AV MEET UP – MUMBAI

How to Interpret Machine Learning Model (Unboxing Black box Algorithm To White Box)

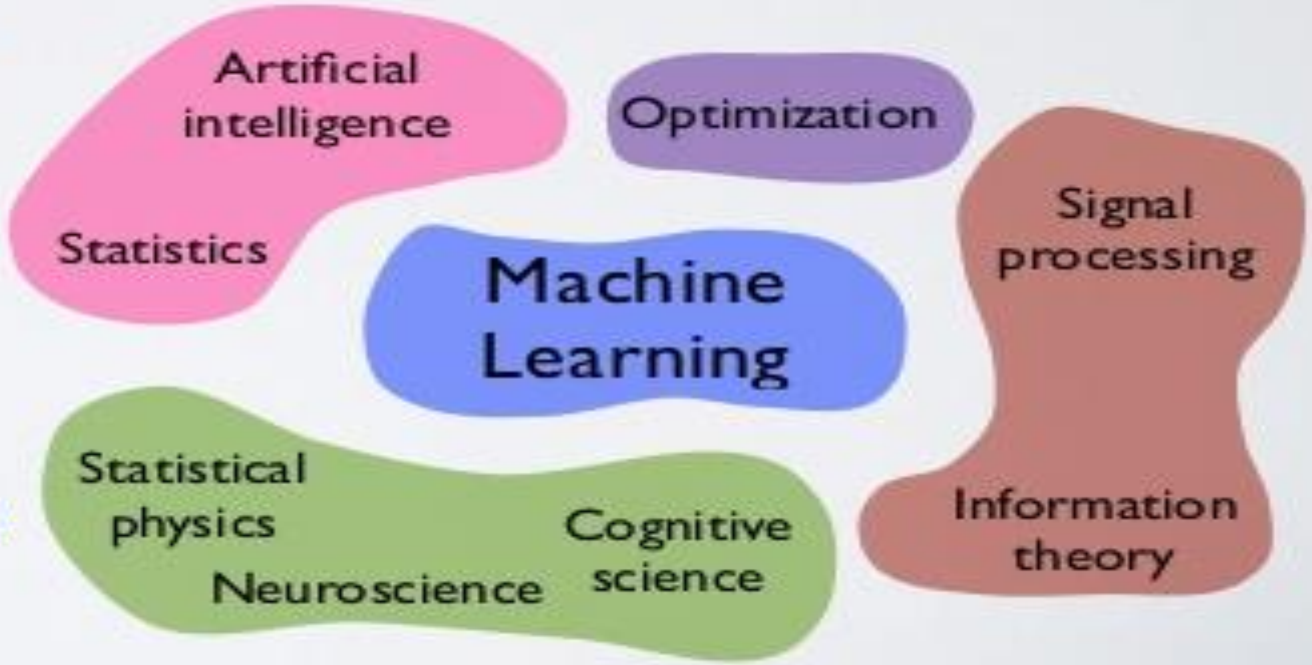
Arihant Jain

Email – arihant.jec2013@gmail.com
Mobile – 8586970038
LinkedIn - https://www.linkedin.com/in/arihantjain15/

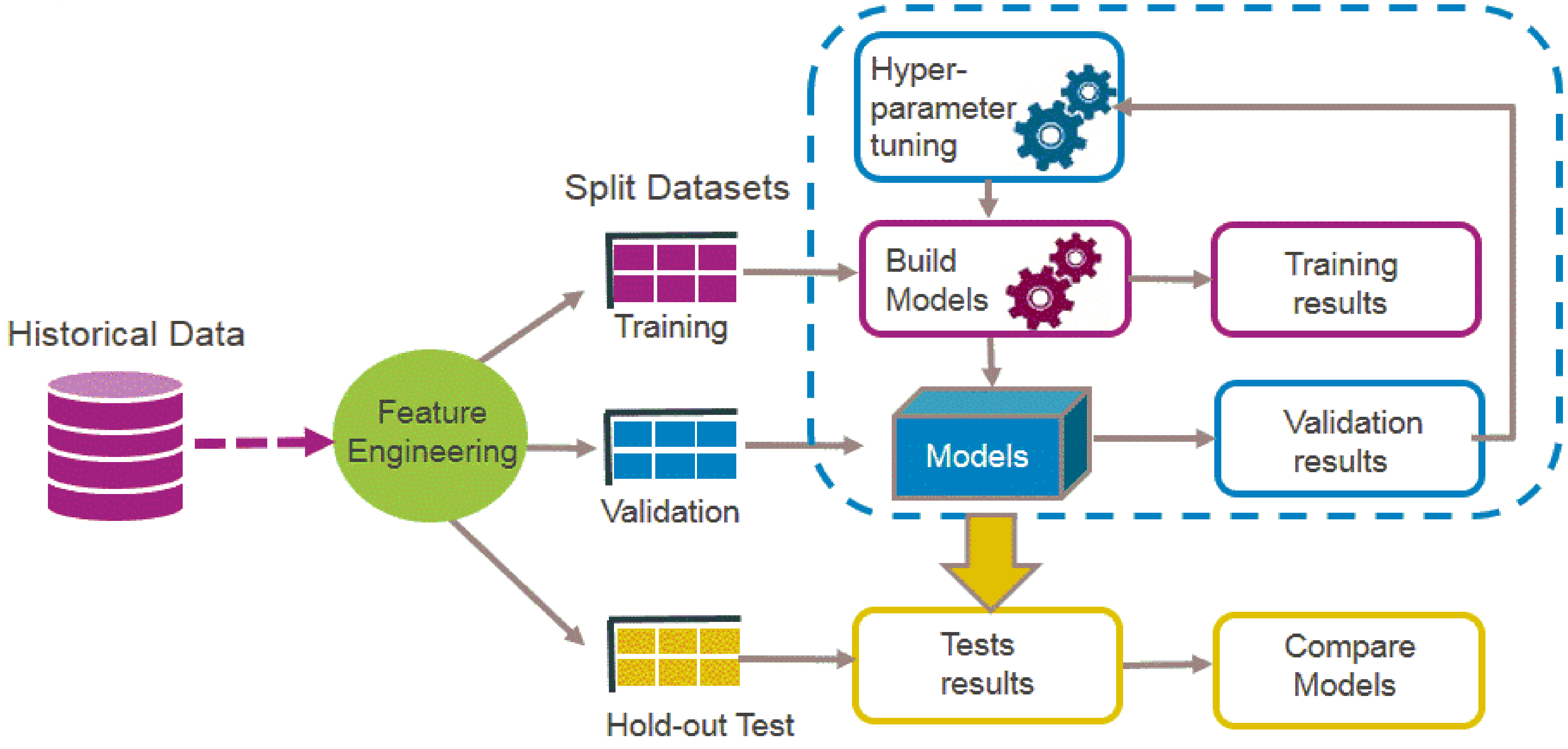
❑ Agenda

WHAT IS MACHINE LEARNING?

- "The science of getting computers to act without being explicitly programmed" - Andrew Ng (Stanford/Coursera)
  - part of standard computer science curriculum since the 90s
  - inferring knowledge from data
  - generalizing to unseen data
  - usually no parametric model assumptions
  - emphasizing the computational challenges



Flow of Machine Learning Model



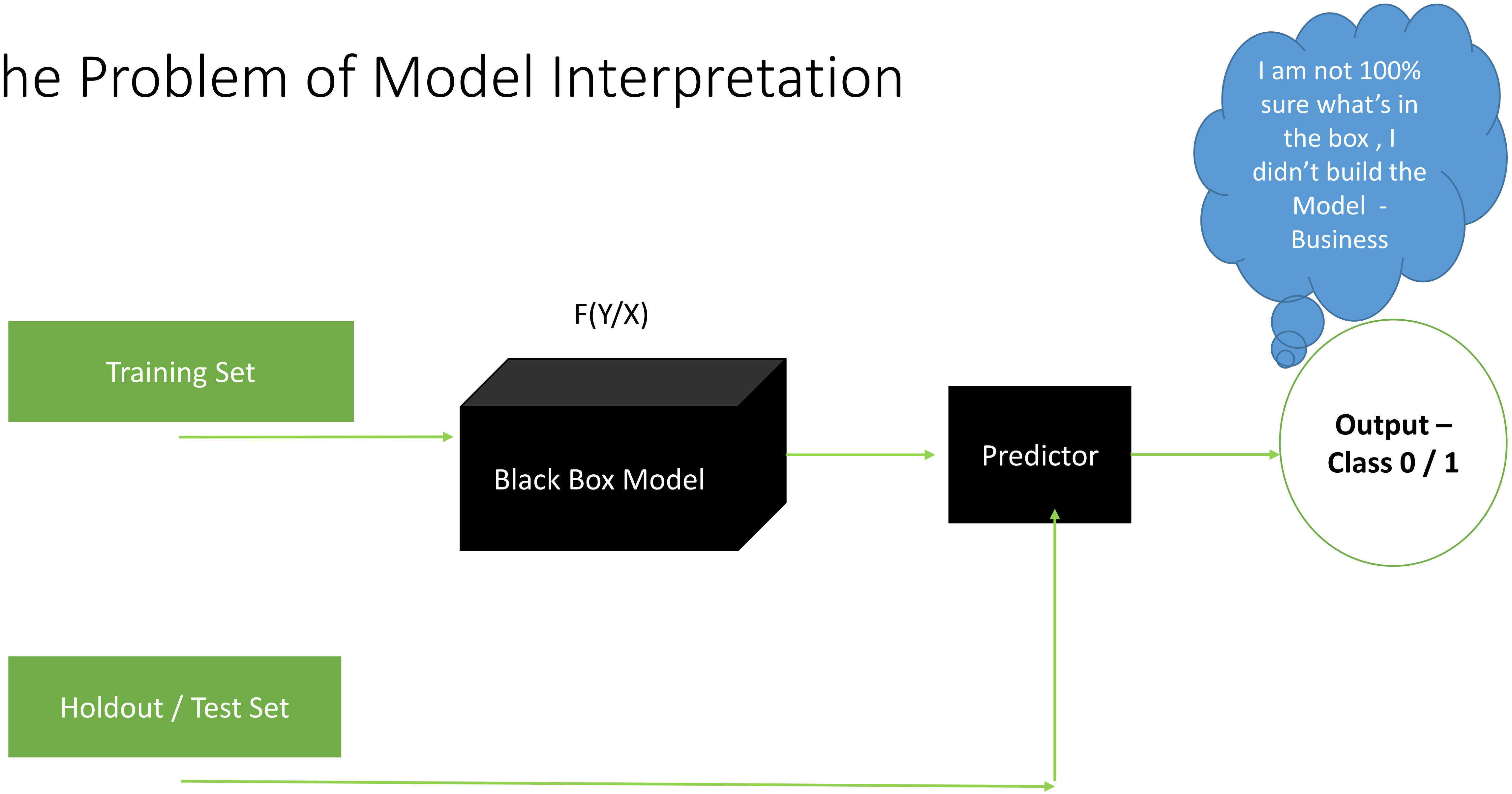
What is Model Interpretation?

- An *extension of model evaluation* that helps to foster a better understanding of a model's learned decision policies

- Ability to explain and present a model in a way that is human understandable/ Such interpretation -

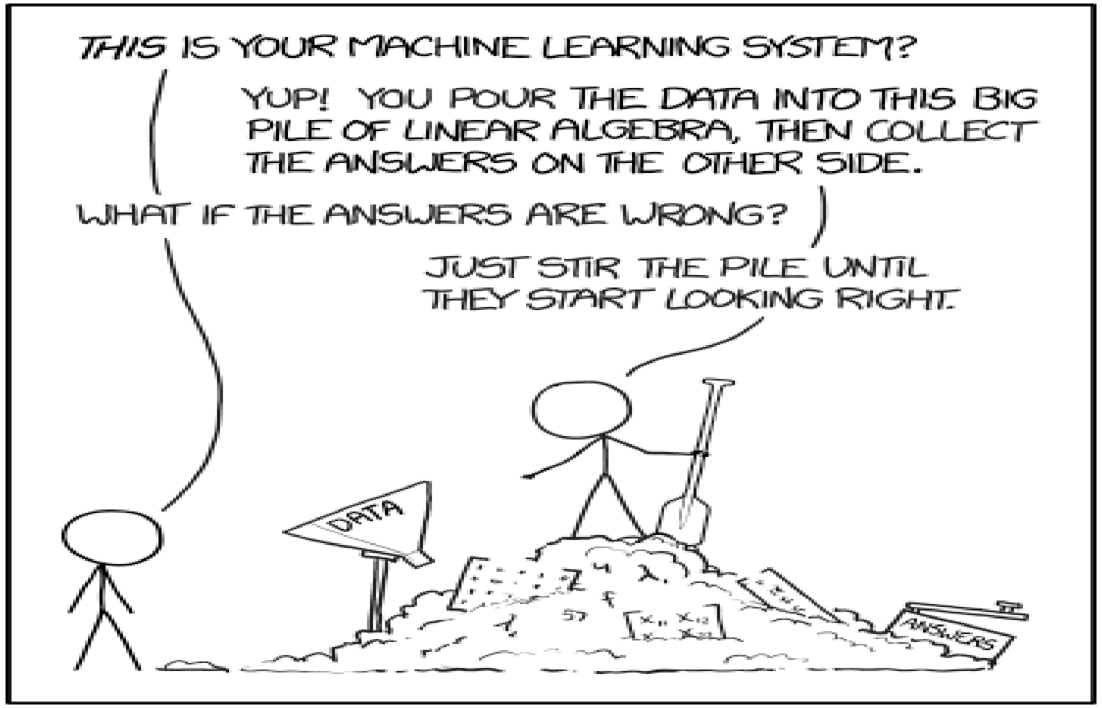
- Human understandable: The model's result is self descriptive & needs no further explanation.

The Problem of Model Interpretation



Why of Interpreting machine learning models

- Whatsoever the end goal of our data science solutions, an end-user /Business user will always prefer solutions that are interpretable and understandable
- As a data scientist we will always benefit from the interpretability of our model to validate and improve your work
- It is easy to lose our self in experimenting with state-of-the-art techniques when building models, being able to properly interpret your findings is an essential part of the data science process
- Identity and mitigate bias
- To Improve Generalization
- To improve Performance
- Ethical & Legal Reasons
- Aligned with context of the problem as per business needs


THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS

What of Interpreting machine learning models

With model interpretation, we want to answer the following questions:

- **Why** did the model behave in a certain way?
- What was the reason for false Negatives? What are the relevant variables driving a model’s outcome e.g. Credit Risk Models , fraud detection, spam detection?
- How can we trust the predictions of a “black box” model? Is the predictive model biased?

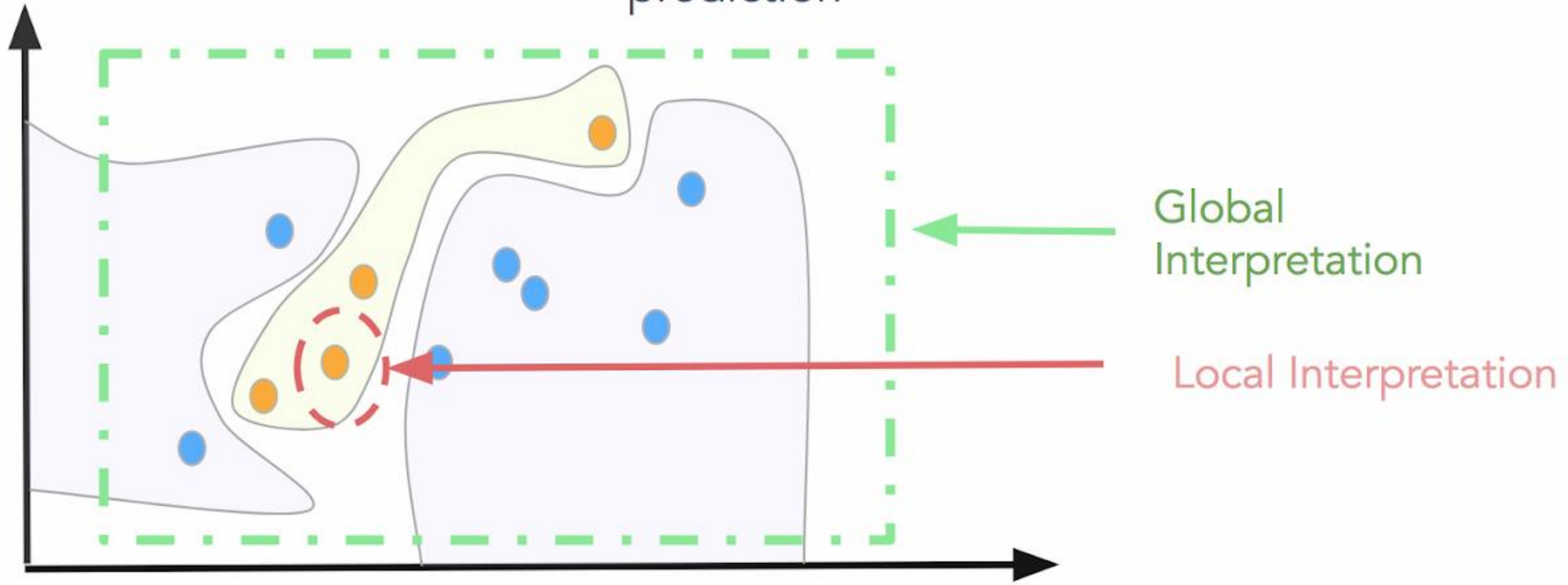
Scope Of Interpretation

Global Interpretation

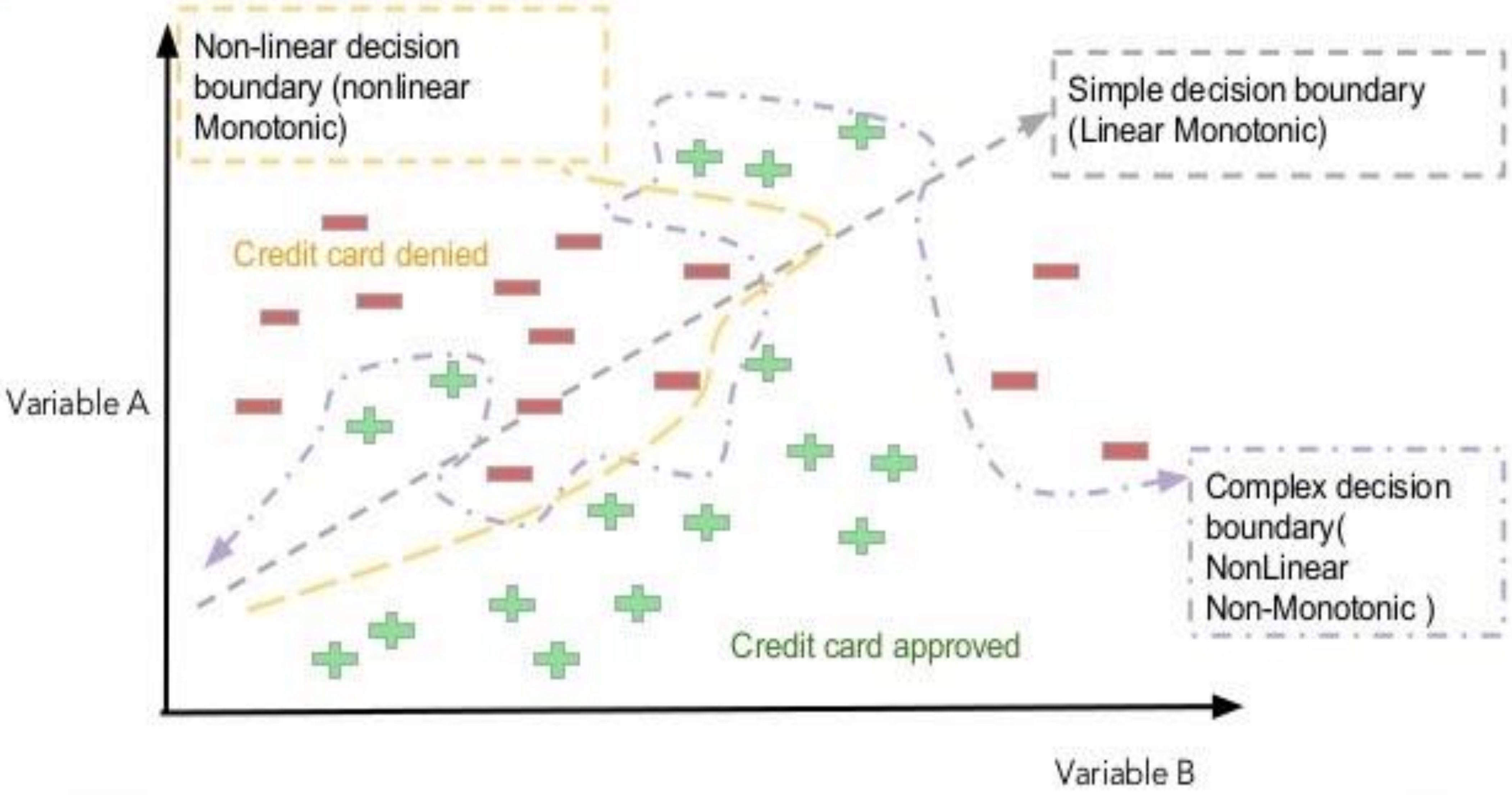
Being able to explain the conditional interaction between dependent(*response*) variables and independent(*predictor, or explanatory*) variables based on the complete dataset

Local Interpretation

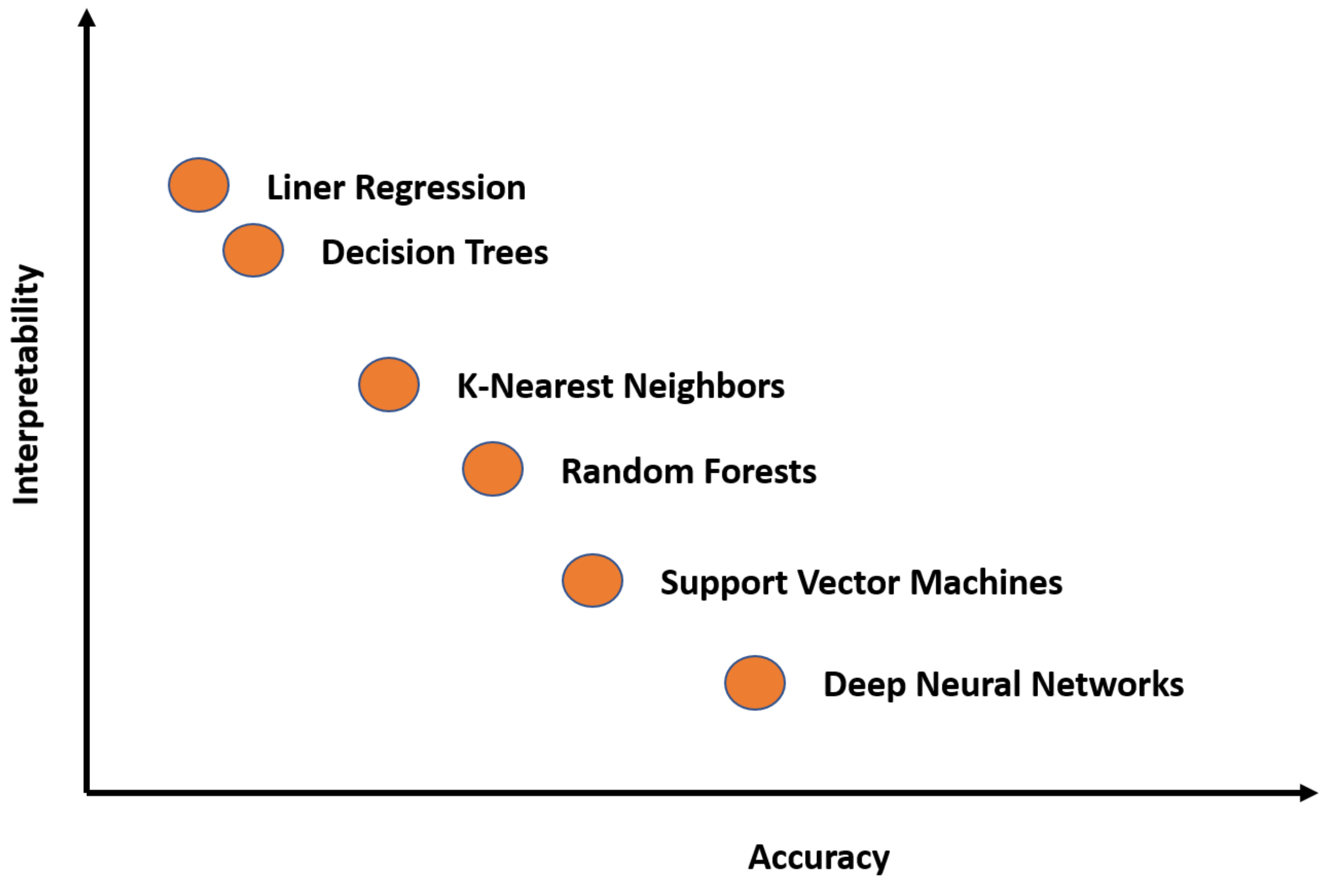
Being able to explain the conditional interaction between dependent(*response*) variables and independent(*predictor, or explanatory*) variables wrt to a single prediction



PERFORMANCE VS. INTERPRETABILITY


Non-linear decision boundary (nonlinear Monotonic)
Simple decision boundary (Linear Monotonic)
Credit card denied
Variable A
Complex decision boundary(NonLinear Non-Monotonic)
Credit card approved
Variable B

Accuracy & Interpretability - Algorithms

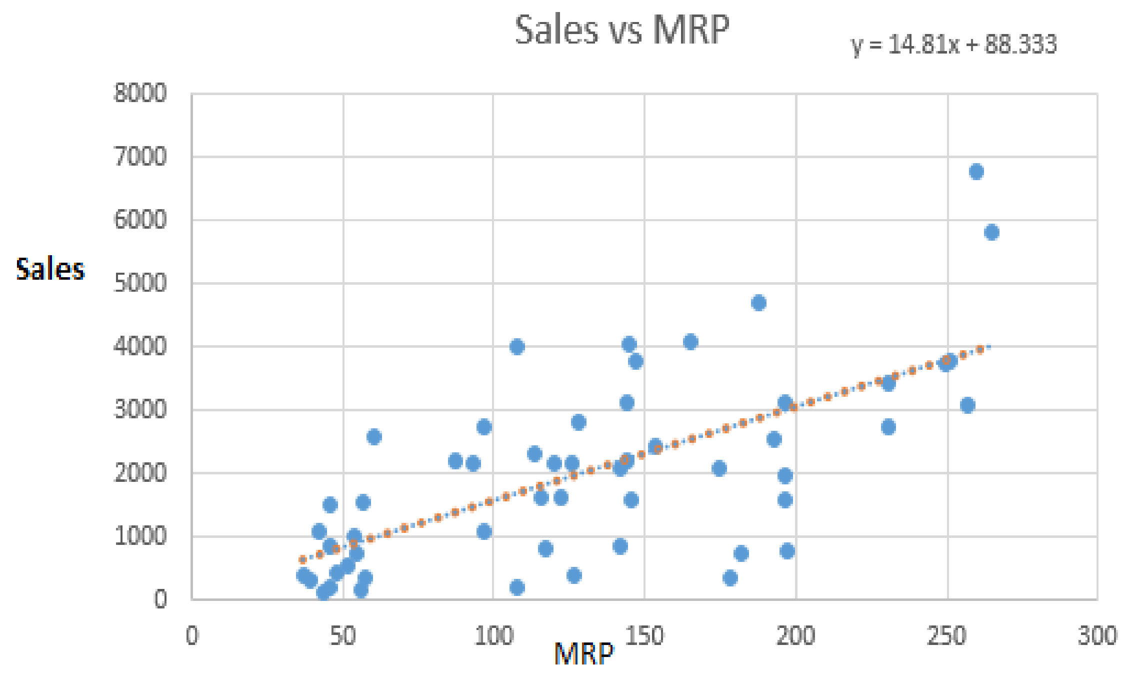


Linear Regression

Linear regression is the simplest and most widely used statistical technique for predictive modeling. It basically gives us an equation, where we have our features as independent variables, on which our target variable [sales in our case] is dependent upon. So what does the equation look like? Linear regression equation looks like this:

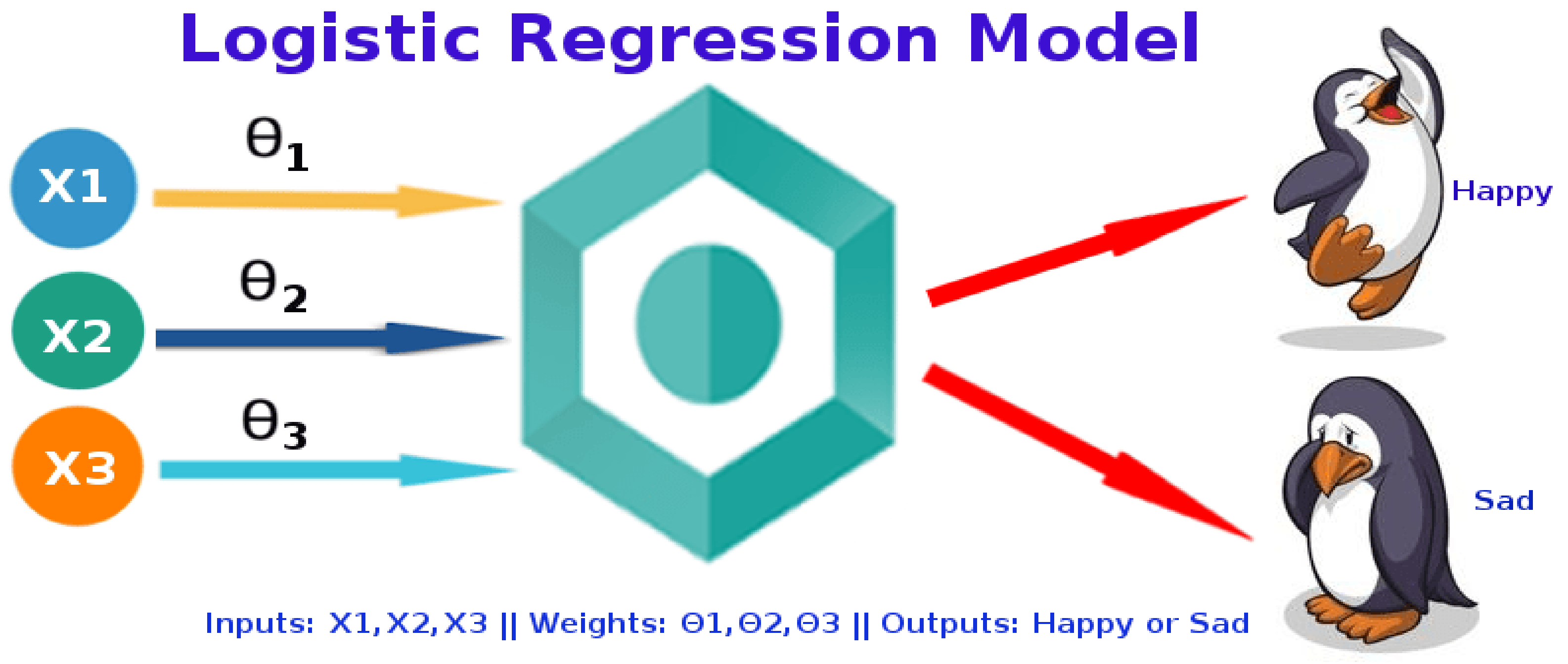
$$Y = \theta_1 X_1 + \theta_2 X_2 + \ldots \theta_n X_n$$

Sales of a product increases with increase in its MRP. Therefore the dotted red line represents our regression line or the line of best fit. But one question that arises is how you would find out this line

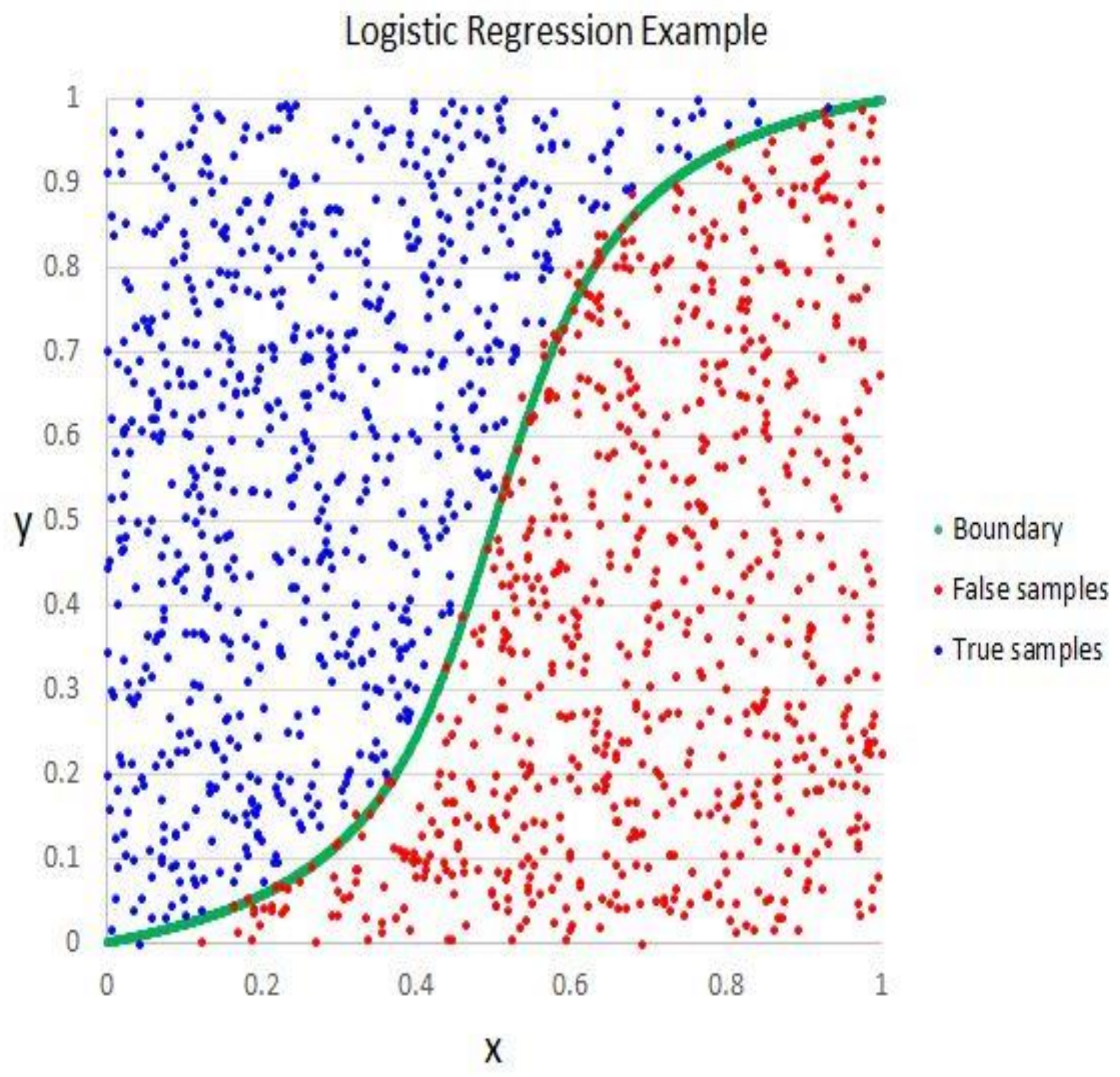


What is Logistic Regression ?

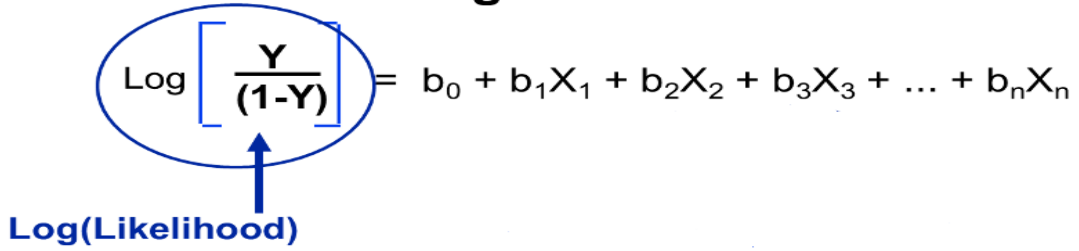
- Logistic Regression is a classification algorithm
- It is used to predict a binary outcome (1 / 0, Yes / No, True / False) given a set of independent variables
- You can also think of logistic regression as a special case of linear regression when the outcome variable is categorical, where we are using log of odds as dependent variable
- In simple words, it predicts the probability of occurrence of an event by fitting data to a logit function


Logistic Regression Model
X1
X2
X3
θ1
θ2
θ3
Happy
Sad
Inputs: X1,X2,X3 || Weights: Θ1,Θ2,Θ3 || Outputs: Happy or Sad

How to Interpret Logistic Regression?



The Logistic Function



$$\log\left[\frac{Y}{(1-Y)}\right] = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + \dots + b_nX_n$$

B Coefficient	e^B (Odds Ratio)	Examples	Interpretation
Positive	Greater than 1	$e^{.10} = 1.105$ $e^{2.3} = 9.97$	For each 1-unit increment on the predictor, the odds of the event increase (multiplied by something greater than 1)
Zero	1	$e^0 = 1$	Predictor has no effect (neutral); with each 1-unit increment on predictor, keep multiplying odds by 1
Negative	Less than 1	$e^{-.20} = .82$ $e^{-3.6} = .03$	For each 1-unit increment on the predictor, the odds of the event decrease (multiplied by something less than 1)

Linear models have drawbacks

- Underlying data is often non-linear
- Feature binning: potentially massive increase in number of features

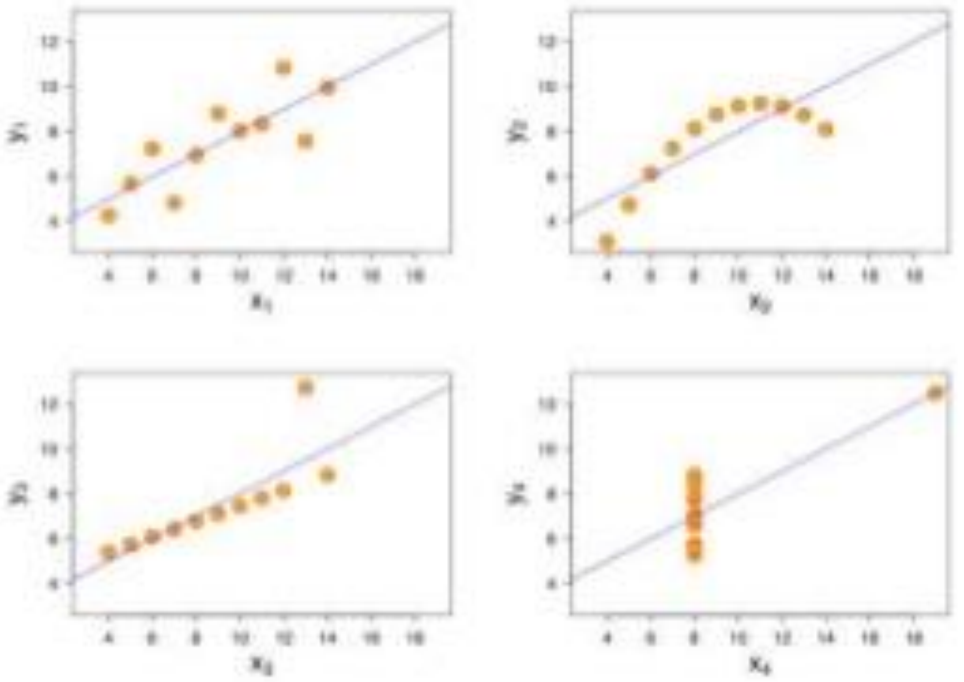
- Basis expansion (non-linear transformations of underlying features)

$$Y = 2x_1 + x_2 - 3x_3$$

vs

$$Y = 2{x_1}^2 + 3x_1 - \log(x_2) + \sqrt{x_2}x_3 + \ \ldots$$

In both cases performance will be traded for interpretability



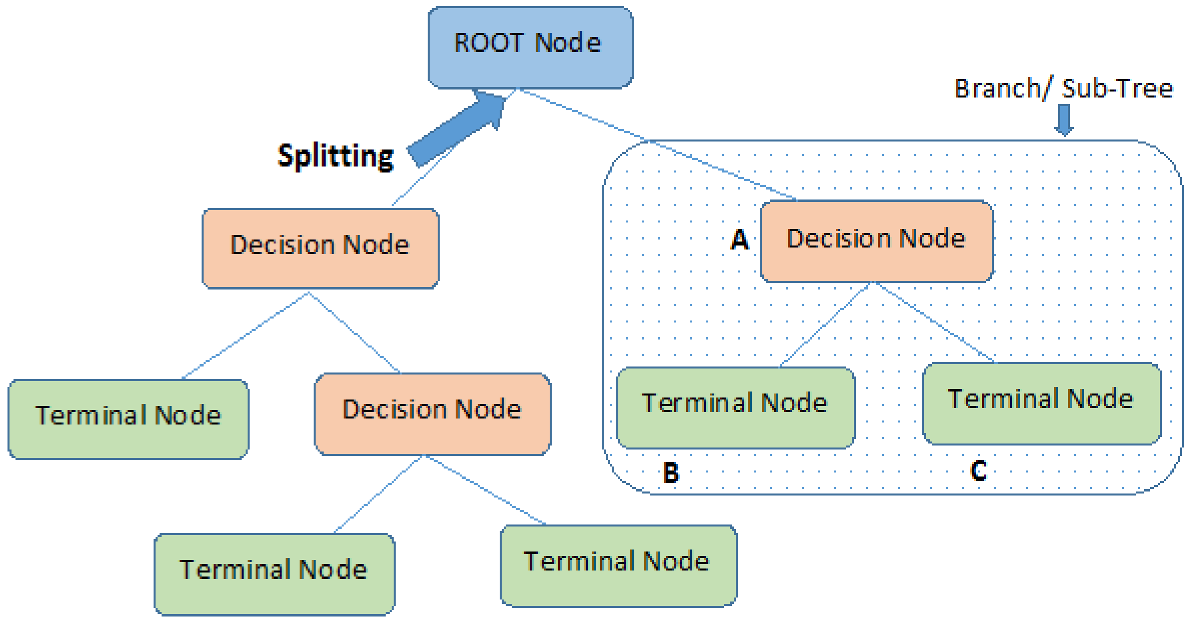
To Overcome the challenges We have in Linear Models , We can use Tree Based Algorithms which captures Non Linearity ..

Let’s discuss **Tree Based Algorithms ..**

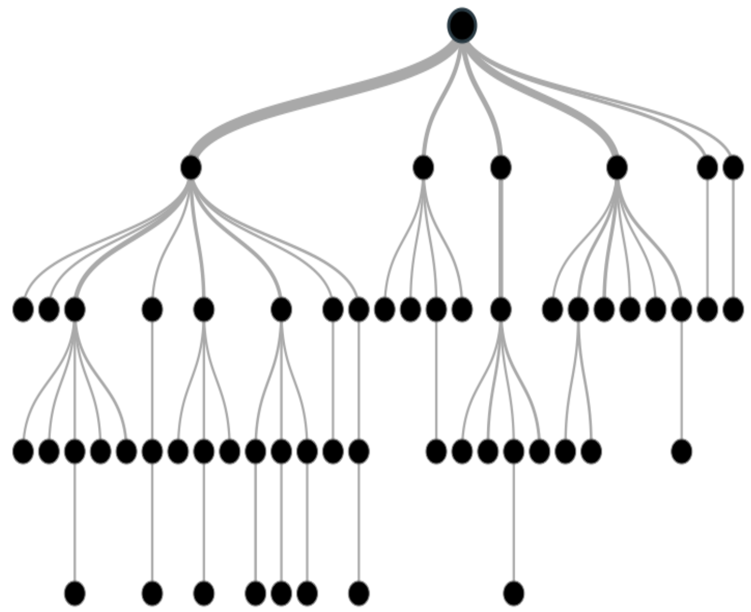
As we Move towards Tree Based Algorithms , Our Model Accuracy Increases but Interpretability of Model Decreases .

Decision Tree

Decision tree is a type of supervised learning algorithm (having a pre-defined target variable) that is mostly used in classification problems. It works for both categorical and continuous input and output variables. In this technique, we split the population or sample into two or more homogeneous sets (or sub-populations) based on most significant splitter / differentiator in input variables.



Note:- A is parent node of B and C.

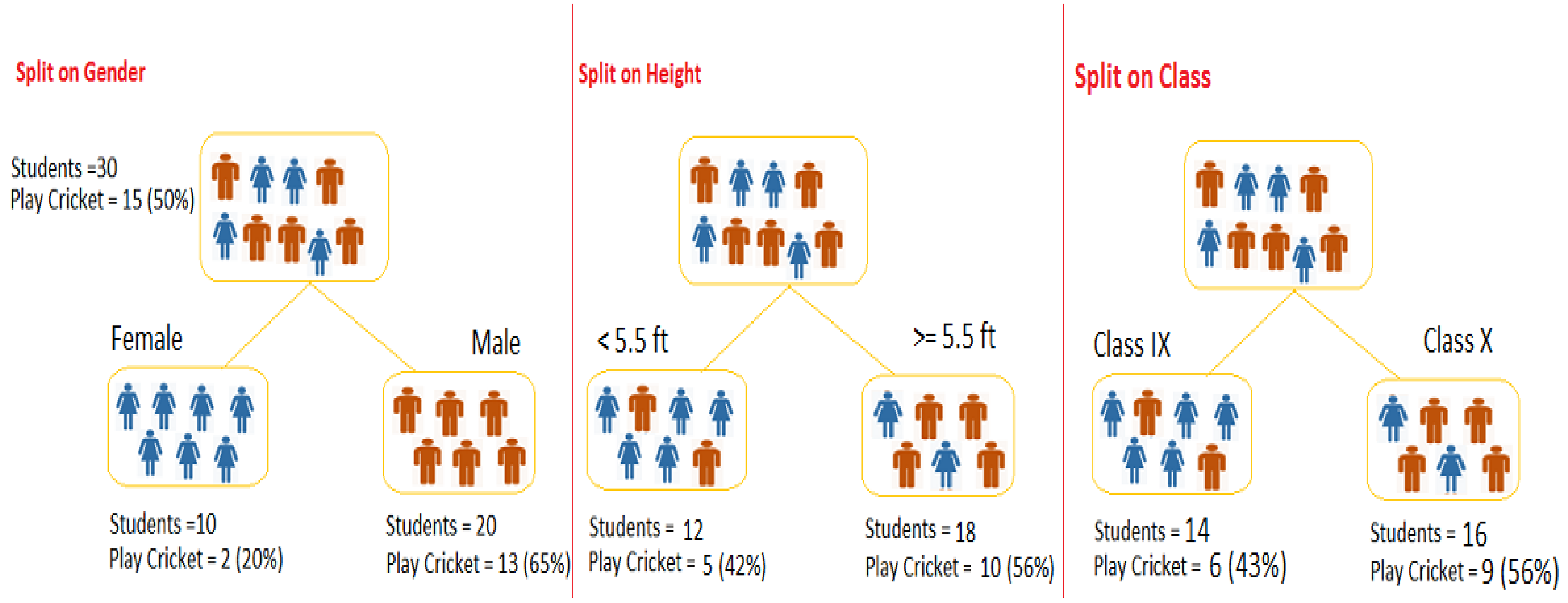


Example of Decision Tree

Let's say we have a sample of 30 students with three variables Gender (Boy/ Girl), Class(IX/ X) and Height (5 to 6 ft.). 15 out of these 30 play cricket in leisure time. Now, I want to create a model to predict who will play cricket during leisure period? In this problem, we need to segregate students who play cricket in their leisure time based on highly significant input variable among all three.

This is where decision tree helps, it will segregate the students based on all values of three variable and identify the variable, which creates the best homogeneous sets of students (which are heterogeneous to each other). In the snapshot below, you can see that variable Gender is able to identify best homogeneous sets compared to the other two variables.

As mentioned above, decision tree identifies the most significant variable and it's value that gives best homogeneous sets of population. To do this, decision tree uses various concepts (Entropy, Information Gain etc.) Internally



Math's Of Split

- **Split on Gender:**
  - Calculate, Gini for sub-node Female = (0.2)*(0.2)+(0.8)*(0.8)=0.68
  - Gini for sub-node Male = (0.65)*(0.65)+(0.35)*(0.35)=0.55
  - Calculate weighted Gini for Split Gender = (10/30)*0.68+(20/30)*0.55 = **0.59**

- **Similar for Split on Class:**
  - Gini for sub-node Class IX = (0.43)*(0.43)+(0.57)*(0.57)=0.51
  - Gini for sub-node Class X = (0.56)*(0.56)+(0.44)*(0.44)=0.51
  - Calculate weighted Gini for Split Class = (14/30)*0.51+(16/30)*0.51 = **0.51**

Above, We can see that Gini score for ***Split on Gender*** **is higher than** ***Split on Class,*** **hence, the node split will take place on Gender.**

Before we deep Dive in Other Algorithms like Random Forest & Xgboost and different ways to interpret this Black Box Models.

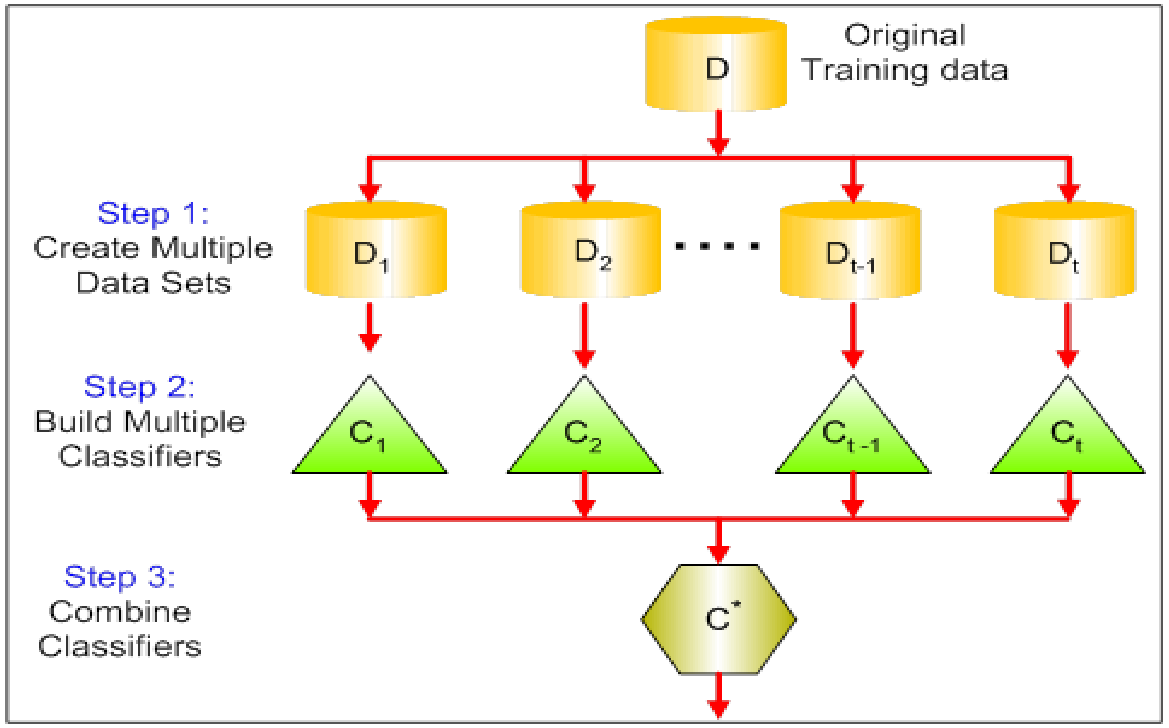
Lets first Understand Bagging & Boosting..

Bagging

Bagging is a technique used to reduce the variance of our predictions by combining the result of multiple classifiers modeled on different sub-samples of the same data set. The following figure will make it clearer:

The steps followed in bagging are:

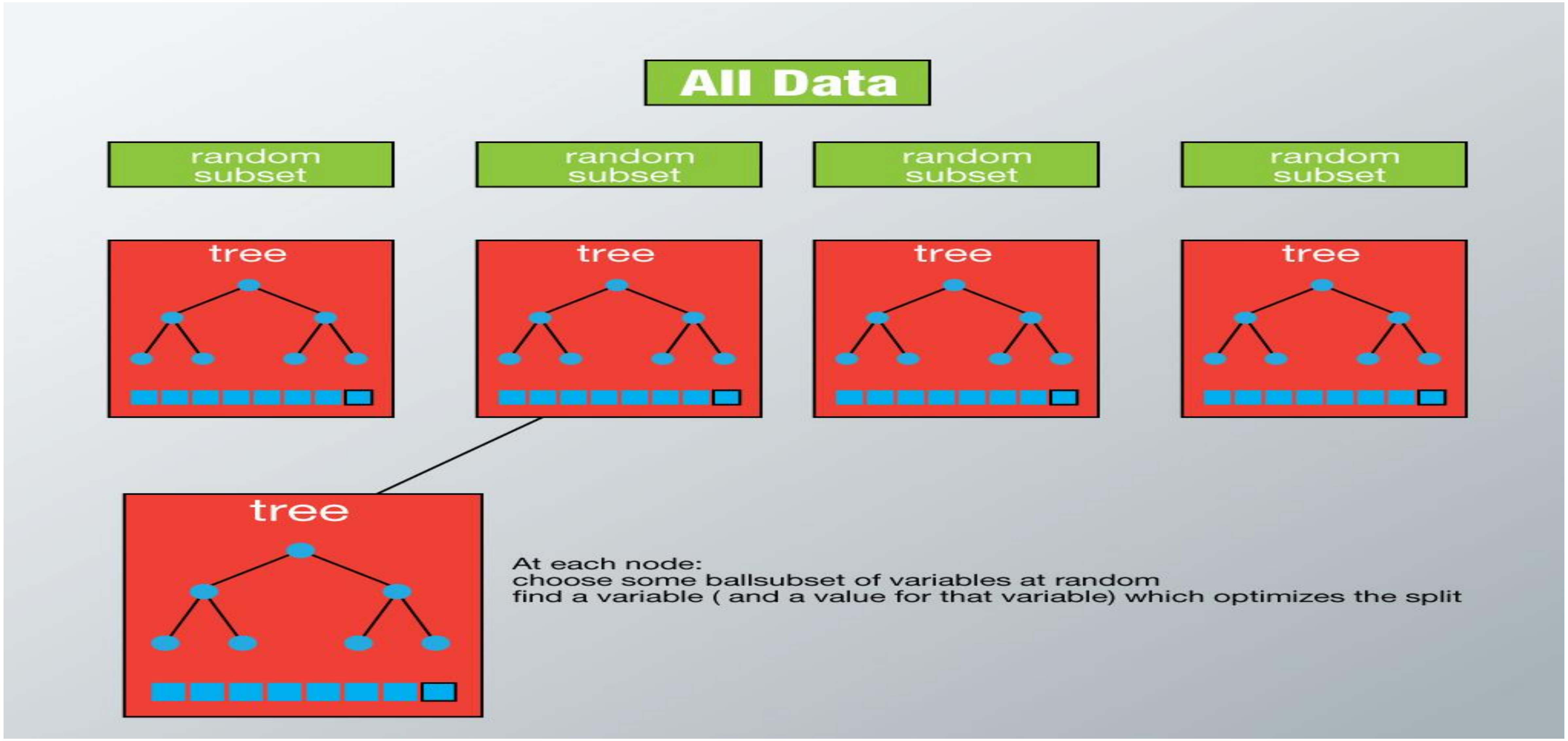
1. **Create Multiple Dataset's**
2. **Build Multiple Classifiers**
3. **Combine Classifiers**



Random Forest

- Random forest is like bootstrapping algorithm with Decision tree (CART) model. Say, we have 1000 observation in the complete population with 10 variables. Random forest tries to build multiple CART model with different sample and different initial variables. For instance, it will take a random sample of 100 observation and 5 randomly chosen initial variables to build a CART model. It will repeat the process (say) 10 times and then make a final prediction on each observation. Final prediction is a function of each prediction. This final prediction can simply be the mean of each prediction.

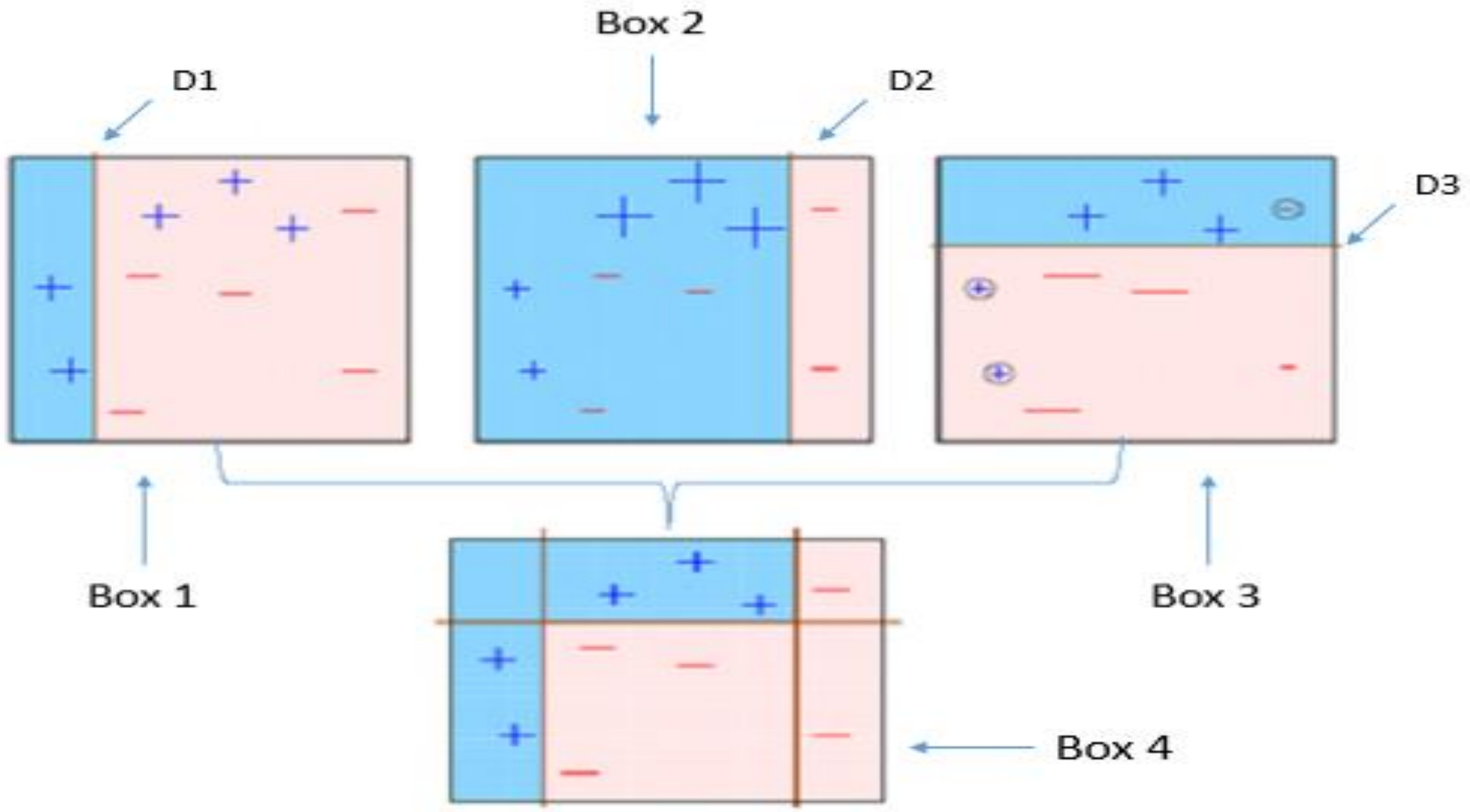
How Random Forest works



At each node:
choose some ballsubset of variables at random
find a variable (and a value for that variable) which optimizes the split

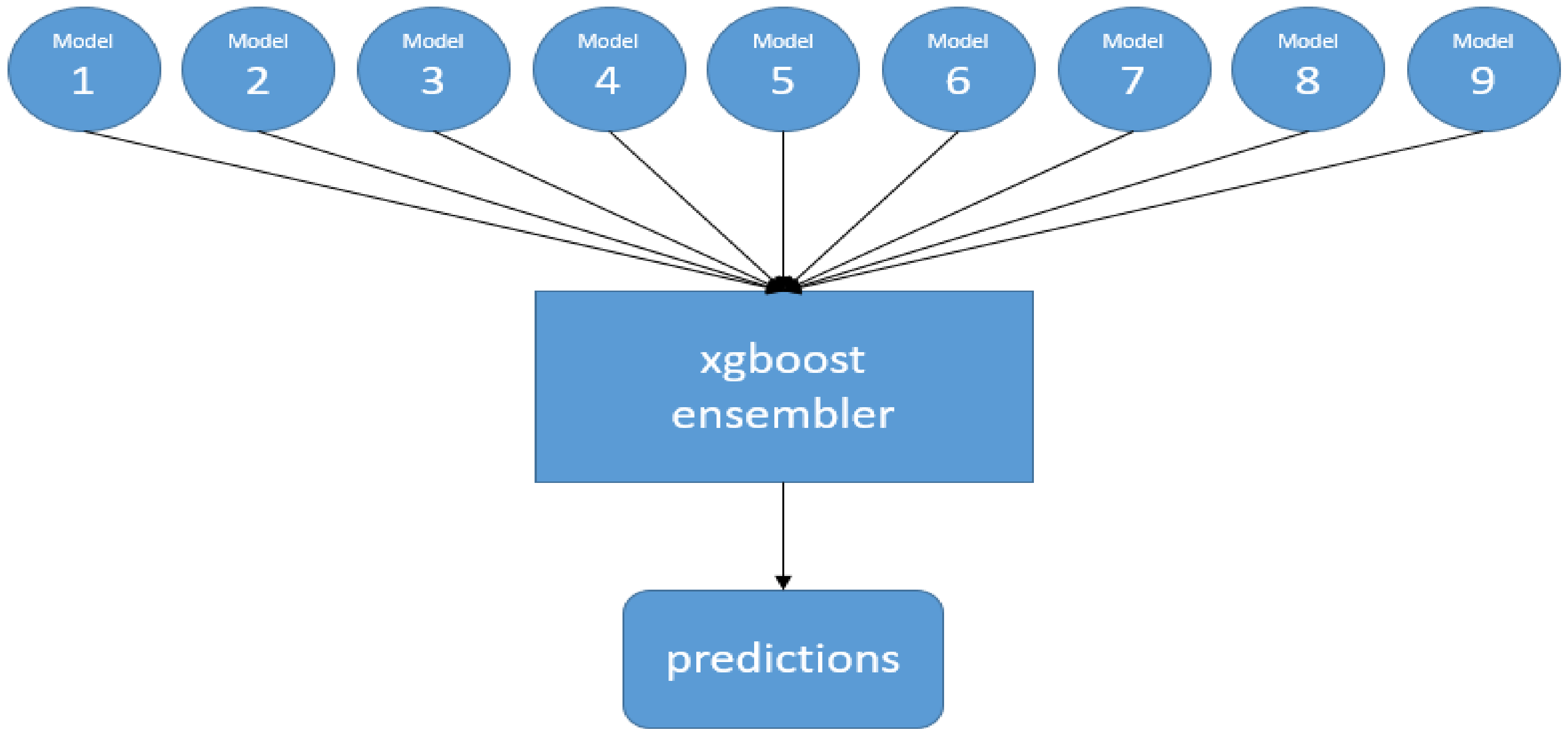
Boosting

__Definition:__ The term ‘Boosting’ refers to a family of algorithms which converts weak learner to strong learners.



Gentle Introduction to Xgboost

- XGBoost is an algorithm that has recently been very dominating applied machine learning and Kaggle competitions for structured or tabular data
- XGBoost is an implementation of gradient boosted decision trees designed for speed and performance

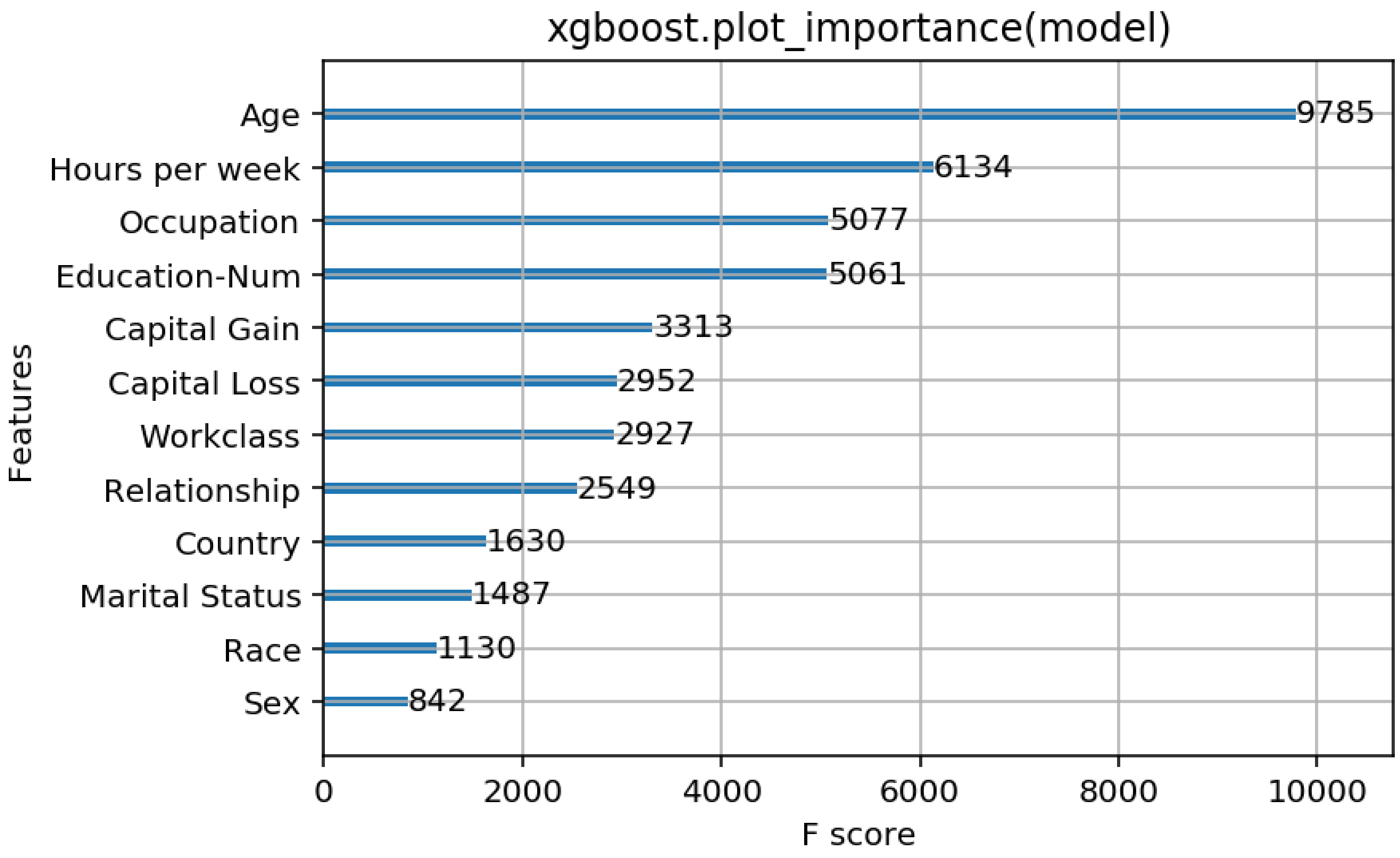


How Do We Enable Interpretation?

A black-box model is built and we need a way to interpret it

- Model agnostic feature importance
- Local interpretable model agnostic explanation (LIME)
- Xgboost Explainer Package
- Eli5
- The What-If Tool: Code-Free Probing of Machine Learning Models by Google
- Skater ,python package for interpreting(via post-hoc evaluation/rule extraction) predictive model

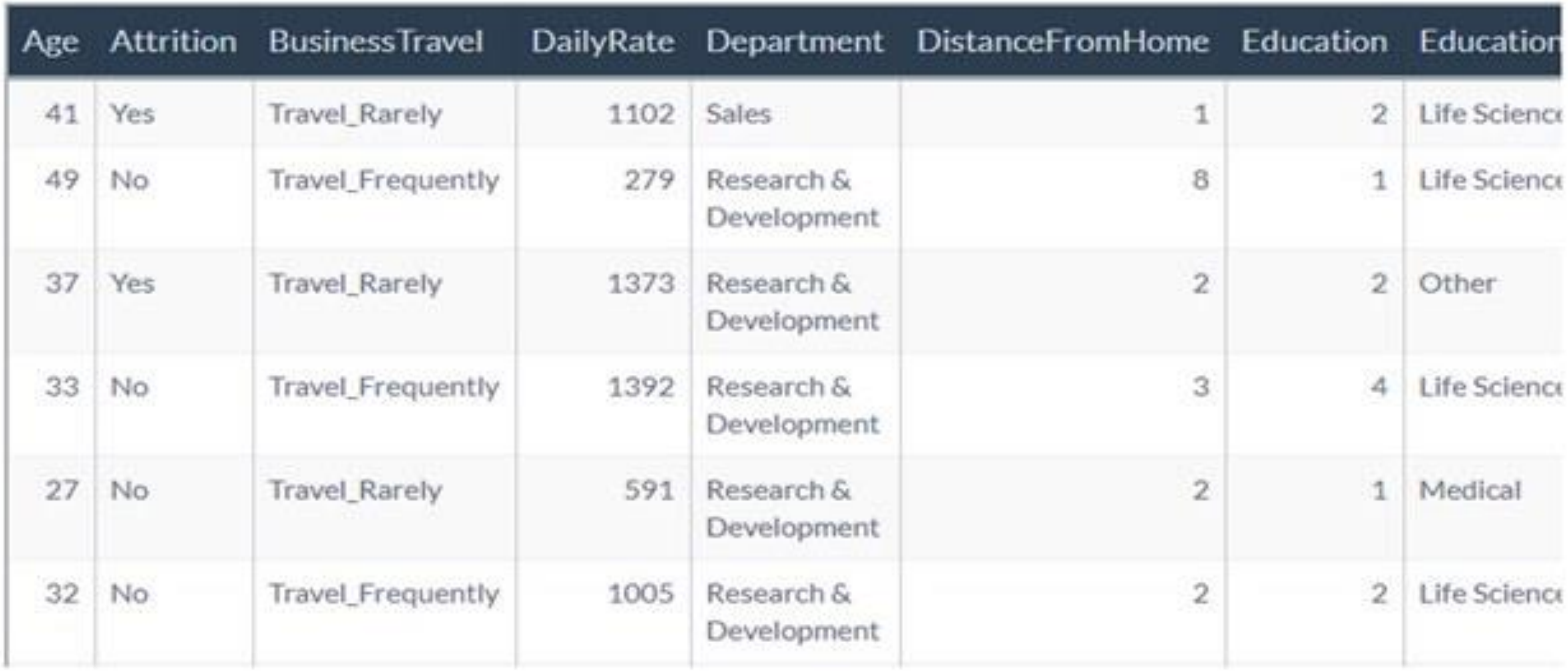
Model agnostic Feature Importance



- Feature Importance for a model trained to predict if people will report over $50k of income from the classic “adult” census dataset
- Simple Binary Classification Problem

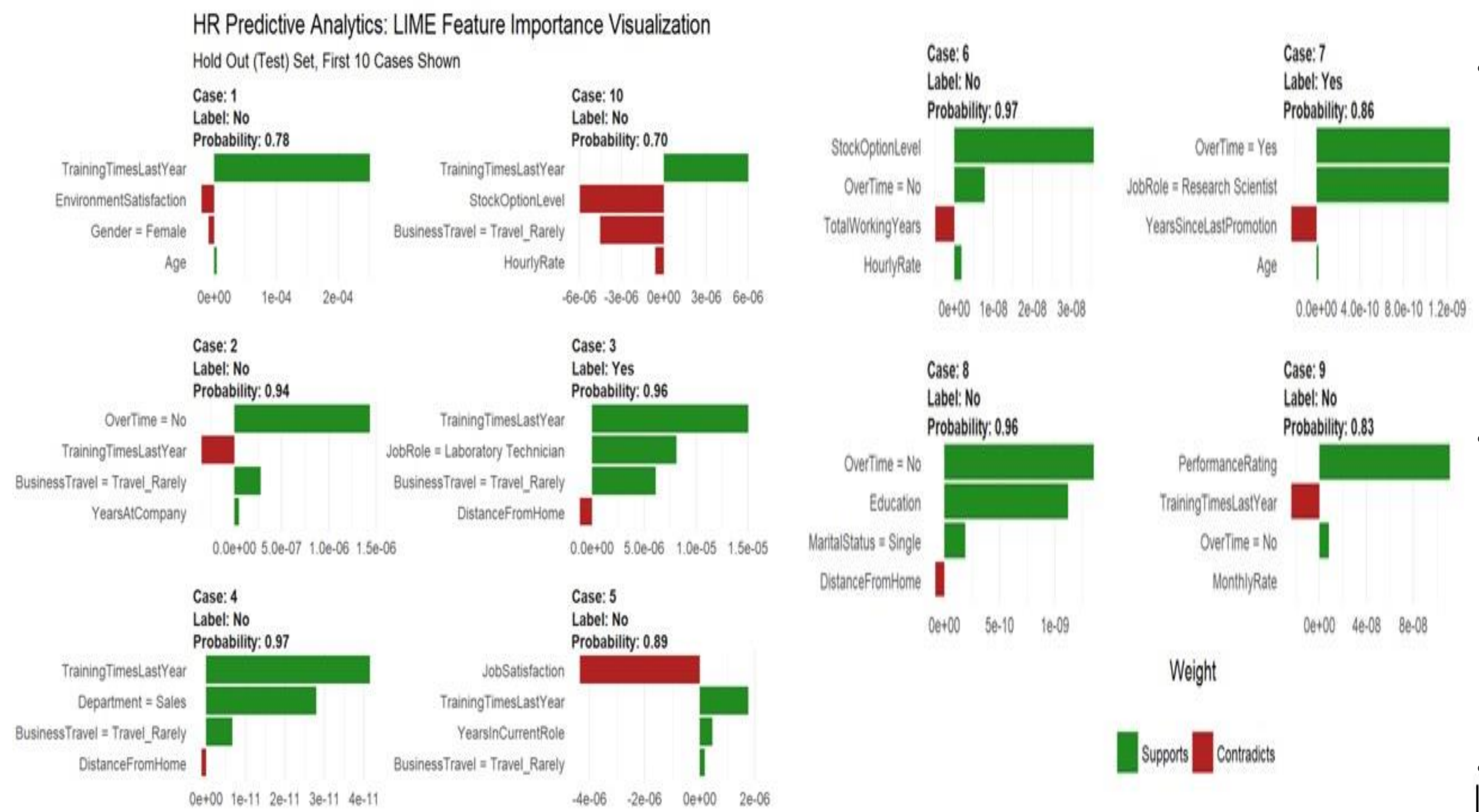
Dataset Explanation - The Problem: Employee Attrition

Let’s Understand the Dataset first which we will be using to explain different concepts for Interpreting Machine Learning Models. The dataset includes 1470 employees (rows) and 35 features (columns) a portion of which have left the organization (Attrition = “Yes”). We have developed Random Forest Model for it.



Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	Educatior
41	Yes	Travel_Rarely	1102	Sales	1	2	Life Scienc
49	No	Travel_Frequently	279	Research & Development	8	1	Life Scienc
37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other
33	No	Travel_Frequently	1392	Research & Development	3	4	Life Scienc
27	No	Travel_Rarely	591	Research & Development	2	1	Medical
32	No	Travel_Frequently	1005	Research & Development	2	2	Life Scienc

LIME – Let’s Understand

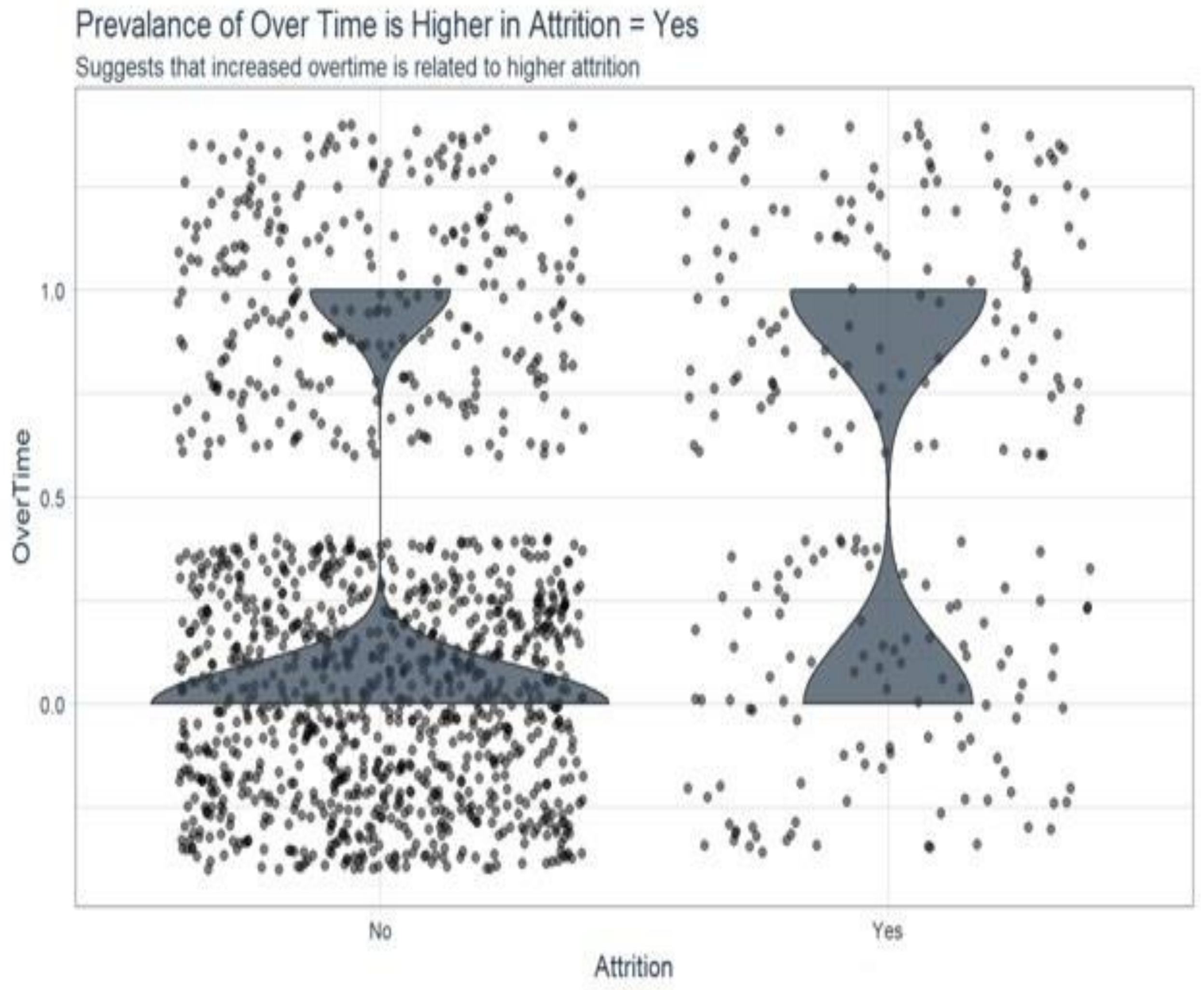


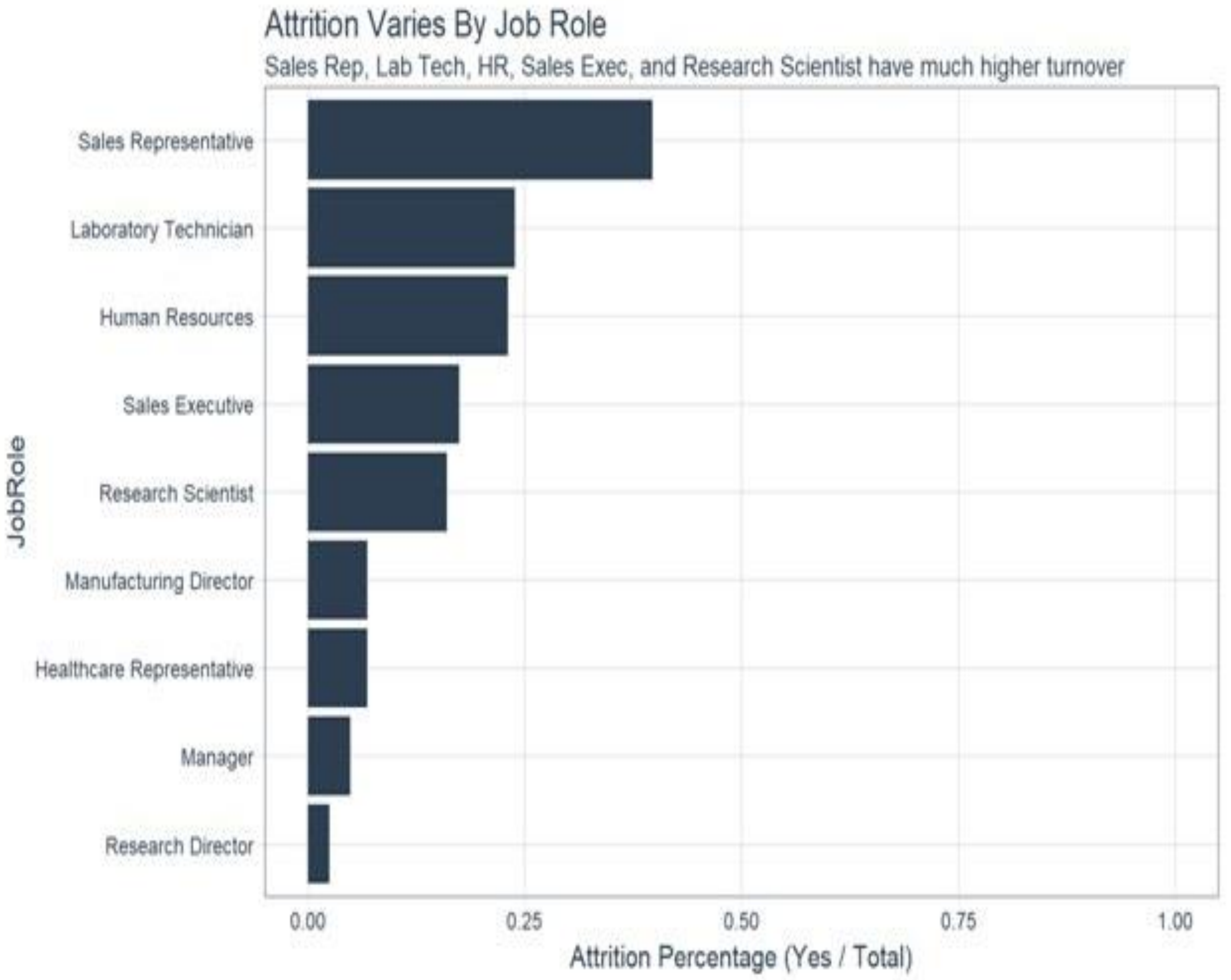
- For features like Over Time and Job Role, the variance appears to be linked. We can see that the group with Attrition = “No” has a lower proportion of employees working overtime

- In addition roles such as Sales Representative, Laboratory Technician and Human Resources had a higher attrition percentage as compare to other Job Roles

- On Random Forest Model

Variance of Overtime & Job Role on Attrition





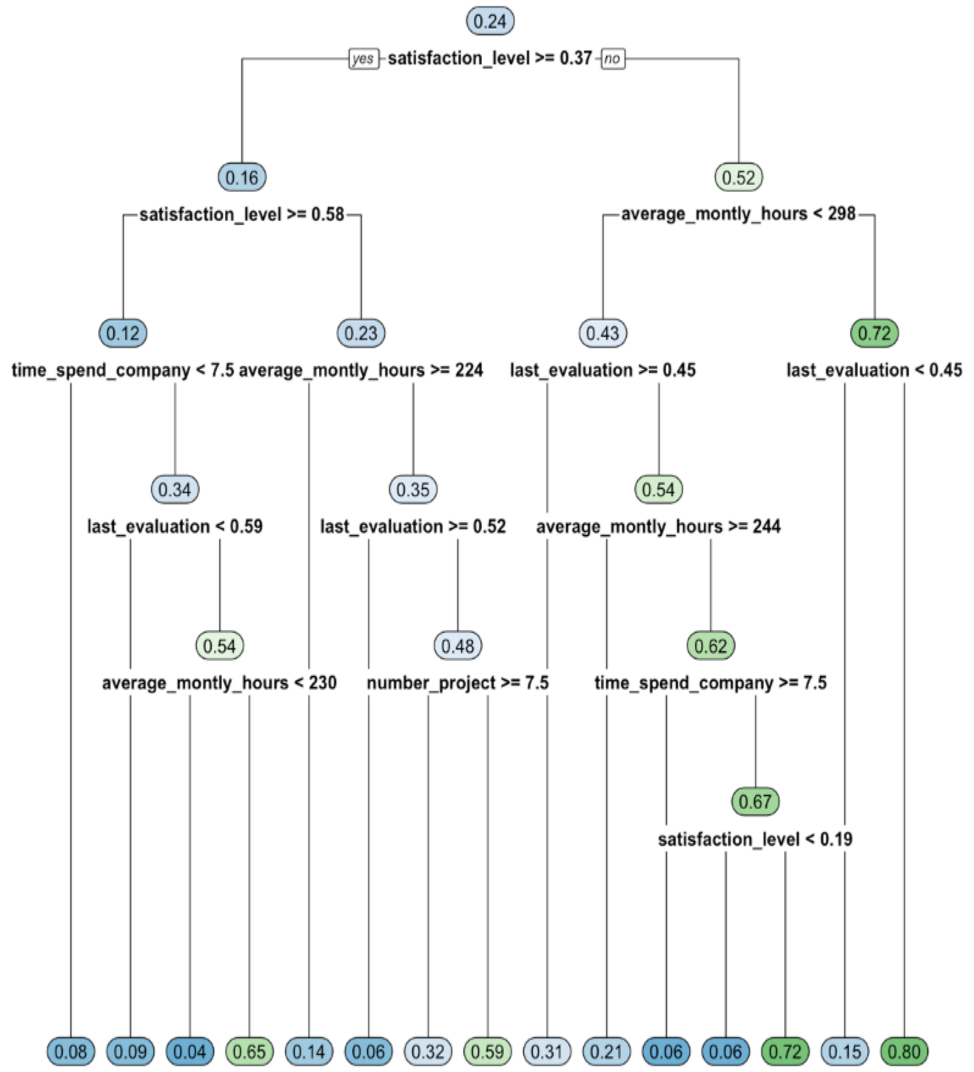
XGBOOST EXPLAINER PACKAGE IN R

Before I explain this , Let’s talk about one of simple Binary Classification Model using Decision Tree & Xgboost Algorithm …

For the demonstration, I'll use a dataset from Kaggle, to predict employee attrition from a fictional company.

Decision Tree (AUC 0.812)

High Interpretational Power , Comparatively less Accuracy



- We can say *exactly* how each feature has influenced the prediction.
- Let's say we have an employee with the following attributes:

  Satisfaction Level = 0.23

  Average Monthly Hours = 200

  Last Evaluation = 0.5

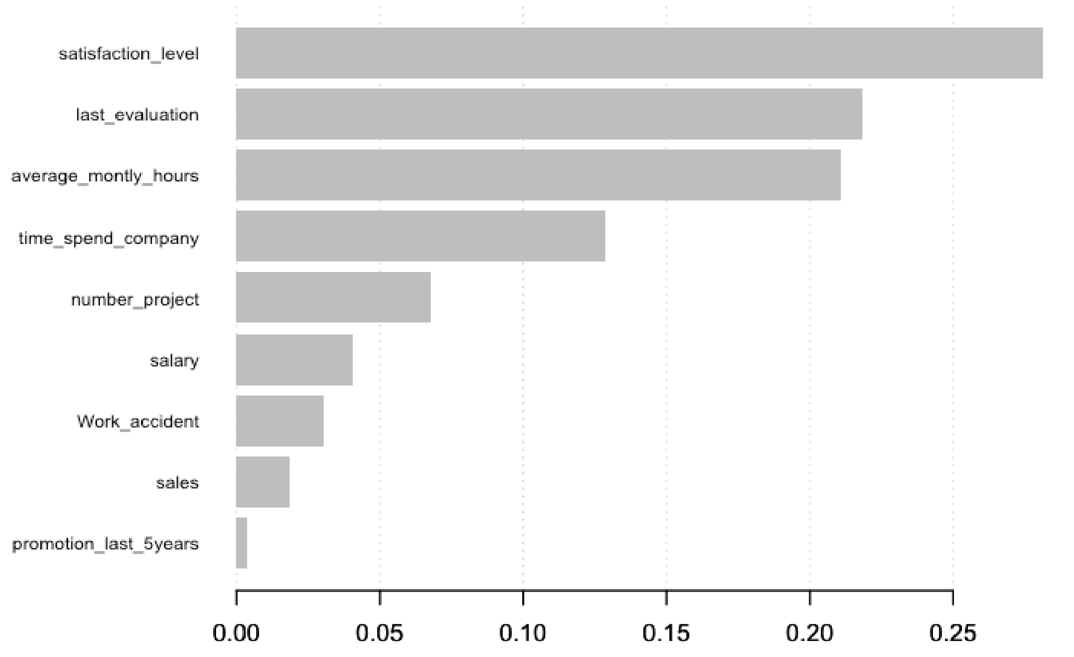
The model would estimate the likelihood of this employee leaving at **0.31** (i.e. 31%)

XGBoost (AUC: 0.923)

- Much better! An xgboost ensemble with 53 trees drastically outperforms the single decision tree

- **We have Observed better accuracy but Can we explain this Prediction to HR Head who wanted to know at Individual level what is driving the Attrition ?**

- **In Most case we show the Feature Importance Graph**



So what ? Is this possible , Yes through **Xgboost explainer package**

This tells us that satisfaction level is the most important variable across all predictions, but there's no guarantee it's the most important for this particular employee

Details About - Xgboost Explainer Package (R)

- XgboostExplainer makes your XGBoost model as transparent and 'white-box' as a single decision tree

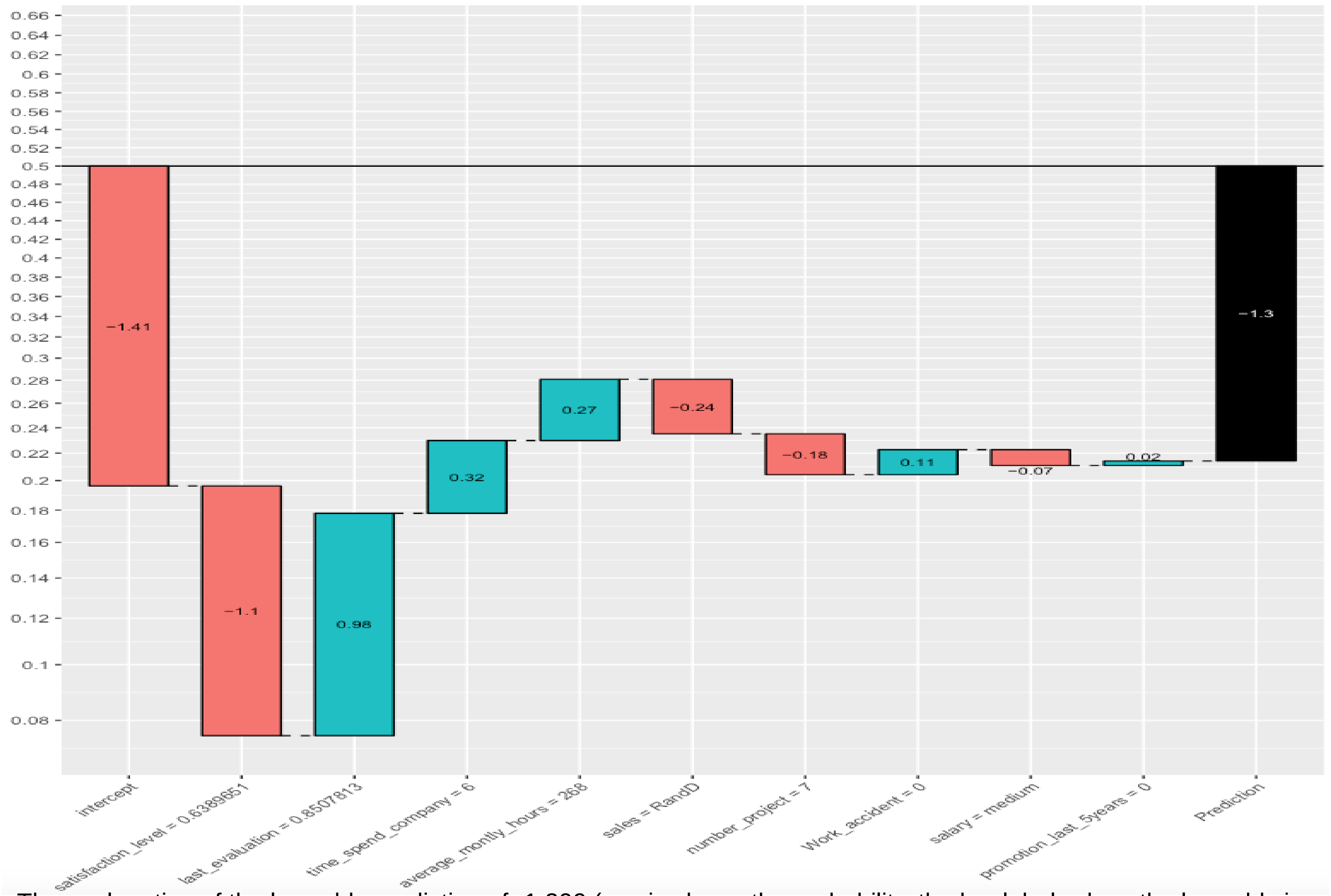
We will Cover 3 Important Things –

1. How Single Decision Tree is not great at predicting but is very interpretable

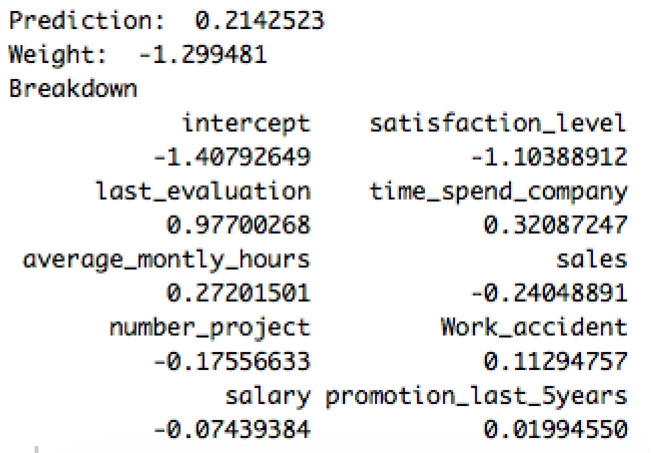
2. How xgboost (Ensemble of Decision Tree) is very good at Predicting , but not very interpretable

3. Show how to make Xgboost model as interpretable as a single decision tree, by using the new **xgboostExplainer** package

Detailed Explanation of Xgboost Explainer



Take the employee with the 21.4% likelihood of leaving - through the XGBoost Explainer, this is what you get back:



```
Prediction:  0.2142523
Weight:  -1.299481
Breakdown
            intercept    satisfaction_level
          -1.40792649           -1.10388912
      last_evaluation    time_spend_company
           0.97700268            0.32087247
 average_montly_hours                 sales
           0.27201501           -0.24048891
       number_project         Work_accident
          -0.17556633            0.11294757
               salary promotion_last_5years
          -0.07439384            0.01994550
```

The explanation of the log-odds prediction of -1.299 (y-axis shows the probability, the bar labels show the log-odds impact of each variable)

- The log-odds of the prediction, which in this case is -1.299

$$\frac{1}{1 + \exp(-(-1.299))} = 0.214333\ldots$$

- Walking through step by step, just like we did for the decision tree, except this time working with log-odds:

-1.41 ::: Baseline (Intercept)

-1.10 ::: Satisfaction Level (prediction is now -2.51)

+0.98 ::: Last Evaluation (prediction is now -1.53)

+0.32 ::: Time Spent At Company (prediction is now - 1.21)

+0.27 ::: Hours Average Monthly (prediction is now -0.94)

-0.24 ::: Sales (prediction is now -1.18)

-0.18 ::: Number of Projects (prediction is now -1.36)

+0.11 ::: Work Accident (prediction is now -1.25)

-0.07 ::: Salary (prediction is now -1.32)

+0.02 ::: Promotion Last 5 Years (prediction is now -1.30)

= -1.299 ::: Prediction

High satisfaction score of the employee does indeed pull the predicted log-odds **down** (by-1.1)**,** but this is more than cancelled out by the impacts of **the last evaluation, time at company and average hours variables**, all of which pull the log-odds **up**.

How does it work ?

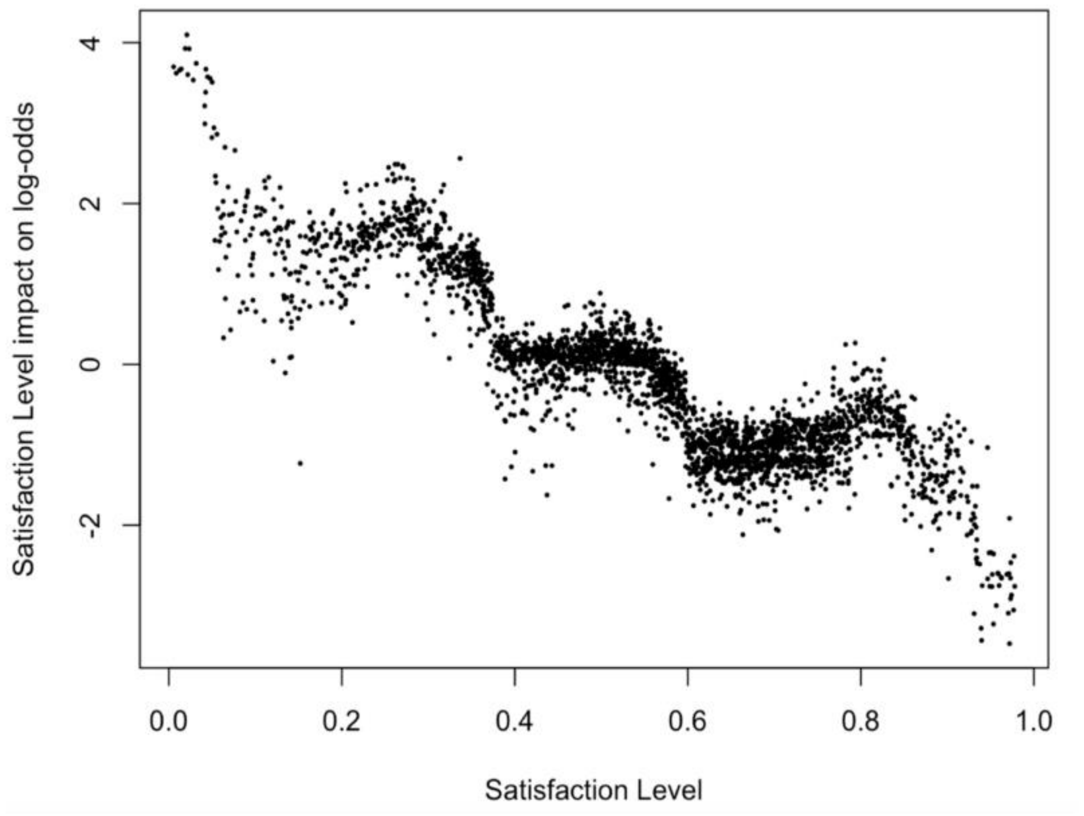
- key point to understand is how the log-odds contribution of each feature is calculated

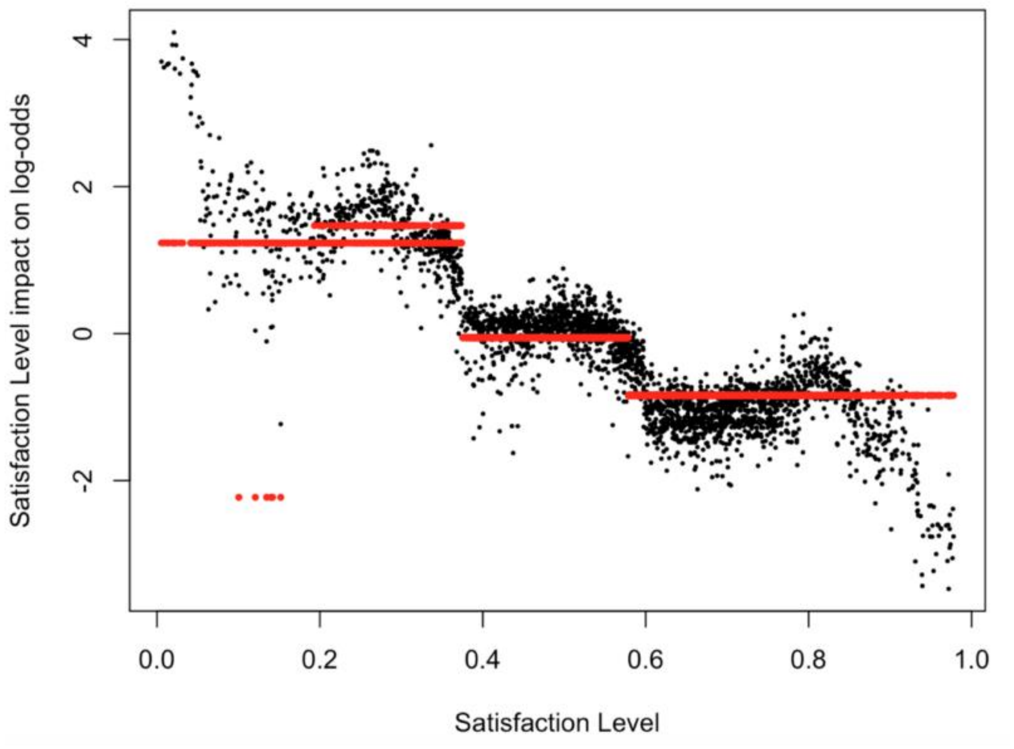
- It adding up the contributions of each feature for every tree in the ensemble, in exactly the same way as it happened in the decision tree

- The R xgboost package contains a function **'xgb.model.dt.tree**' that exposes the calculations that the algorithm is using to generate predictions

- It is important to understand that the 'impacts' here are not static coefficients as in a logistic regression. The impact of a feature is dependent on the specific path that the observation took through the ensemble of trees

1. If this were a logistic regression, the points would be on a straight line, with the slope dependent on the value of the coefficient
2. If this were a decision tree, the points would lie on a set of horizontal lines
3. The xgboostExplainer beautifully captures the non-linearity, but is not restricted by the requirement of a single straight line or steps.





Black: XGBoost Explainer, Red: single decision tree

Each point is one employee from the test set. The satisfaction level of the employee is plotted on the x-axis; the impact of the satisfaction level on the log-odds of leaving is plotted on the y-axis.

THANK YOU

Questions and Answers

References

- https://medium.com/applied-data-science/new-r-package-the-xgboost-explainer-51dd7d1aa211
- https://www.analyticsvidhya.com/blog/2017/06/building-trust-in-machine-learning-models/
- https://github.com/marcotcr/lime
- https://www.analyticsvidhya.com/blog