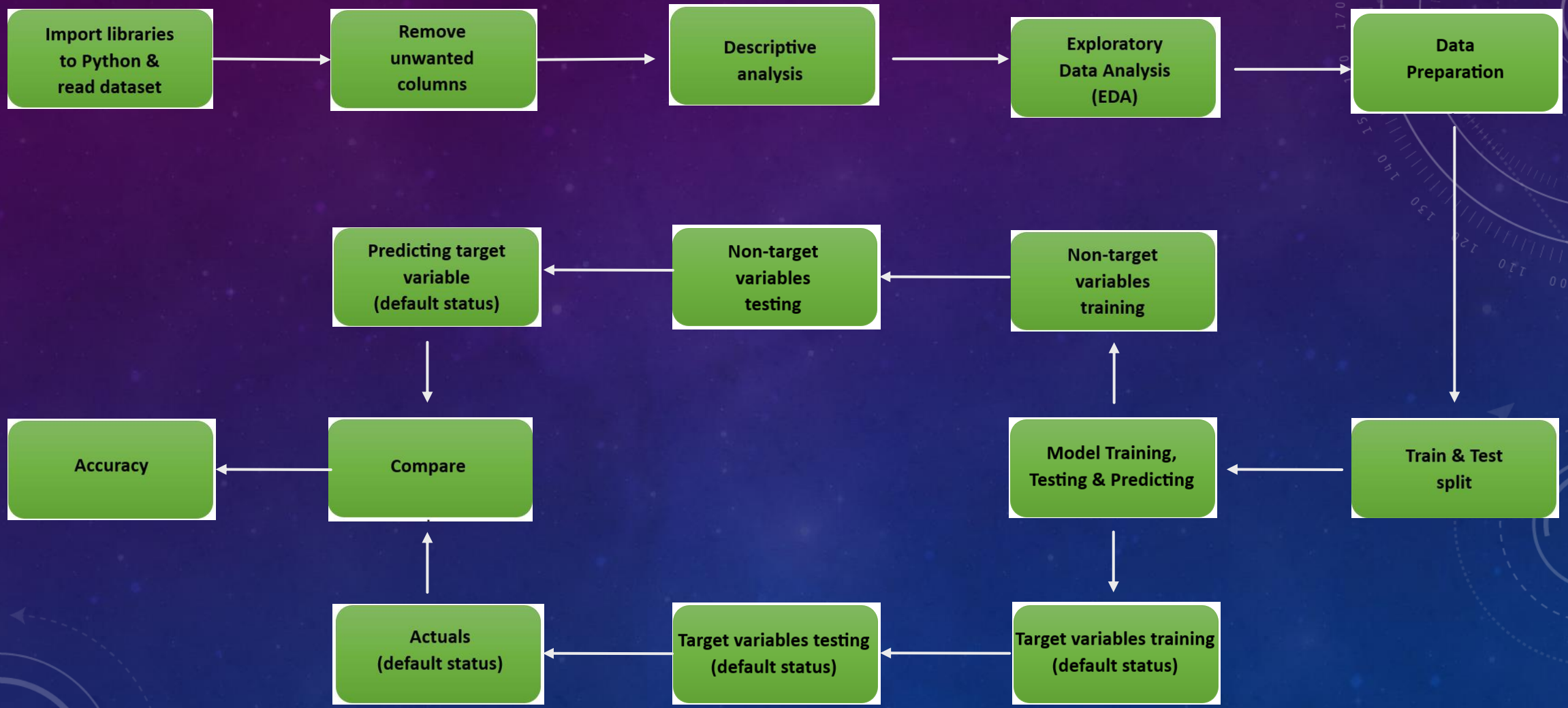
DATA UNDERSTANDING AND PREPARATION

- Exploring Credit Score with Default status (target variable), both false and true default status have similar values for their medians, maximum and minimum, lower quartiles, etc.
- There are no outliers (values that are exceptionally high or low)
- The spread of the data is identical for both categories and symmetric

	Default Status	
	False	True
Maximum:	870-880	870-880
Upper Quartile (75%):	710-720	700
Median (50%):	570-580	570-580
Lower Quartile (25%):	420-430	420-430
Minimum:	300	300



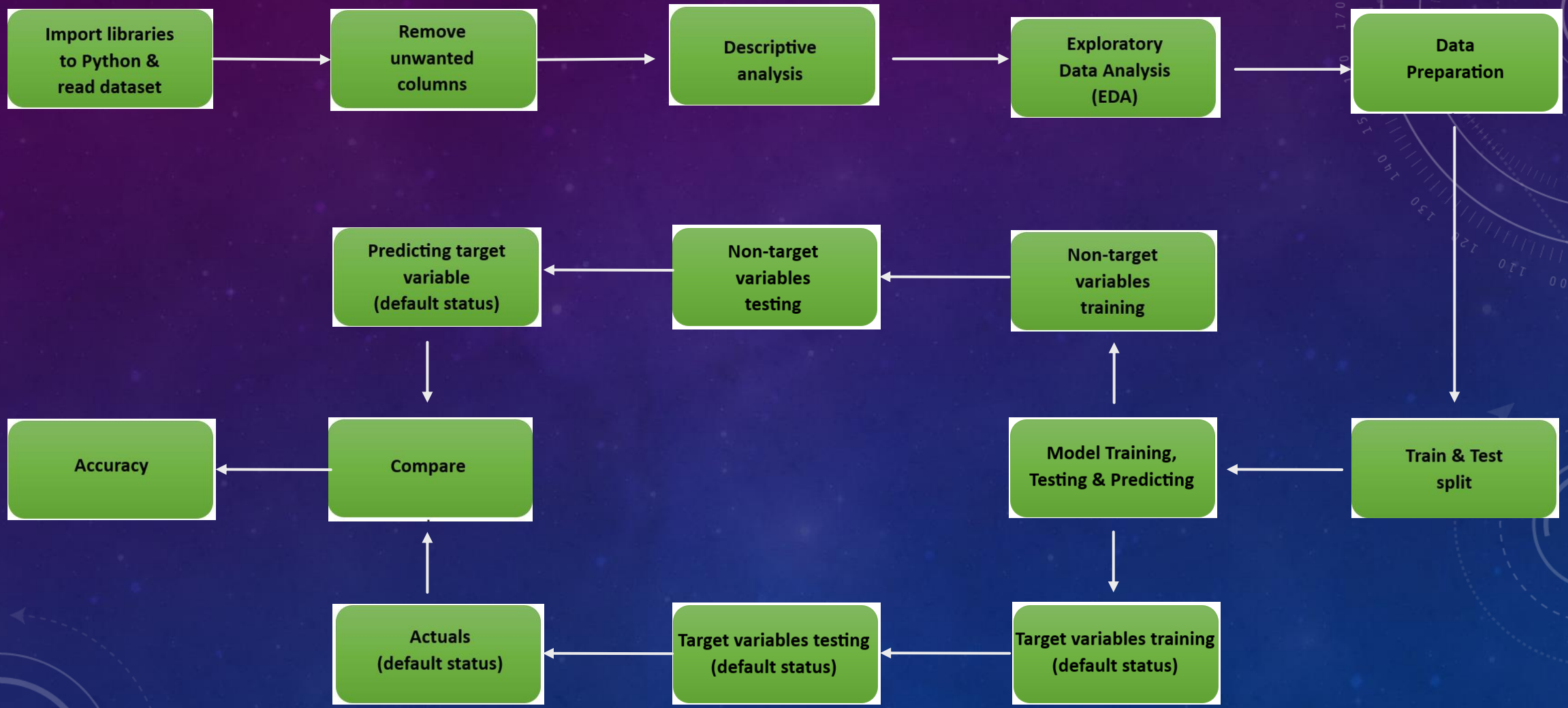
MIND-MAP


Import libraries to Python & read dataset
Remove unwanted columns
Descriptive analysis
Exploratory Data Analysis (EDA)
Data Preparation
Train & Test split
Model Training, Testing & Predicting
Target variables training (default status)
Target variables testing (default status)
Actuals (default status)
Compare
Accuracy
Predicting target variable (default status)
Non-target variables testing
Non-target variables training

DATA PREPARATION AND MODELLING

- For Python to be able to work with text columns, categories in all text columns are transformed into numbers (e.g. for income level: Low – 1, Medium – 2, High – 3)
- To make predictions, different machine models have to be trained and tested on the dataset before making predictions on the target variable/column for new/unseen data
- First, the dataset is broken down into two parts, train and test for both non-target variables and the target variable (default status)
- Both non-target and target variables are trained into the models to see what scenarios will result in the default status (target variable) being true and false
- Once trained, all needed variables are then tested with the model and the result of the tested target variable (default status) is called actuals (this is also the default status result in the dataset)
- Now we have the target variable actuals and the models have been trained and tested on the dataset, we can now start to make predictions
- To make predictions, the non-target variables that have been tested on the model are used to predict the target variable default status which are known as the predictions

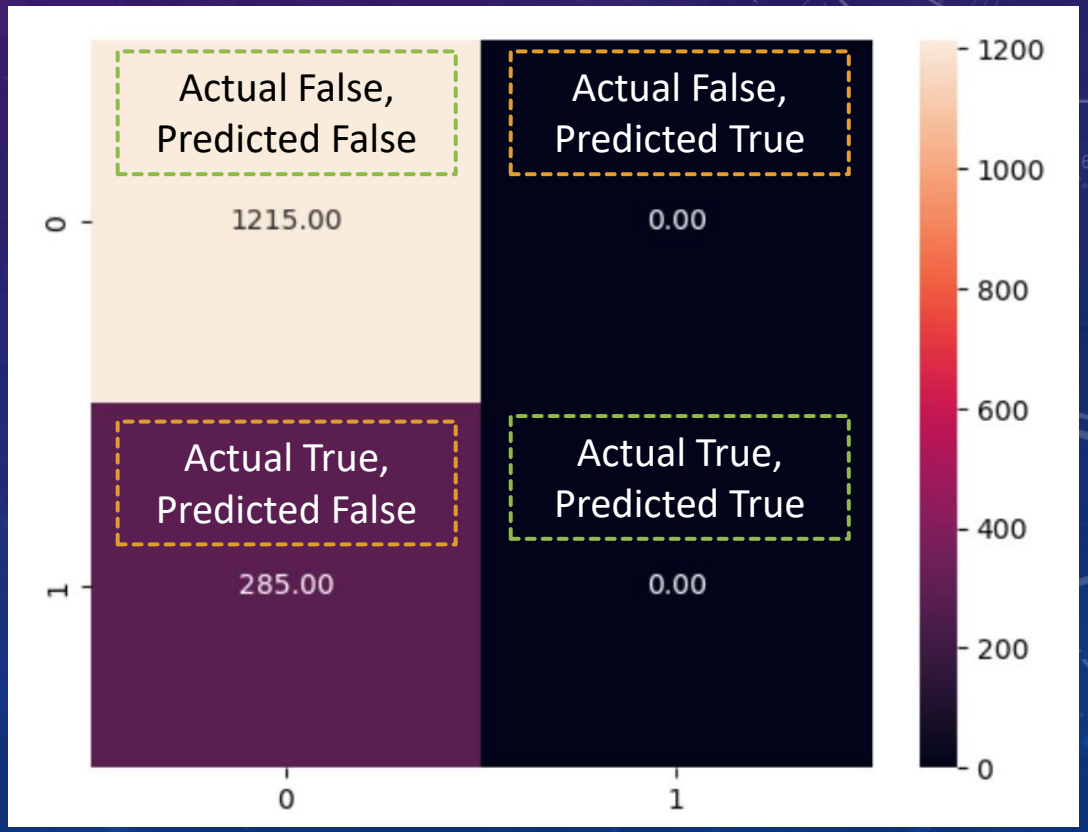
MIND-MAP


Import libraries to Python & read dataset
Remove unwanted columns
Descriptive analysis
Exploratory Data Analysis (EDA)
Data Preparation
Train & Test split
Model Training, Testing & Predicting
Target variables training (default status)
Target variables testing (default status)
Actuals (default status)
Compare
Accuracy
Predicting target variable (default status)
Non-target variables testing
Non-target variables training

RESULT INTERPRETATION

- Comparing the predictions with the actuals, we then get the accuracy of the model for the correct predictions
- For a model that is 70% training and 30% testing with 5000 records:
  1215 + 285 = 1500
  0.3 x 5000 = 1500
  (1215/1500) x 100 = 81%
- From the matrix, we can see for the target variable default status:
  - 1215 are predicted correctly for false
  - 285 are predicted to be false but in actual they are true

	Model	Accuracy
0	Logistic Regression	81.0
1	SVM	81.0
2	Decision Tree	81.0
3	Random Forest	81.0
4	GaussianNB	81.0
5	CategoricalNB	80.0



MODEL TESTING

- Different tests run by different models, solver algorithms and different model parameters to give the accuracy performance for the predictions at 70% of dataset for training and 30% for testing the model:

Train	Test	Model	Algorithm Solver Types	Iterations/Neighbors /Max Leaf Nodes	Accuracy
70% of 5000 is 3500 records/rows	30% of 5000 is 1500 records/rows	Logistic Regression	liblinear	5000 iterations	81%
			newton-cg		
			sag		
			saga		
			lbfgs		
		Support Vector Machines (SVC)	linear	251 iterations	42.73%
			poly	5000 iterations	81%
				-	
		K Neighbors Classifier	-	30 Neighbors	81%
		Decision Tree Classifier		2 Max Leaf Nodes	81%
		Random Forest Classifier		30 Max Leaf Nodes	81%
		Categorical Naïve Bayes (NB)		-	80%
		Gaussian Naïve Bayes (NB)			81%

	Model	Accuracy
0	Logistic Regression	81.0
1	SVM	81.0
2	Decision Tree	81.0
3	Random Forest	81.0
4	GaussianNB	81.0
5	CategoricalNB	80.0

MODEL TESTING

- Different tests run by different models, solver algorithms and different model parameters to give the accuracy performance for the predictions at 75% of dataset for training and 25% for testing the model:

Train	Test	Model	Algorithm Solver Types	Iterations/Neighbors /Max Leaf Nodes	Accuracy
75% of 5000 is 3500 records/rows	25% of 5000 is 1500 records/rows	Logistic Regression	liblinear	5000 iterations	80.72%
			newton-cg		
			sag		
			saga		
			lbfgs		
		Support Vector Machines (SVC)	linear	251 iterations	43.44%
			poly	5000 iterations	80.72%
				-	
		K Neighbors Classifier	-	30 Neighbors	80.72%
		Decision Tree Classifier		2 Max Leaf Nodes	80.72%
		Random Forest Classifier		30 Max Leaf Nodes	80.72%
		Categorical Naïve Bayes (NB)		-	79.60%
		Gaussian Naïve Bayes (NB)			80.72%

	Model	Accuracy
0	Logistic Regression	80.72
1	SVM	80.72
2	Decision Tree	80.72
3	Random Forest	80.72
4	GaussianNB	80.72
5	CategoricalNB	79.60

MODEL TESTING

- Different tests run by different models, solver algorithms and different model parameters to give the accuracy performance for the predictions at 80% of dataset for training and 20% for testing the model:

Train	Test	Model	Algorithm Solver Types	Iterations/Neighbors /Max Leaf Nodes	Accuracy
80% of 5000 is 3500 records/rows	20% of 5000 is 1500 records/rows	Logistic Regression	liblinear	5000 iterations	82%
			newton-cg		
			sag		
			saga		
			lbfgs		
		Support Vector Machines (SVC)	linear	251 iterations	52.50%
			poly	5000 iterations	82%
				-	
		K Neighbors Classifier	-	30 Neighbors	82%
		Decision Tree Classifier		2 Max Leaf Nodes	82%
		Random Forest Classifier		30 Max Leaf Nodes	82%
		Categorical Naïve Bayes (NB)		-	80.80%
		Gaussian Naïve Bayes (NB)			82%

	Model	Accuracy
0	Logistic Regression	82.0
1	SVM	82.0
2	Decision Tree	82.0
3	Random Forest	82.0
4	GaussianNB	82.0
5	CategoricalNB	80.8

RESULT INTERPRETATION

- Running at different training and testing percentages, all models give accuracy of around 81% for the predictions, therefore since all are similar accuracy, the first set is chosen at 70% training and 30% testing
- The dataset also falls under supervised learning therefore supervised learning models were used. Supervised learning means there are non-target variables and a target variable output (default status)
- All models in all train and test sets have correct predictions around 81%, this is due to the spread of data being very similar in every column

70% training & 30% testing

	Model	Accuracy
0	Logistic Regression	81.0
1	SVM	81.0
2	Decision Tree	81.0
3	Random Forest	81.0
4	GaussianNB	81.0
5	CategoricalNB	80.0

75% training & 25% testing

	Model	Accuracy
0	Logistic Regression	80.72
1	SVM	80.72
2	Decision Tree	80.72
3	Random Forest	80.72
4	GaussianNB	80.72
5	CategoricalNB	79.60

80% training & 20% testing

	Model	Accuracy
0	Logistic Regression	82.0
1	SVM	82.0
2	Decision Tree	82.0
3	Random Forest	82.0
4	GaussianNB	82.0
5	CategoricalNB	80.8

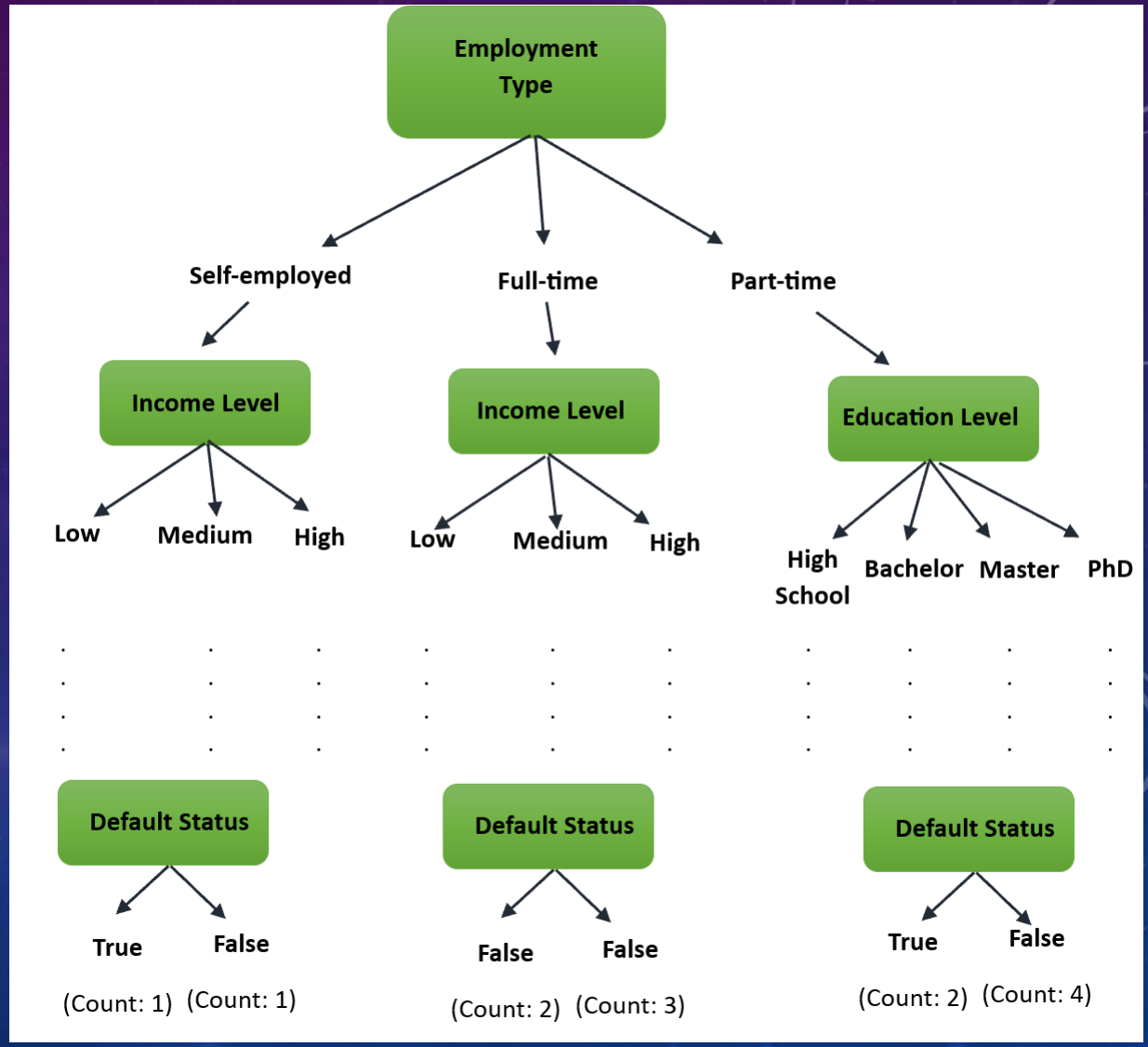
RESULT INTERPRETATION

PURPOSE OF MODELS:

- **Logistic Regression** – is used for predictive analysis and when the target variable is at least a binary (0&1) and explains the relationship between the target variable and the non-target variables, a s-curve graph
- **Support Vector Machines (SVM)** – finds the best boundary to separate two class of data, positive and negative. Can be linear or non-linear
- **Decision Tree** – A flow chart tree split into decisions based on the previous title, each split/branch is a result of the decision. Begins with one title and expands into many due to many decisions
- **Random Forest** – Combines multiple decision trees to make predictions, however may not perform well if one class has exceptionally more then the others and requires a lot of computational power
- **Gaussian Naïve Bayes (NB)** – calculates the probability of target variable (default status) getting true or false based on non-target variables to make predictions. Assumes numerical variables are normally distributed (symmetric)
- **Categorical Naïve Bayes (NB)** – similar to Gaussian Naïve Bayes and is used for categorical data instead, when categorical variables are discrete (different from one another)

RESULT INTERPRETATION

- Chosen Model: Decision Tree
- A flow chart tree split into decisions based on the previous title, each split/branch is a result of the decision. Begins with one title and expands into many due to many decisions
- Split to get best split based on previous title
- Goes through all non-target variables first according to best split then comes to a conclusion on the target variable (default status)
- Counts how many true and false and these are the predictions which are compared with the actuals to get the accuracy



PROS & CONS OF DECISION TREE

PROS:

- Able to handle categorical and numerical variables
- Performs classification without needing much computation
- Easy to read
- Informs of which variables are most important in predicting due to best fit

CONS:

- Does not work well for predicting target variables that are numerical
- Is vulnerable to errors in classification problems where the target variable has many categories and small number of training examples
- A small change in the input will cause the decision tree to change completely

MACHINE LEARNING

- Machine Learning is a smaller part of AI and involves building systems that can learn from data and make predictions without being programmed to
- It is able to handle very large datasets where normal methods don’t do well
- Machine Learning allows for automation of tasks that may take too much time or is complex, increasing the productivity
- Machine Learning is very good at predicting using data that is readily available
- Allows for personalizations according to business needs
- Able to detect complex patterns and anomalies which may not have been detected by human analyst
- Improves with experience
- Is able to help businesses make better decisions based on predictions and have an advantage over competitors

CONCLUSION

- Certain variables when plotted default status based on income level for example, from the graph it looks like it would not go to as we expect, but this is because the variable may be influenced by other variables as when we look at the dataset, many variables are taken into account to get the target variable default status.
- The accuracy to predict correctly for all models in all train and test sets are around 81% due to the spread of data being very similar in every column
- All models in all train and test sets give around the same accuracy, therefore the first train test set at 70% train and 30% test is chosen
- The Decision Tree model is chosen as it is suitable for this dataset as seen from the model comparisons and pros and cons
- Machine Learning improves with experience and has many advantages and uses including predicting very well by using data readily
- This provides businesses to make better decisions based on the predictions, avoid more financial loss and have a competitive edge over competitors