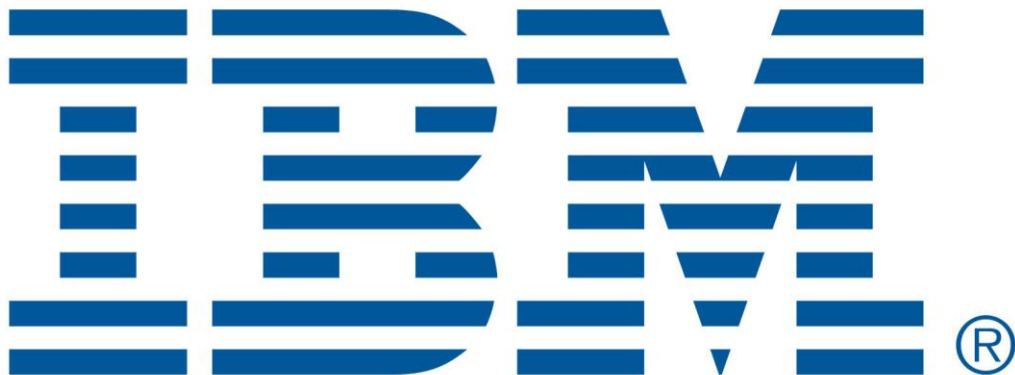


IBM Employee HR Attrition

3X Data Mining

Mohammed Topiwalla, Patricia Londono, Sanchita Kumari,
Valerio Trotta, Igor Pedevani, Millicent Ottchere



Problem Statement

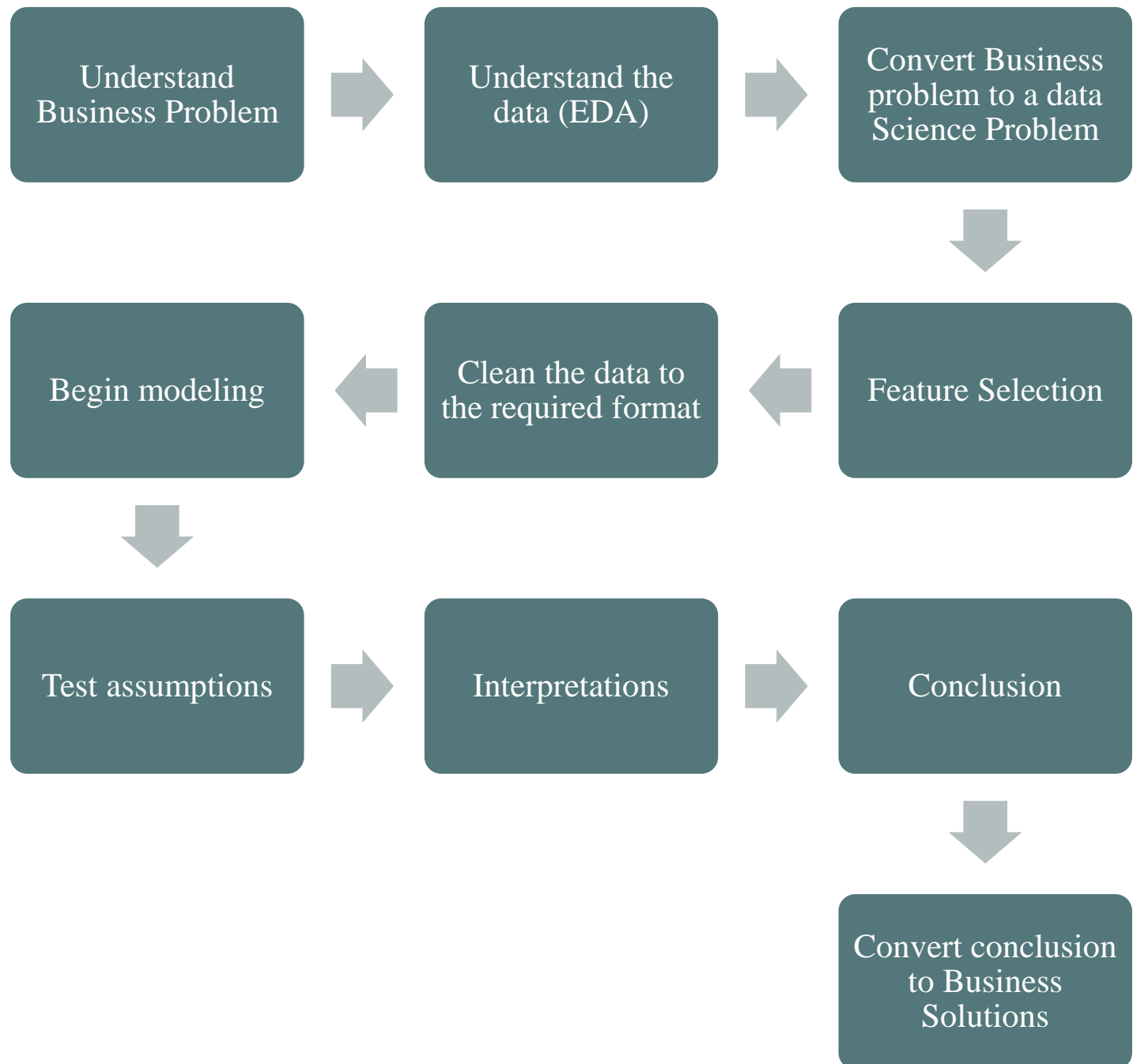
How can we reduce IBM company's attrition rate by predicting if a candidate will exit in India within the year?

- **Specific :-** To Indian geography in IBM
- **Measurable:-** To reduce attrition rate(By at least 5%)
- **Action oriented:-** Reduce employee attrition & suggest employee engagement & satisfaction programs
- **Relevant:-** Direct impact on company's top and bottom line
- **Time bound :-** 12 months

Our client is IBM a leading firm and in the IT sector. It is recently facing a steep increase in its employee attrition . Employee attrition has gone up from 14% to 25% in the last 1 year . We are asked to prepare a strategy to immediately tackle this issue such that the firm's business is not hampered and also to propose an efficient employee satisfaction program for the long run. Currently, no such program is in place . Further salary hikes are not an option.

The attrition problem is not only unique to ibm but to other IT companies such as Infosys, India's second largest IT services company, that is also battling high attrition, with a peak attrition of 20.4 % in the October-December quarter of FY15.

Methodology to solve the problem



Data cleansing and Smote

Our data set had no missing values therefore no special treatment was required, however since we are running a classification algorithm

- A. We had many categorical variables which we needed to convert into dummy variables or ordered integer variables
- B. There was a massive class imbalance to the proportion of 83:16 , in order to tackle this we understood that even without any algorithm if we predicted that the employee wouldn't leave we would still be right 83% of the time. So any model we make should be able to provide a better accuracy than this or else it would not be worth it.
- C. The second solution was to balance the classes (A paper by Chawla led us the way - > <https://www.jair.org/media/953/live-953-2037-jair.pdf>)
- D. With the help of smote and code help from stack overflow we achieved Smote
 - a. What smote does is simple, it first of all takes the class with low proportion and artificially boosts its values to increase the number of records, it tries to generate new rows by replicating them in the bases of a range of values already present making various random combinations
 - b. It takes the overpopulated proportion and performs sampling to pick up those rows which are the most representative of the population and then it uses them to reduce this proportion
 - c. This is repeated till the classes are nearly balanced
- E. Two techniques of smote from 2 different packages were used therefore we have 2 smote data sets

Who made this – Mohammed

Who reviewed this – Sanchita , Igor

Classification

In order to create an early warning system, it was essential to perform a supervised classification.

We had 3 sets of data

1. The original data (With one hot encoding)
2. Data generated from SMOTE (Set1)
3. Data generated from SMOTE (Set 2)

Before running any model it was ensured that the best parameters were selected on the basis of validation set or cross validation set. The splits were always on the basis of 50:25:25 (Training:Validation:Testing). All the sampling is stratified sampling in order to ensure perfect class balance

The approach was to run simple models like

- A. Decision tree
- B. Logistic Regression

On both these models we would train each of the 3 data sets

After the process was completed we trained data on complex models like

- A. Support vector machines
- B. Artificial Neural Network
- C. Random forest
- D. Extreme Gradient Boosting

The output for running all those models are

Model	Data	ACC	ROC	Kappa
Logistic Regression	All data	88.04%	0.67	0.42
	Smote 1	81.52%	0.82	0.63
	Smote 2	79.25	0.79	0.58
	Fs via Backward elimination	86.41%	0.64	0.34
	Fs via CFS subset evaluation	85.32%	0.52	0.05
Decision Tree	All Data	84.23%	0.61	0.25
	Smote 1	85.57%	0.86	0.71
	Smote 2	92.22%	0.92	0.84
	Fs via Backward elimination	83.96%	0.64	0.31
	Fs via CFS subset evaluation	84.23%	0.61	0.25
Random Forest	All data	87.72%	0.59	0.26
ANN	All data	85.86%	0.64	0.33
SVM	All data	83.9%	0.58	0.19
XGBOOST	All data	88.04%	0.66	0.4
	Smote 1	91.41%	0.92	0.82
	Smote 2	87.35%	0.87	0.74

The interpretation is simple SMOTE data is the winner and decision tree stole the show!

But what are we missing at times it may be necessary to explain the HR our results, why our model believes our person is planning to leave, in that case decision tree or logistic regression are the best options.

However, if why decision made was not required we can simply use more powerful models like XGBoost, because as data increases the accuracy of XGBOOST will be higher than a decision tree because it's a powerful ensemble algorithm.

Who made this – Mohammed

Who reviewed this – Patricia , Valerio

Clustering

For clustering our goal was to check if our dataset falls into 2 perfect clusters or not. If not what additional insights does it provide.

A K-means clustering algorithm was used on the original dataset.

1. The silhouette index tells us about the effectiveness of the classification and using the K-means (with two clusters) on the last set of variables, we obtained a 0.7 (Silhouettes average width).
2. We all pointed out that it was a very good result, plus upon reviewal of the work everyone correctly pointed that we are setting 2 as the number of cluster for testing matters, but the elbow plot at the beginning of the scripts shows that 3 or 4 is the best number to choose.
3. In order to explain such a high silhouette score, a reasoning was done to check why people in the clusters are likely to leave or not leave. Do they have something in common? That would have required further analysis on the clusters. And a future scope of research could be done on this data point

We wanted to work using clustering on this project because we wanted to improve our knowledge on the project and we still feel there is much to learn.

Who made this – Valerio

Who reviewed this – Sanchita , Mohammed

Challenges faced –

- The main difficulty was plotting. When it comes to clustering, plots become hard to code and interpret.
- Luckily enough all group members supported the clustering and a great repository of examples from this course.
- Ultimate struggle faced was also with time. A lot of time was spent learning and correct methods of coding. A lot of simple random bugs took up more time than it should have.

Conclusion of Author-

I wish to take this project out again in the future and try to answer the unanswered questions.