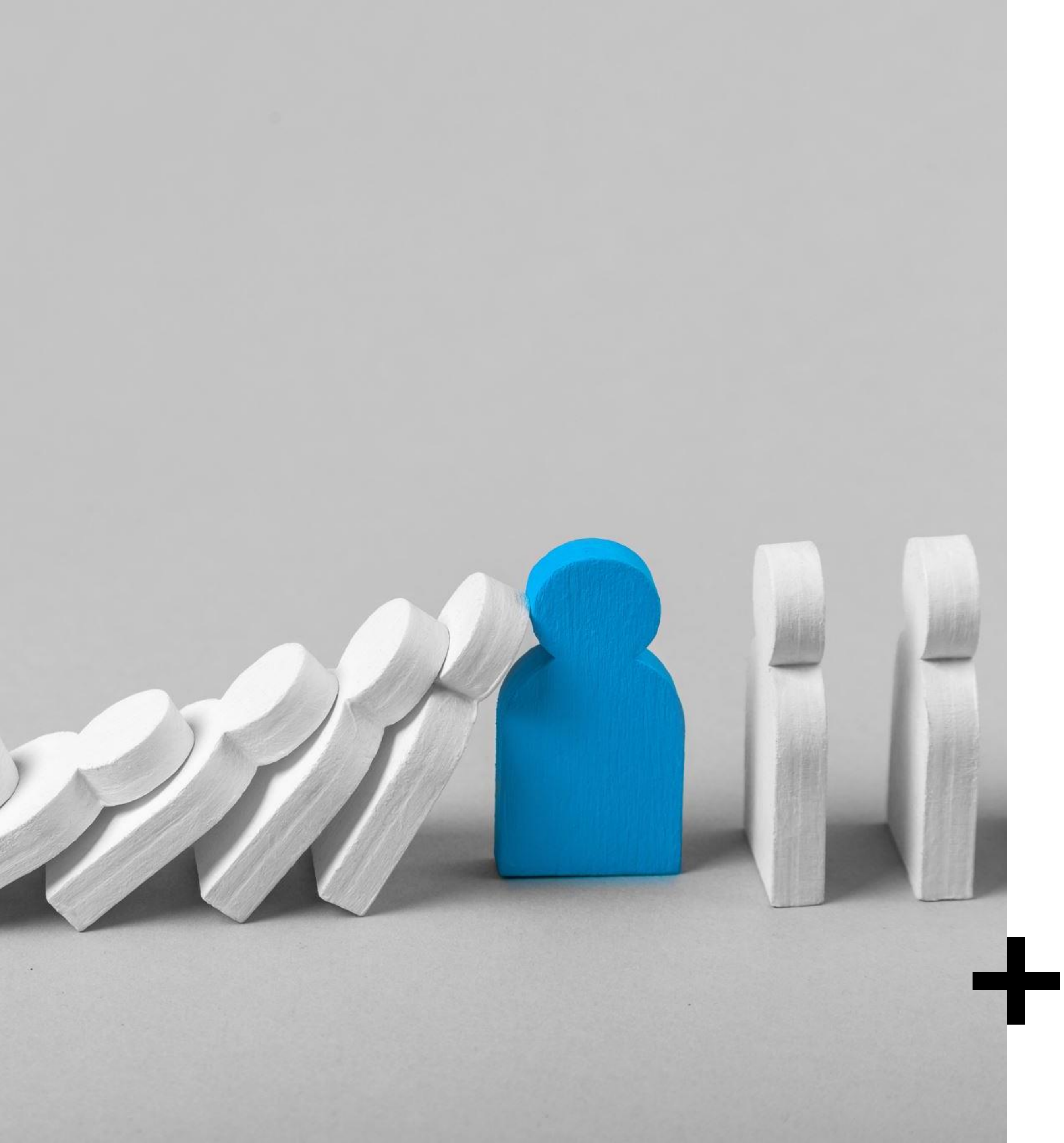


Machine Learning based Employee attrition prediction

Aarish Shahab

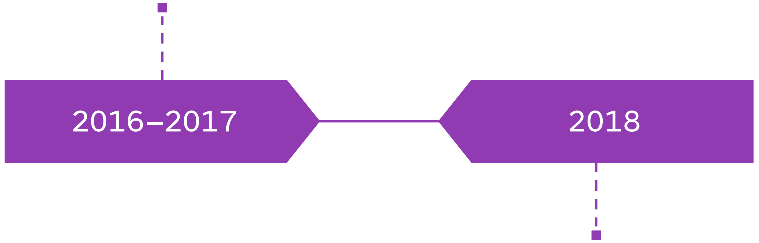
Machine Learning Engineer

Topics

- Problem Statement
- Dataset Inspection
- Approach for solution
- Feature Engineering
- Exploratory Data Analysis
- Model Selection
- Feature Importance
- Hyperparameter Tuning

Problem Statement

Given Employees two (2016-2017) year sales and performance data:



Build a mathematical model to predicts how many employees are going to resign in the first two Quarters of 2018.

Dataset Inspection

Train dataset:

- Dataset consist of 13 columns (features) and 19104 rows (Data point).
- Data point were collected first of every month which includes individual Employee Sale, Performance and other details.
- Features were *'Date of Data Point', 'Employee ID', 'Age', 'Gender', 'Education Level', 'Salary', 'Date of Joining', 'Last working Date', 'Joining Designation', 'Current Designation', 'Total Sales for past month', and 'Quarterly rating'* .

Test dataset:

- Dataset consists of 1 feature and 741 data points.
- Data points are Employee IDs of Employee who have not Resigned.

Approach for Solution

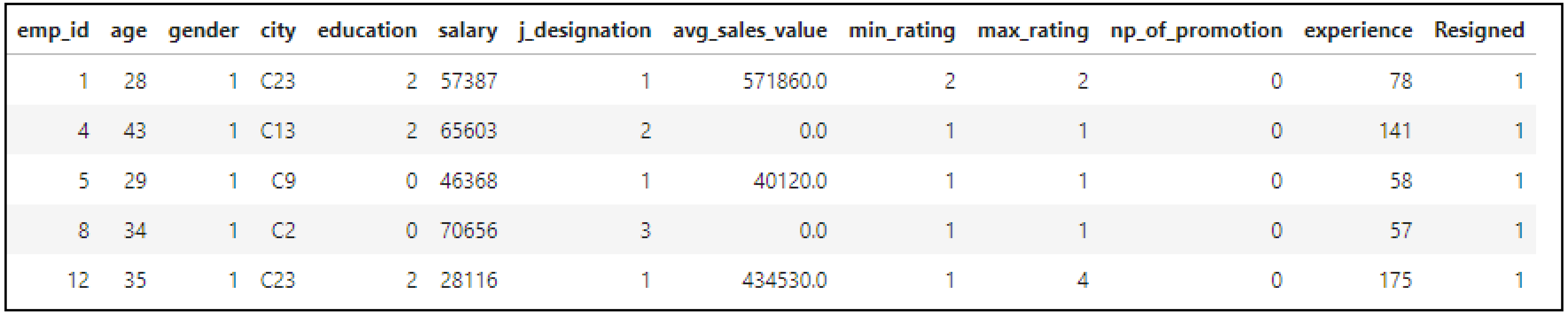
Using Supervised Learning, train a Machine Learning model using half-yearly (6 month) sales and performance data.

Calculate the probability of an Employee resigning in the next 6 month

Feature Engineering

- Divided given 2 years of data points in 4 half-yearly dataset as “2016_HY1, 2016_HY2, 2017_HY1, 2017_HY2” respectively.

- Created new features like total experience, number of half-yearly promotions, minimum and maximum Quarterly rating and a dichotomous feature Resigned.
- Updated existing features like Average half-yearly sales value.
- Encoded existing feature like Education, Gender.



emp_id	age	gender	city	education	salary	j_designation	avg_sales_value	min_rating	max_rating	np_of_promotion	experience	Resigned
1	28	1	C23	2	57387	1	571860.0	2	2	0	78	1
4	43	1	C13	2	65603	2	0.0	1	1	0	141	1
5	29	1	C9	0	46368	1	40120.0	1	1	0	58	1
8	34	1	C2	0	70656	3	0.0	1	1	0	57	1
12	35	1	C23	2	28116	1	434530.0	1	4	0	175	1

Exploratory Data Analysis

There are 24 Employee’s who are neither with the company nor officially resigned.

Employee promotion is highly correlated with Employee total experience.

Employee promotion is highly correlated with Average Sales.

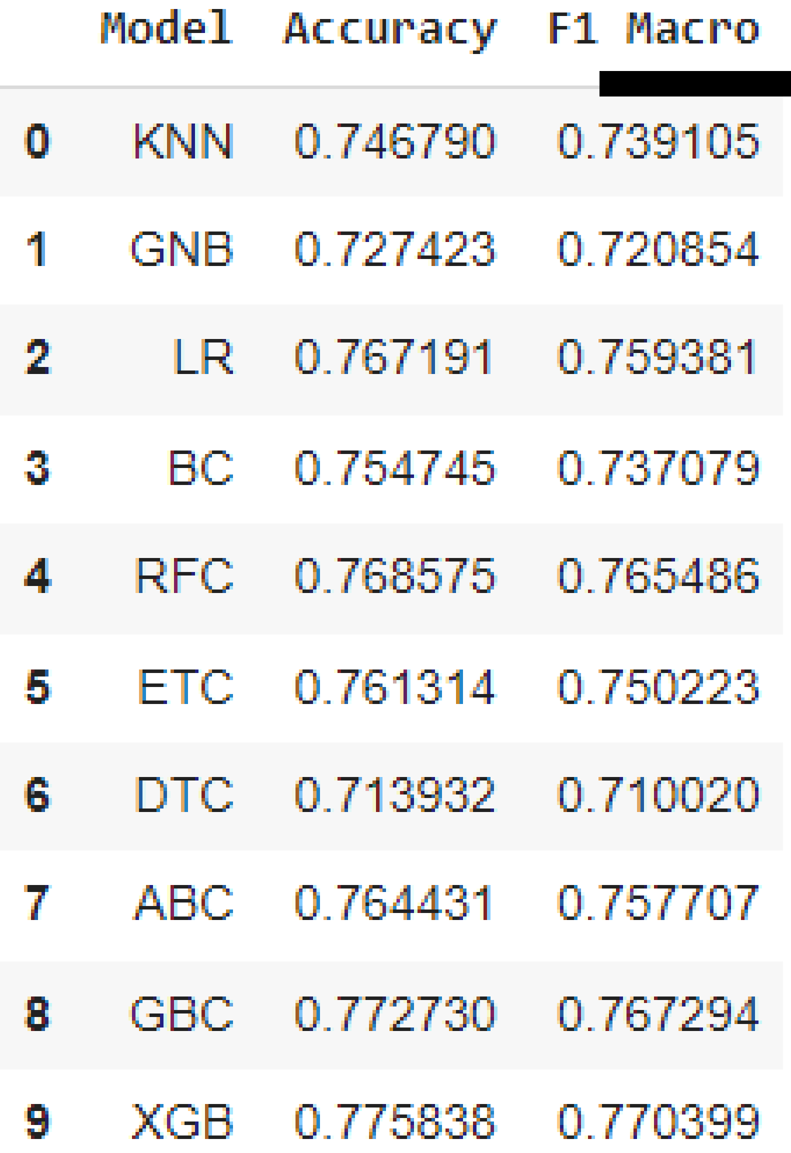
Employee Quarterly rating is highly correlated with Average Sales.

Employee Resignation is moderately correlated with quarterly rating, experience, and promotions.

Model Selection

– Tried multiple machine learning models like:
    - XGBoost Classifier
    - Gradient Boosting Classifier
    - AdaBoost Classifier
    - Random Forest Classifier

Where highest accuracy of 77.58% was achieved with XGBoost Classifier.

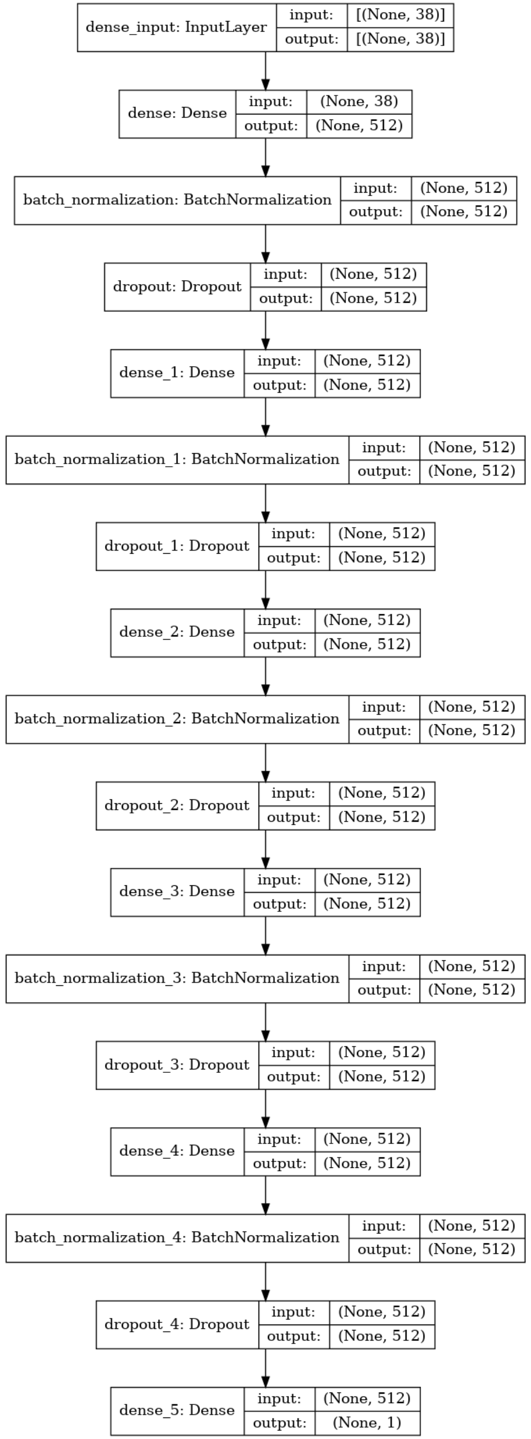


	Model	Accuracy	F1 Macro
0	KNN	0.746790	0.739105
1	GNB	0.727423	0.720854
2	LR	0.767191	0.759381
3	BC	0.754745	0.737079
4	RFC	0.768575	0.765486
5	ETC	0.761314	0.750223
6	DTC	0.713932	0.710020
7	ABC	0.764431	0.757707
8	GBC	0.772730	0.767294
9	XGB	0.775838	0.770399

Model Selection

– Tried Deep Learning models:
    - 5 Layer architecture.
    - RELU activation for hidden layers and SIGMOID activation for output layer.
    - ‘Adam’ optimizer.
    - ‘binary_crossentropy’ as loss function.
    - batch_size 32 and 19 epoch.

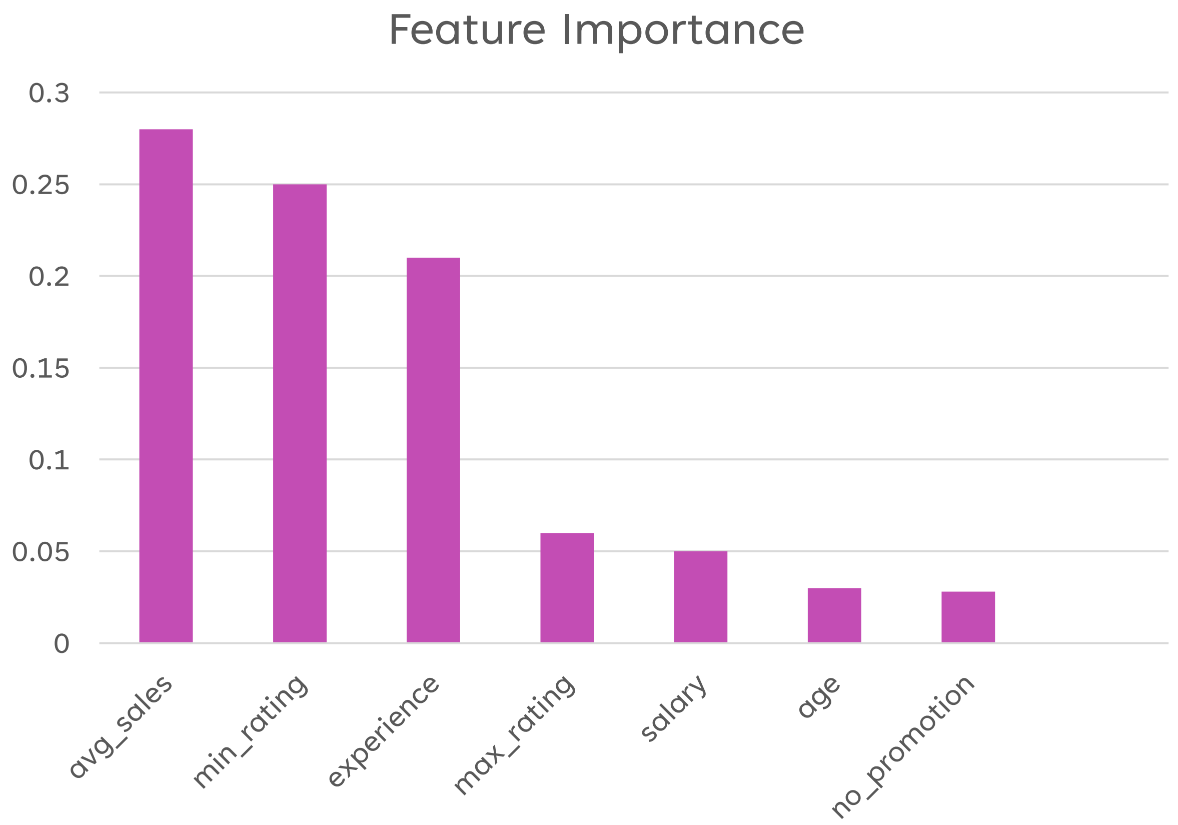
Obtained 70.71 F1-macro accuracy on Public leader board of Hack-a-thon.



dense_input: InputLayer | input: [(None, 38)] | output: [(None, 38)]
dense: Dense | input: (None, 38) | output: (None, 512)
batch_normalization: BatchNormalization | input: (None, 512) | output: (None, 512)
dropout: Dropout | input: (None, 512) | output: (None, 512)
dense_1: Dense | input: (None, 512) | output: (None, 512)
batch_normalization_1: BatchNormalization | input: (None, 512) | output: (None, 512)
dropout_1: Dropout | input: (None, 512) | output: (None, 512)
dense_2: Dense | input: (None, 512) | output: (None, 512)
batch_normalization_2: BatchNormalization | input: (None, 512) | output: (None, 512)
dropout_2: Dropout | input: (None, 512) | output: (None, 512)
dense_3: Dense | input: (None, 512) | output: (None, 512)
batch_normalization_3: BatchNormalization | input: (None, 512) | output: (None, 512)
dropout_3: Dropout | input: (None, 512) | output: (None, 512)
dense_4: Dense | input: (None, 512) | output: (None, 512)
batch_normalization_4: BatchNormalization | input: (None, 512) | output: (None, 512)
dropout_4: Dropout | input: (None, 512) | output: (None, 512)
dense_5: Dense | input: (None, 512) | output: (None, 1)

Feature Importance

- Feature Importance for Gradient Boosting Classifier



Hyperparameter Tuning

- Selected Deep Learning model as final mode because I was best F1-macro accuracy on Hack-a-thon Public leader board. Later to tune the model to get best Hyperparameter, Experimented with:
  - Different layered architecture and obtained best result with 5 Layer architecture.
  - Different number of neurons at different layers and obtained 512 units of neurons yielding best results.
  - batch_size 16, 32, 64 and obtained best result with batch_size 32.
  - Optimizers RMSprop, SGD and ADAM and obtained best result with ADAM optimizer.

Thank You