Analysis of Employee Retention rate

Internship in eSparsh Technologies Pvt Ltd(Bangalore)

Table of Contents

Abstract

Employee retention is a critical concern for organizations, and predicting retention rates can help proactively identify factors influencing attrition. In this project, we propose a machine learning-based approach to predict employee retention rates using historical employee data. Through data preprocessing, feature engineering, and training various machine learning models, we evaluate and optimize their performance. The selected model is deployed in a production environment for real-time predictions. This project aims to provide valuable insights into factors influencing retention, enabling organizations to implement targeted strategies for improving employee satisfaction and reducing attrition, ultimately leading to enhanced talent management and organizational performance.

Introduction

- Employee retention is a critical concern for organizations as high turnover rates can incur significant costs and disrupt business operations.

- To address this challenge, organizations are increasingly adopting data-driven approaches to gain insights into factors influencing employee retention.

- This project aims to develop a machine learning-based model for predicting employee retention rates using historical employee data.

- By analysing factors such as job satisfaction, age, promotion history, work-life balance, and employee demographics, organizations can understand the drivers of attrition and implement targeted strategies for improving retention.

- The project involves stages such as data collection from the HR database, data preprocessing to handle missing values and outliers, and feature engineering to extract meaningful insights.

- Machine learning algorithms including logistic regression, decision trees, random forests, and XGB Classifier are trained and evaluated to identify the best model for predicting retention rates.

enhance employee satisfaction, engagement, and overall talent management.

- By leveraging machine learning, organizations can gain a deeper understanding of retention factors, leading to improved organizational performance and a positive work environment.

The Dataset

EMPCODE	Name	BirthDat	Age	Gende	Departme	Designation	JoiningDa	LeavingDate	WORKPERIOD IN MONTHS	WORKPERIOD IN DAYS	Work Period	Retention	Reason	PermanentState
0259	Manikantan P C	07-Mar-82	41.345632	Male	Engineering	Associate Engineer	14-Nov-05	13-Jun-19	165	4959.00	13 years-6 months	NO	Higher Education	
0809	Karunakara Reddy M	09-May-82	41.173142	Male	Engineering	Lead Engineer	14-Feb-11	12-Oct-18	93	2797.00	7 years-7 months	NO	Career Growth	Karnataka
1043	NAVYA THADATIIL	10-Oct-83	39.751519	Female	HR & Admin	Deputy Manager	28-Nov-11	24-Apr-18	78	2339.00	6 years-4 months	NO	Career Growth	Karnataka
1044	SUBBAREDDY K	23-Nov-77	45.631015	Male	Engineering	Associate Engineer	05-Dec-11	14-Oct-21	120	3601.00	9 years-10 months	NO	Personal Reason	Karnataka
1049	LIJIN SHAJI	31-Jan-82	41.44146	Male	Engineering	Associate Senior Manager	09-Jan-12	21-May-21	114	3420.00	9 years-4 months	NO	Personal Reason	Kerala
1063	YOGESHA Y S	19-Oct-85	37.726832	Male	Engineering	Associate Senior Manager	27-Feb-12	06-Feb-23	133	3997.00	10 years-11 months	NO	Better Opportunity	Karnataka
1086	BASAVANNEVVA CHOUDHARI	22-Jun-86	37.05267	Female	Engineering	Lead Engineer	11-Jun-12	01-Jun-20	97	2912.00	7 years-11 months	NO	Personal Reason	
1090	GURURAJAN R	31-Dec-85	37.526959	Male	Engineering	Associate Engineer	02-Jul-12	10-Jan-19	79	2383.00	6 years-6 months	NO	Better Opportunity	
1100	PRADEEP KUMAR B	01-Jan-85	38.523589	Male	Engineering	Lead Engineer	16-Jul-12	16-Jul-18	73	2191.00	6 years-0 months	NO	Career Growth	Karnataka
1118	VIKAS G HEGDE	07-Oct-82	40.759713	Male	Engineering	Senior Manager	17-Sep-12			-41169.00		YES		Karnataka
1172	MADHUSUDAN SAHOO	20-Jun-87	36.058162	Male	Engineering	Associate Architect	11-Mar-13	30-Jul-21	102	3063.00	8 years-4 months	NO	Career Growth	Odisha
1180	VIJAYALAXMI S ULVEKAR	10-Nov-85	37.666596	Female	Engineering	Architect	01-Apr-13			-41365.00		YES		
1182	VANGARA NAGAMANI	10-Jun-81	42.085514	Female	Engineering	Architect	01-Apr-13	06-Dec-21	106	3171.00	8 years-8 months	NO	Better Opportunity	Andhra Pradesh
1209	N HARIKRISHNARAJAN	09-Oct-74	48.754244	Male	Engineering	Associate Engineer	01-Jul-13	14-Jun-19	72	2174.00	5 years-11 months	NO	Career Growth	Tamil Nadu
1217	ANANTH K L	07-Jun-88	35.091034	Male	Engineering	Associate Engineer	22-Jul-13	22-Nov-18	65	1949.00	5 years-4 months	NO	Career Growth	Karnataka
1228	ASHWINI K.G	09-Mar-91	32.340164	Female	Engineering	Senior Software Engineer	19-Aug-13	31-Dec-18	65	1960.00	5 years-4 months	NO	Career Growth	
1230	PAVAN KUNTE A	13-Mar-91	32.329213	Male	Engineering	Senior Software Engineer	19-Aug-13	23-Nov-18	64	1922.00	5 years-3 months	NO	Career Growth	Karnataka
1233	KAVYA.R	07-May-92	31.175907	Female	Engineering	Senior Software Engineer	19-Aug-13	14-Aug-18	61	1821.00	4 years-11 months	NO	Career Growth	Karnataka
	DHANARAJ.													

Attribute Description

Attribute	Description	Type
Age	Age of Employees	Numerical Discrete
Gender	Gender of the person	Categorical
Department	1-Engineer,2-IT,3-Finance,……………	Categorical
Designation	1- Associate Engineer,2-Architect,3-Assistant manger,…………….	Categorical
Work period in Months	Numeric	Discrete
Joining Date	Date	dd//mm//yyyy
Leaving Date	Date	dd//mm//yyyy
Retention	1- Yes, 0 - No	Categorical

Data Preprocessing

- Removing the unwanted columns/fields in the dataset.
- Checking for Null values.
- Fill the Null values.
  - Data Type – Numerical – Use Mean / Median
  - Data Type – Text – Mode
- Change the categorical values to numerical.
  - Label encoder
  - Word to vector

Training and testing the Data into Model

```
[ ]  from sklearn.model_selection import train_test_split
     from sklearn.ensemble import GradientBoostingClassifier
     from sklearn.metrics import accuracy_score
     from sklearn.preprocessing import LabelEncoder
```

```
[ ]  # Splitting the dataset into input features (X) and target variable (y)
     X = data[['Age', 'Workperiod in Months', 'Gender', 'Department', 'Designation']]
     y = data['Retention']
```

```
[ ]  X_encoded = pd.get_dummies(X)
```

```
     # Convert the target variable to numerical labels
     label_encoder = LabelEncoder()
     y = label_encoder.fit_transform(y)
     print(y)
```

```
[0 0 0 ... 1 0 1]
```

```
[ ]  print(X_encoded)
```

```
[ ]  # Splitting the data into training and testing sets
     X_train, X_test, y_train, y_test = train_test_split(X_encoded, y, test_size=0.2,random_state=42)#,stratify = y
```

Choose the Machine Learning Model

For the above features and the target variable we can use the following model,

- GradientBoostingClassifier – 0.794
- LogisticRegression – 0.686
- XGBClassifier – 0.8

Hence here we have high accuracy in XGB Classifier.

GradientBoostingClassifier

```
# Create an instance of GradientBoostingClassifier
model = GradientBoostingClassifier()

# Train the model
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print('Accuracy:', accuracy)
```

```
Accuracy: 0.7940298507462686
```

LogisticRegression

```
from sklearn.linear_model import LogisticRegression
# Create an instance of LogisticRegression
model = LogisticRegression()

# Train the logistic regression model
model.fit(X_train, y_train)

# Make predictions on the test set using logistic regression
y_pred = model.predict(X_test)

# Calculate accuracy for logistic regression
accuracy = accuracy_score(y_test, y_pred)
print('Accuracy (Logistic Regression):', accuracy)
```

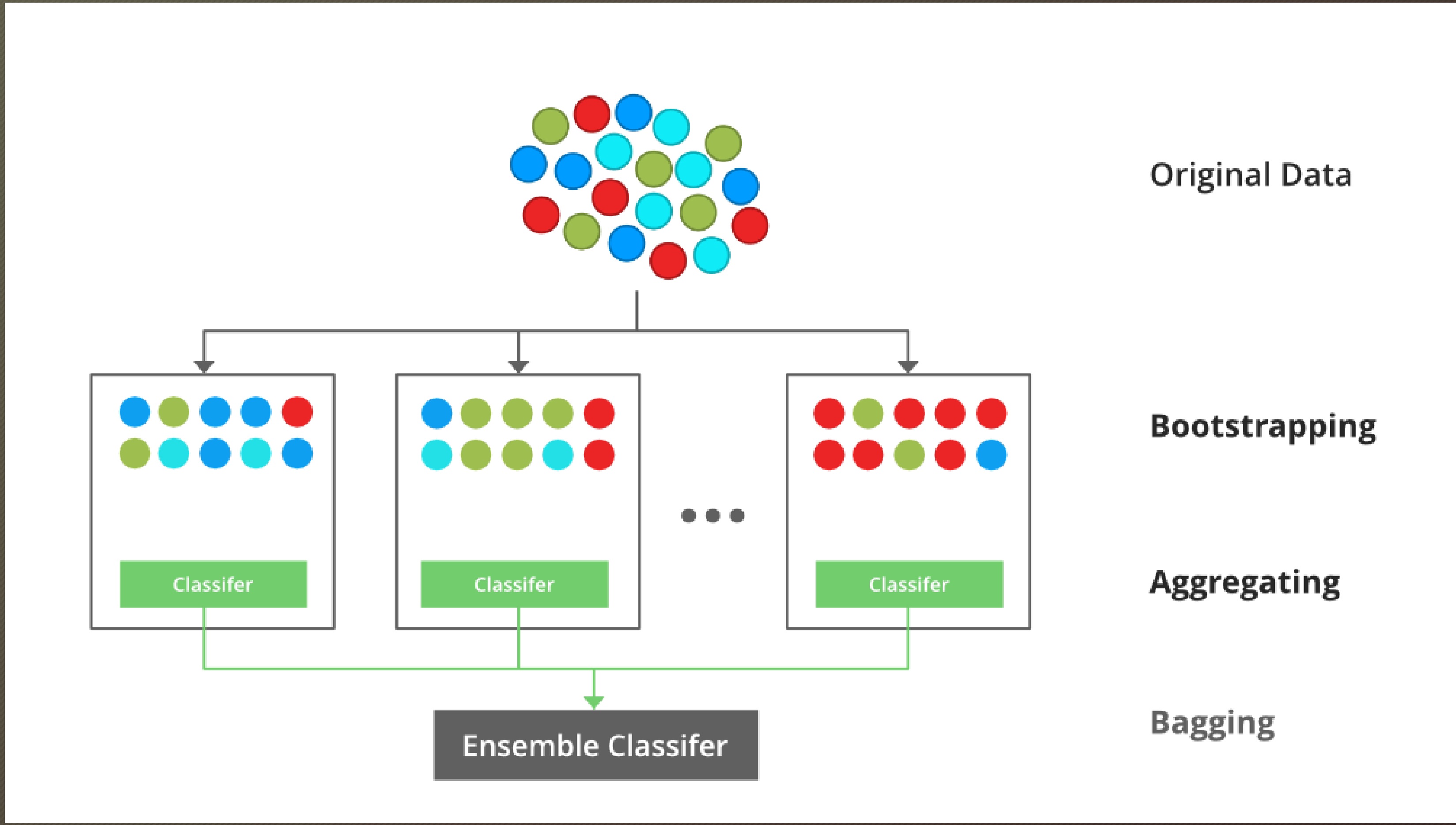
```
Accuracy (Logistic Regression): 0.6865671641791045
```

XGB Classifier

```
# Create an instance of XGBClassifier
model5 = xgb.XGBClassifier()

# Train the model
model5.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model5.predict(X_test)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print('Accuracy:', accuracy)
```

```
Accuracy: 0.8
```

About the M.L Model(XGB Classifier)



- XGBClassifier uses an ensemble of weak prediction models called decision trees. It builds trees sequentially, where each subsequent tree tries to correct the mistakes made by the previous trees. This process is known as gradient boosting.
- XGBClassifier optimizes an objective function that quantifies the model's performance. The objective function measures the difference between the predicted and actual values and guides the algorithm to minimize this difference.

Deploy the Model

To deploy the model first we need to load the model as a file(.jkl) for which we are using the joblib library function.

Save the model for further use

```
import joblib


# Save the trained model to a file
joblib.dump(model5, 'xgb_model.pkl')
```

```
['xgb_model.pkl']
```

```
[21] # Load the saved Random Forest Classifier
     model5 = joblib.load('xgb_model.pkl')
```

- After saving the model we need to get the user input.
- Then deploy the model with the user input data.

Deploying the Machine Learning Model

Where the user input is collected in the getvalue_.

```python
# Create a DataFrame from user input
#input_data = pd.DataFrame({'Age': [31],'Gender': ['Female'],'Department': ['Engineering'],
#'Designation': ['Senior Engineer'],'Workperiod in Months': [36]})

# Using the user input
input_data = pd.DataFrame({'Age': [getvalue3],
                           'Gender': [getvalue2],
                           'Department': [getvalue1],
                           'Designation': [getvalue],
                           'Workperiod in Months': [getvalue4]})

# Encoding categorical features using one-hot encoding
input_data_encoded = pd.get_dummies(input_data)
input_data_encoded = input_data_encoded.reindex(columns=X_train.columns, fill_value=0)


# Make the prediction using the trained model
retention_prediction = model5.predict(input_data_encoded)
print(retention_prediction)

# Display the predicted retention status
if retention_prediction == [1]:
    print("The employee is predicted to stay.")
else:
    print("The employee is predicted to leave.")
```

```
[0]
The employee is predicted to leave.
```

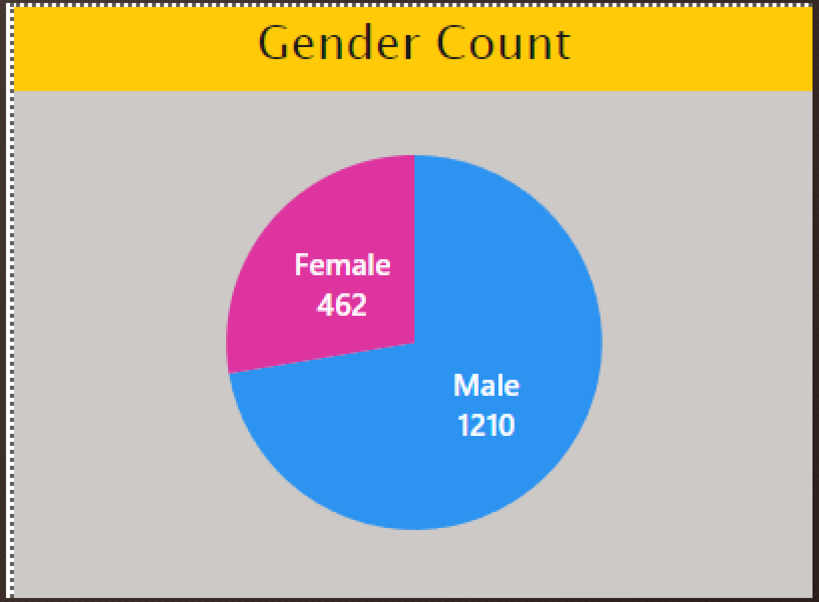
```python
# Create a DataFrame from user input
#input_data = pd.DataFrame({'Age': [34],'Gender': ['Male'],'Department': ['Engineering'],
#'Designation': ['Associate Architect'],'Workperiod in Months': [114]})

# Using the user input
input_data = pd.DataFrame({'Age': [getvalue3],
                           'Gender': [getvalue2],
                           'Department': [getvalue1],
                           'Designation': [getvalue],
                           'Workperiod in Months': [getvalue4]})

# Encoding categorical features using one-hot encoding
input_data_encoded = pd.get_dummies(input_data)
input_data_encoded = input_data_encoded.reindex(columns=X_train.columns, fill_value=0)


# Make the prediction using the trained model
retention_prediction = model5.predict(input_data_encoded)
print(retention_prediction)

# Display the predicted retention status
if retention_prediction == [1]:
    print("The employee is predicted to stay.")
else:
    print("The employee is predicted to leave.")
```

```
[1]
The employee is predicted to stay.
```

Visualization

- Data visualization is the most important step in the data analysis and prediction process, because the visuals would be easily captured by our brain then the text.
- We have used the Power BI software to visual the dataset.

- And this report is linked with the Power BI visual so we could directly have the interaction.



Employee Report

Total Employee

1672

Total Role

22

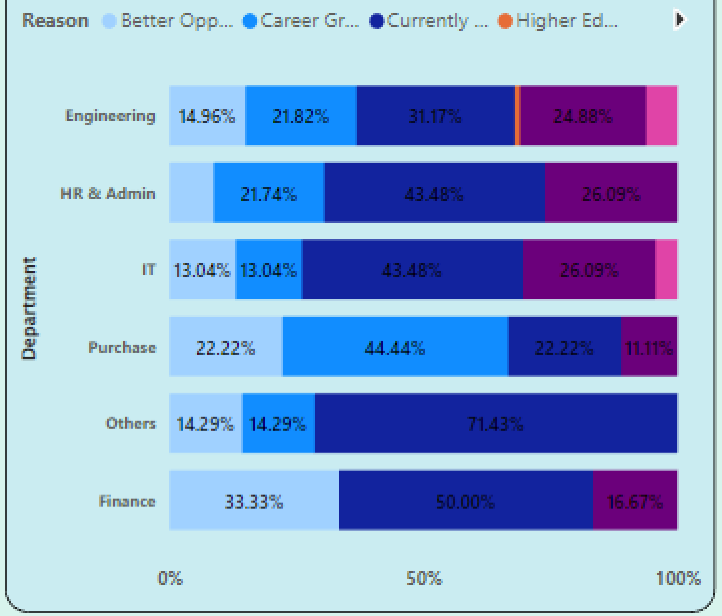
Avg Age

32

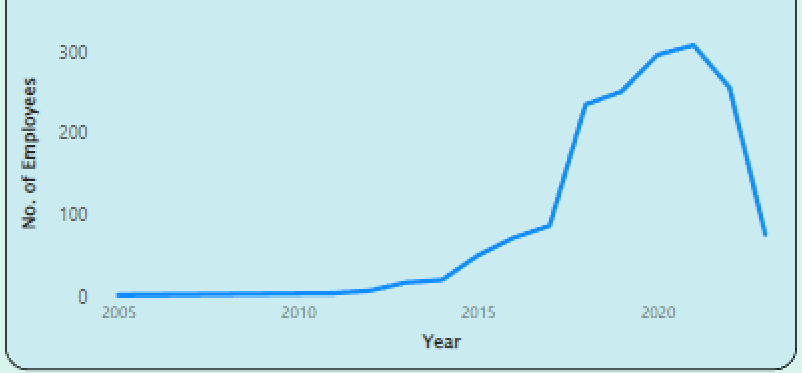
Department

Engineering | Finance | HR & Admin

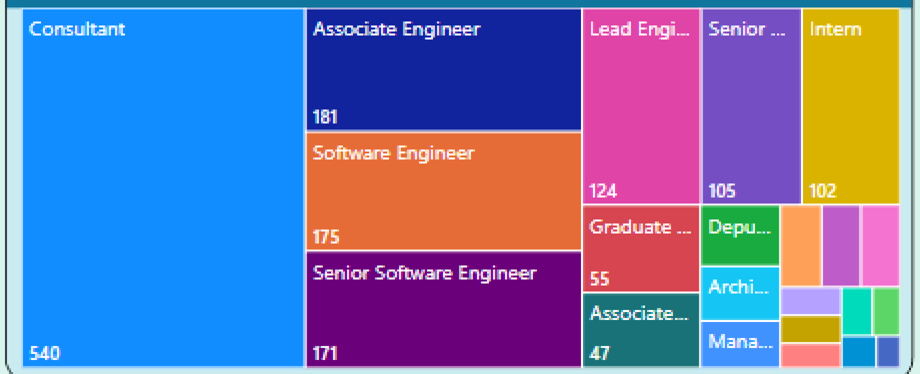
Employees leave based on their Reason



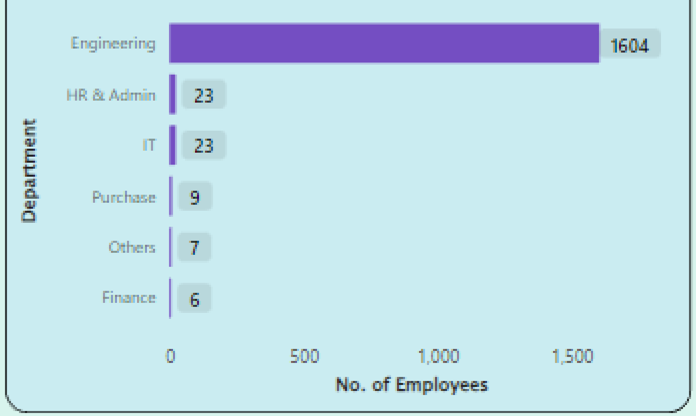
Employees Count Based on their Joining Date



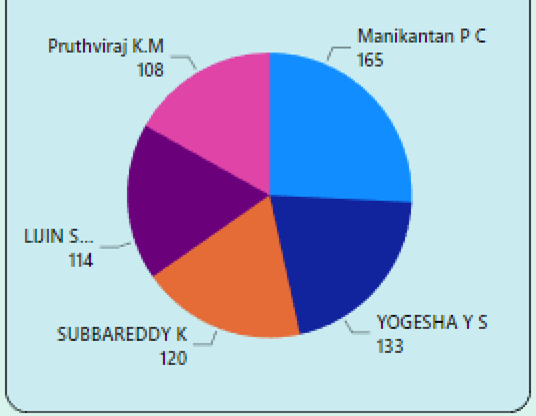
Dept Count by Designation



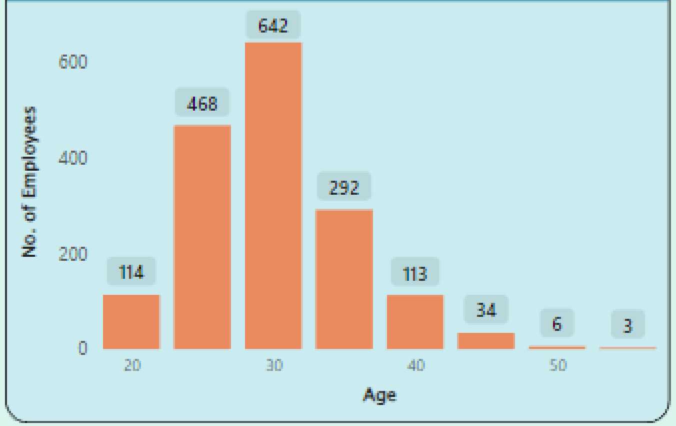
Employees Count Based on their Dept



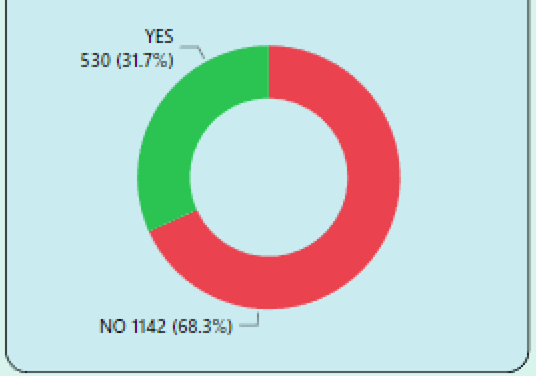
Top 5 Employee by month working



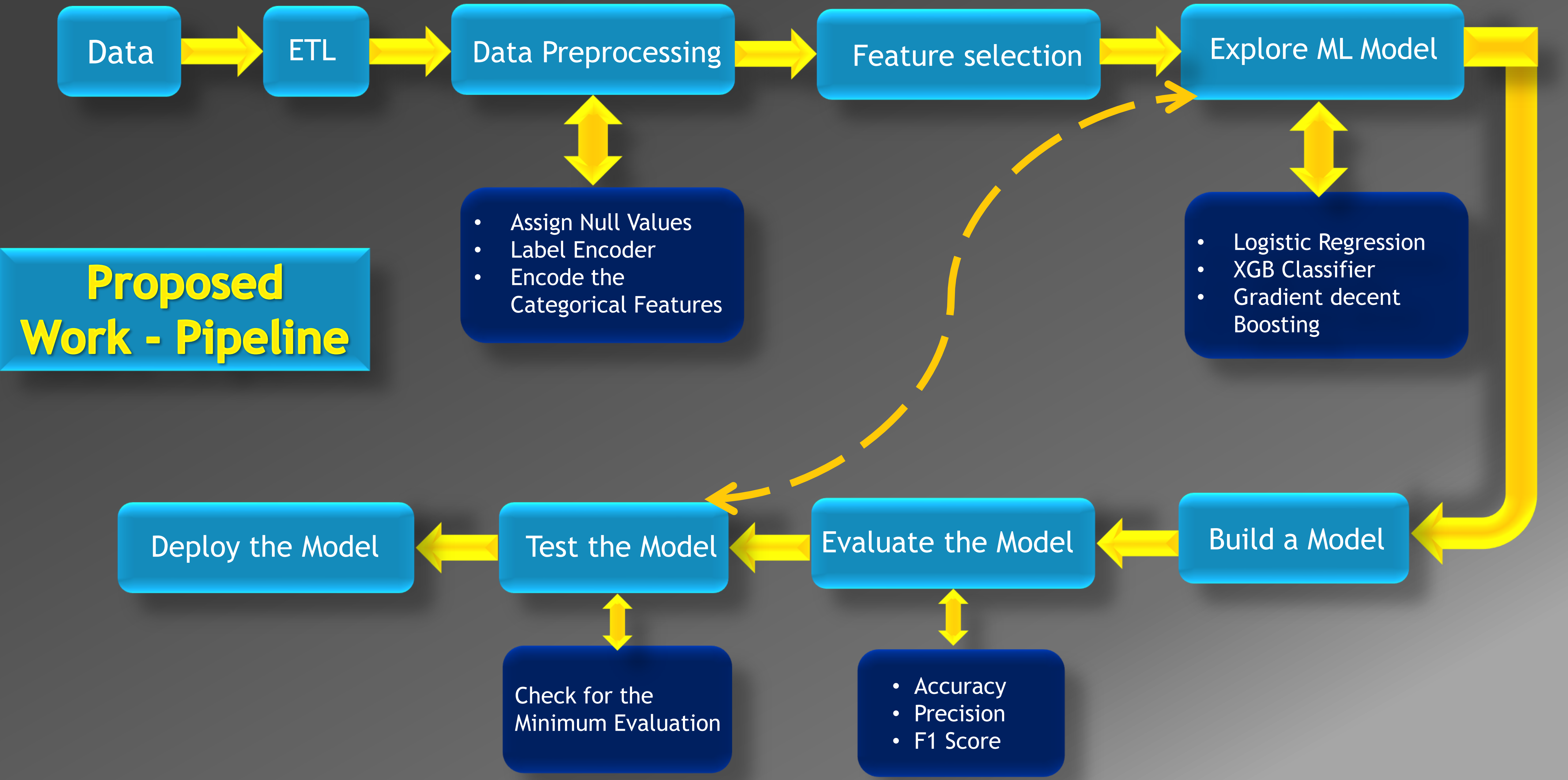
Employees Count Based on their Age



Employees Count Based on their Rention



Filters



Data
ETL
Data Preprocessing
Feature selection
Explore ML Model
Assign Null Values
Label Encoder
Encode the Categorical Features
Logistic Regression
XGB Classifier
Gradient decent Boosting
Proposed Work - Pipeline
Deploy the Model
Test the Model
Evaluate the Model
Build a Model
Check for the Minimum Evaluation
Accuracy
Precision
F1 Score

Conclusion

- Thus a complete end to end ML pipeline was explored for predicting the employee retention rate.
- The dataset is a good representative of the general workforce in today's organizations. The good
- results from multiple classifiers justify that the features chosen are causes that contribute to voluntary attrition.
- The XGBoost classifier performed well than other ML algorithms with a validation accuracy of 80%
- The reason for attrition of employees can't be exactly predicted, because each person would have different ideas for their future goals.
- Future work might include more number of attributes pertaining to the employee and a Sentiment Analysis can be made by collecting data from employees.

Thank You

- S.Jeevith Kumar