Imarticus Learning Private Limited

A Capstone Project on

Employee Attrition Prediction

By

Prasad Vasant Verulkar (PGA11, Pune Batch)

Problem Statement

To Uncover the factors that cause employees to leave the organization

Build classification models to predict which employee is likely to churn and help the business to devise policies and attract back the right talent.

Dataset Description

Rows: 1470
Columns: 35

Attrition (Y Variable) Distribution:
Yes: 237
No: 1233

- Columns
  - Numerical: 5
  - Categorical: 29
    - Nominal: 9
    - Ordinal: 20

#	Column	Non-Null Count	Dtype
0	Age	1470 non-null	int64
1	Attrition	1470 non-null	object
2	BusinessTravel	1470 non-null	object
3	DailyRate	1470 non-null	int64
4	Department	1470 non-null	object
5	DistanceFromHome	1470 non-null	int64
6	Education	1470 non-null	int64
7	EducationField	1470 non-null	object
8	EmployeeCount	1470 non-null	int64
9	EmployeeNumber	1470 non-null	int64
10	EnvironmentSatisfaction	1470 non-null	int64
11	Gender	1470 non-null	object
12	HourlyRate	1470 non-null	int64
13	JobInvolvement	1470 non-null	int64
14	JobLevel	1470 non-null	int64
15	JobRole	1470 non-null	object
16	JobSatisfaction	1470 non-null	int64
17	MaritalStatus	1470 non-null	object
18	MonthlyIncome	1470 non-null	int64
19	MonthlyRate	1470 non-null	int64
20	NumCompaniesWorked	1470 non-null	int64
21	Over18	1470 non-null	object
22	OverTime	1470 non-null	object
23	PercentSalaryHike	1470 non-null	int64
24	PerformanceRating	1470 non-null	int64
25	RelationshipSatisfaction	1470 non-null	int64
26	StandardHours	1470 non-null	int64
27	StockOptionLevel	1470 non-null	int64
28	TotalWorkingYears	1470 non-null	int64
29	TrainingTimesLastYear	1470 non-null	int64
30	WorkLifeBalance	1470 non-null	int64
31	YearsAtCompany	1470 non-null	int64
32	YearsInCurrentRole	1470 non-null	int64
33	YearsSinceLastPromotion	1470 non-null	int64
34	YearsWithCurrManager	1470 non-null	int64

Steps in Building ML Model

1. **Reading Data**
2. **Data Preprocessing**
   - Handling Missing values
   - Handling Outliers
3. **Exploratory Data Analysis**
   Analysing data to reveal hidden insights.
4. **Feature Engineering**
   - Categorical Encoding
   - Binning
5. **Setting up Validation Strategy**
   - Selection of a validation set based on the distribution of train and test set
6. **Model Building**
7. **Feature Selection**
   Selecting dimensionality of dataset.
8. **Hyperparameter Optimization**
   To reduce over fitting
9. **Cross Validation**
   - To make model robust and reduce over fitting

Check for Missing Values

- No missing Values in Dataset

```
at.isnull().sum().sum()
```

```
0
```

Class Imbalance

```
# Highly imbalanced
at.Attrition.value_counts()
```

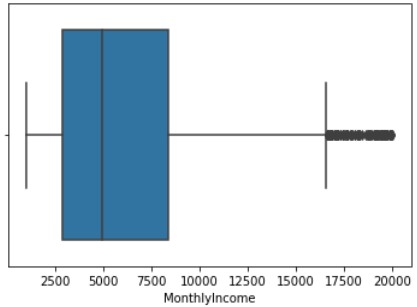
```
No      1233
Yes      237
Name: Attrition, dtype: int64
```

- Oversampling
- SMOTE

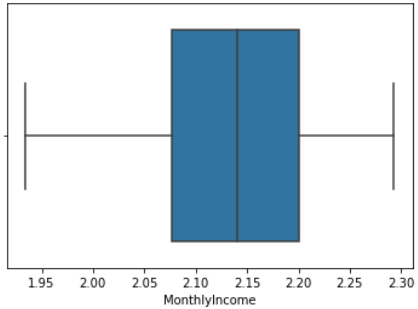
Handling Outliers

- Monthly Income column contains outliers.
- 7.8% of Monthly Income values are greater than upper whisker.
- Can’t remove them hence transform feature.

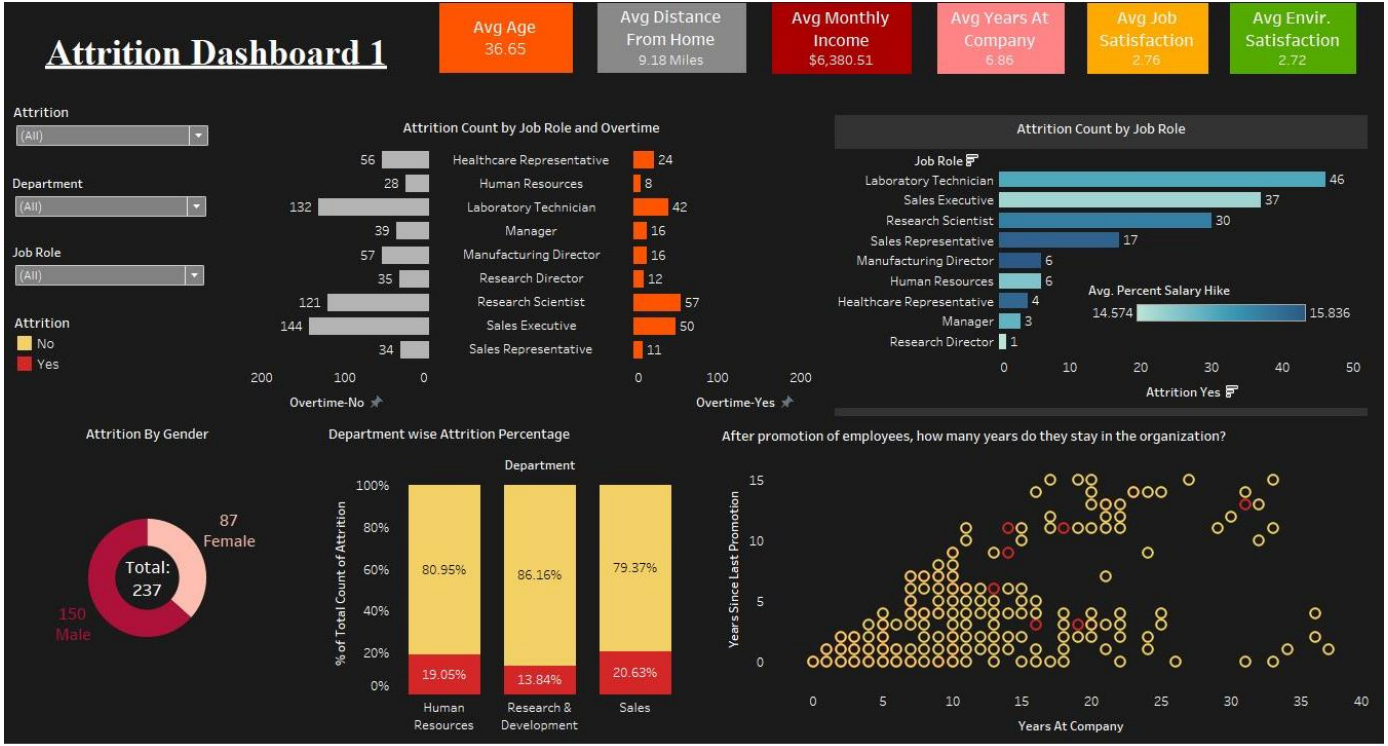
Before applying log



After applying log



Data Pre processing → **Exploratory Data Analysis** → Feature Engineering → Setting up Validation Strategy → Model Building & Comparison → Feature Selection



Dashboard Created using Tableau Public

Binning Ordinal Variables

- Binning ordinal variables with higher class levels.
- Bins are decided based on quartiles.
- To avoid over fitting and generalizing the model
- Might improve model performance.

Age_bins	Young, Adults, Middle_Aged, Old
DistanceFromHome	VeryClose, Medium, Far, VeryFar
Experiance_bins	Freshers, Associate, SnAssociate, Lead
Promotion_bins (Based on quantiles)	0_1, 2_3, 3_15

Feature Encoding

•One Hot Encoding

```
# One hot encoding

at_x = pd.get_dummies(at_x , columns = at_x.select_dtypes(include = 'object').columns)
at_x.head()
```

Total Features after One Hot Encoding: 44

Features Eliminated using above methods: 21

Some of the removed Features removed

- Education Field
- Performance Rating
- Education
- PercentSalaryHike
- Gender
- Business Travel

Zero Variance Predicators

- Features which contain only one class or only one unique number.

```
at.EmployeeCount.value_counts()
```

```
1    1470
Name: EmployeeCount, dtype: int64
```

```
at.StandardHours.value_counts()
```

```
80    1470
Name: StandardHours, dtype: int64
```

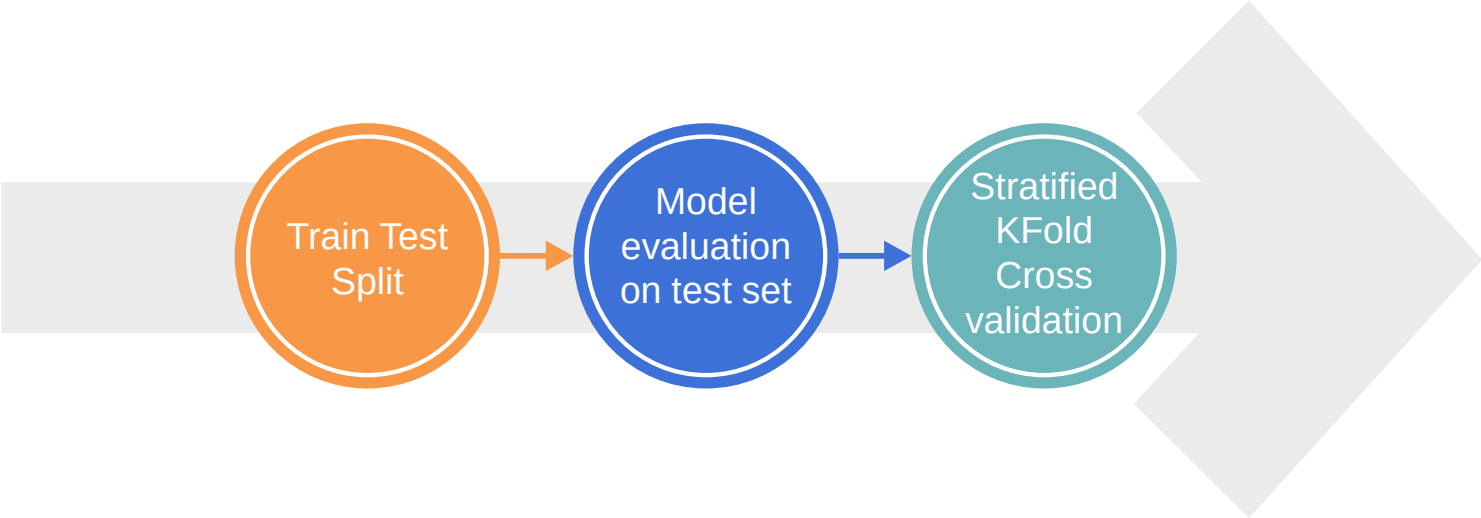
```
at.Over18.value_counts()
```

```
Y    1470
Name: Over18, dtype: int64
```

Unique Identifier

EmployeeNumber

Data Pre processing → Exploratory Data Analysis → Feature Engineering → Feature Selection → **Setting up Validation Strategy** → Model Building & Comparison

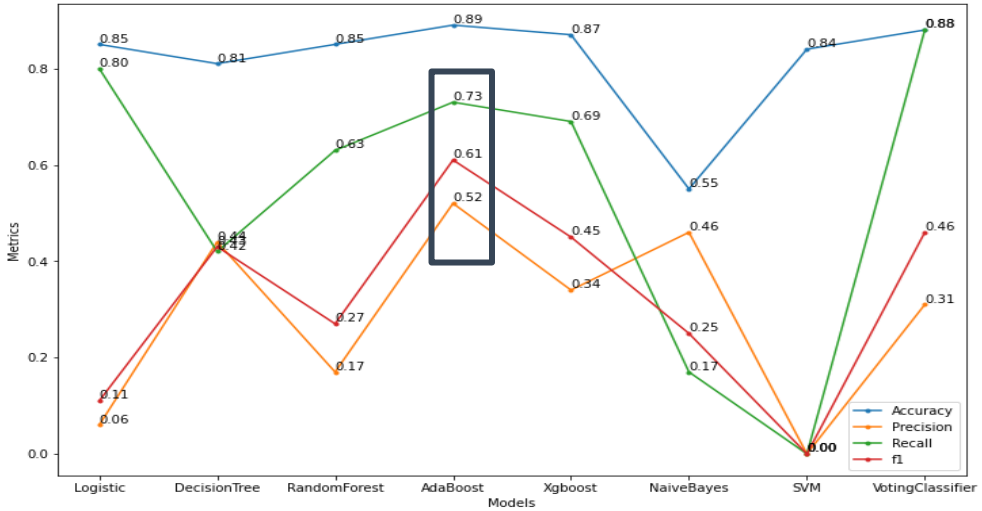


- Splitting data in 70:30 ratio keeping distribution of Y variable same.
- Building models and tuning hyper parameters on train set and evaluating on test.
- Stratified KFold cross validation on complete data for final model evaluation.

Data Pre processing → Exploratory Data Analysis → Feature Engineering → Feature Selection → Setting up Validation Strategy → **Model Building & Comparison**

Total Models Built: 8

Logistic Regression	Decision Tree	Random Forest	AdaBoost	XgBoost	NaiveBayes	SVM	VotingClassifier



Best Performing model is AdaBoost Classifier

```
*******For test data********
[[356  34]
 [ 14  37]]
              precision    recall  f1-score   support

           0       0.96      0.91      0.94       390
           1       0.52      0.73      0.61        51

    accuracy                           0.89       441
   macro avg       0.74      0.82      0.77       441
weighted avg       0.91      0.89      0.90       441

******For train Training*******
              precision    recall  f1-score   support

           0       0.98      0.91      0.94       924
           1       0.51      0.80      0.62       105

    accuracy                           0.90      1029
   macro avg       0.74      0.86      0.78      1029
weighted avg       0.93      0.90      0.91      1029
```

-Being the highest F1-score of 0.61, Precision 0.52 and recall 0.73 for class 1, AdaBoost Classifier model is selected among all other models.

- These scores are for the test set and thus final scores are evaluated using Cross Validation.

Hyper Parameter Tuning

Grid Search CV

```python
## HyperParameter Optimization

# Grid Search CV

grid = {'n_estimators':[50,100,150,200,300],'learning_rate':[0.1, 1, 1.1, 1.2, 1.3, 1.4]}

ada = AdaBoostClassifier()
cv = StratifiedKFold(n_splits=5,shuffle = True,random_state = 100)
scorer = make_scorer(f1_score)
from sklearn.model_selection import GridSearchCV

clf=GridSearchCV(estimator = ada, param_grid = grid, cv=cv, n_jobs=-1,scoring = scorer)
grid_result = clf.fit(at_train_x, at_train_y)
```

- Best Parameters: {'learning_rate': 1.4, 'n_estimators': 50}

Cross Validation

- Performed Stratified Kfold on complete data with K = 5.
- Mean Accuracy: 0.86
- Mean F1_score: 0.514
- Mean Test Set Accuracy: 0.90
- Mean Test F1score: 0.6677
- Top Features: MonthlyRate, DailyRate, TotalWorkingYears, YearsAtCompany, YearsInCurrentRole, YearsWithCurrManager