Summary Report of the '*Lead Scoring Case Study*'

- The first task was to understand the Problem Statement, and interpret the business objectives and the relevant statistics accurately.
- In this case, an education company selling online courses, wants to improve sales efficiency through optimal use of resources available and maximise sales.
- The concept of Sales funnel was applied in this scenario, where for every 100 leads pursued by the sales team, around 30 were converted into a sale. What is construed as a lead may also vary based on the business strategy, which in this case is either through past referrals, or filling up a form providing details of email address or phone number.
- The company requires us to build a model wherein we need to assign a lead score between 0 and 100 to each of the leads which can be used by the company to target potential leads. A higher score would mean that the lead is hot, i.e. is most likely to convert whereas a lower score would mean that the lead is cold and will mostly not get converted.
- The company could then deploy more resources to nurture the hot leads to maximise sales.
- Interpreting the target lead conversion rate given by the CEO of 80% was a challenge. In particular whether to interpret it as a Precision or Recall.
- Since, ideally the company's objective would be to not miss out on any potential lead, we interpreted it as Precision of 80%. Where-in it means out of every 100 leads that the model predicts as hot, around 80 are actually converted.
- The next step was to understand and analyse the data used to assign the Lead Score and hence map the business objective into a data science problem.
- Since the business objective is to classify whether a particular lead is hot or otherwise (binary prediction) logistic regression seemed an obvious choice, which also specifically mentioned as a part of the problem statement.
- The next step was to analyse the data given and prepare it to use for modelling. In particular,
    - Which columns to drop
    - Which rows to exclude
    - Handling outliers
    - How to impute missing values and
    - The rationale for the above treatments were done.
- The next step was Model Building which was relatively straight-forward using the Python packages/ procedures for logistic regression –
    - Used sklearn model selection to split data into train and test sets
    - Used sklearn preprocessing to scale the features (independent variables)
    - Used stats models to build the create the logistic regression model on the train data set and dropped irrelevant features
    - Used RFE to select the 7 most important features
    - Checked VIF of features to eliminate multi-collinearity
    - Assessed the model with remaining columns and removed any columns with high p-values.
    - Utilized the precision-recall curve to find out the optimal cut-off for lead score for required precision value of 80 percent
    - Finally, checked the precision value of the model against test data set to check if it meets the required criterion.
- Lastly, an important learning was to interpret the model results back to the business context in the form of recommendations based on resource availability.