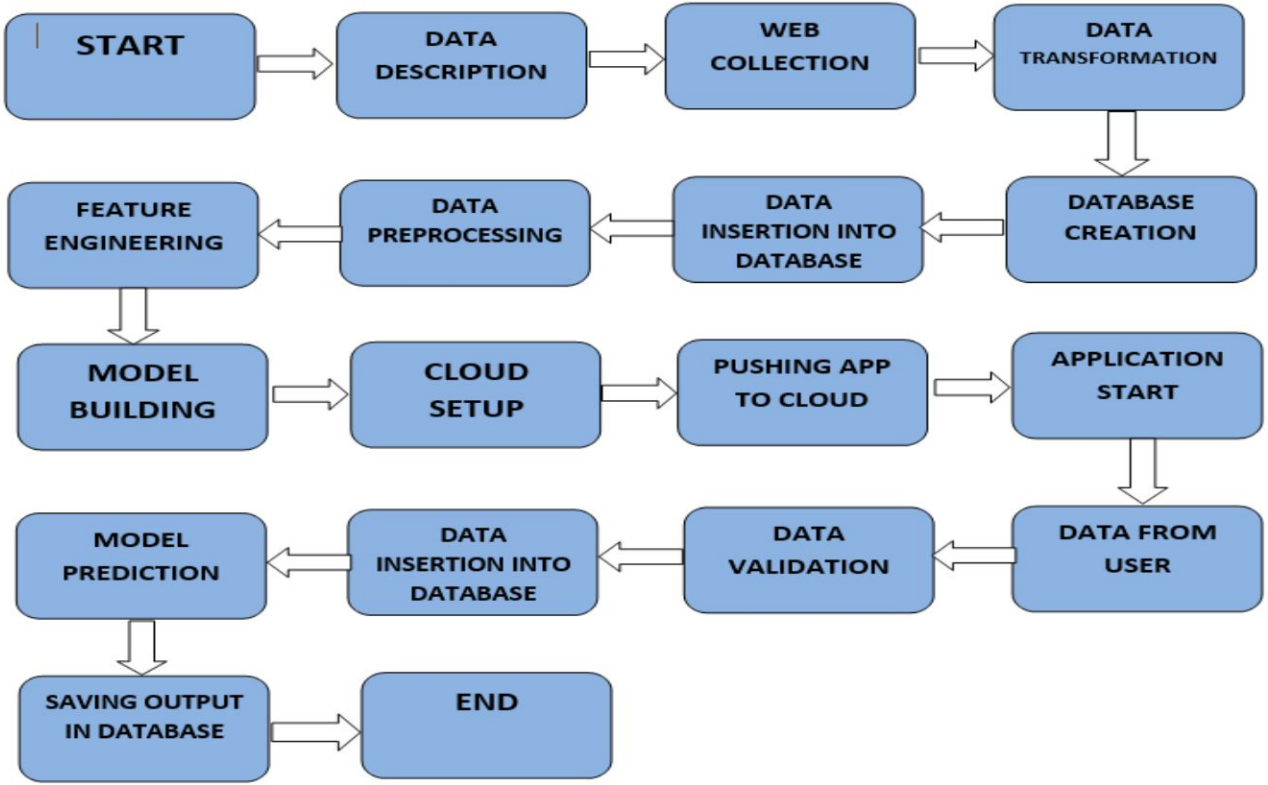
BackOrder Prediction

Introduction

Backorders are unavoidable, but by anticipating which things will be backordered, planning can be streamlined at several levels, preventing unexpected strain on production, logistics, and transportation. ERP systems generate a lot of data (mainly structured) and also contain a lot of historical data; if this data can be properly utilized, a predictive model to forecast backorders and plan accordingly can be constructed. Based on past data from inventories, supply chain, and sales, classify the products as going into backorder (Yes or No).

ARCHITECTURE



DATA ANALYSIS STEPS

DATA COLLECTION

In step 1, we collect data which is generally present in a database or on internet.

DATA PREPROCESSING

In step 2, we preprocess the data which involves data cleaning by handling outliers, null values etc.

EXPLORATORY DATA ANALYSIS

In step 3, we explore the data by performing univariate and bivariate analysis on the features.

FEATURE SELECTION

In step 4, we use feature selection techniques to filter out the most important features to perform model creation

MODEL CREATION AND EVALUATION

In step 5, we finally build models on our dataset and choose the model which gives the best accuracy.

RANDOM FOREST MODEL

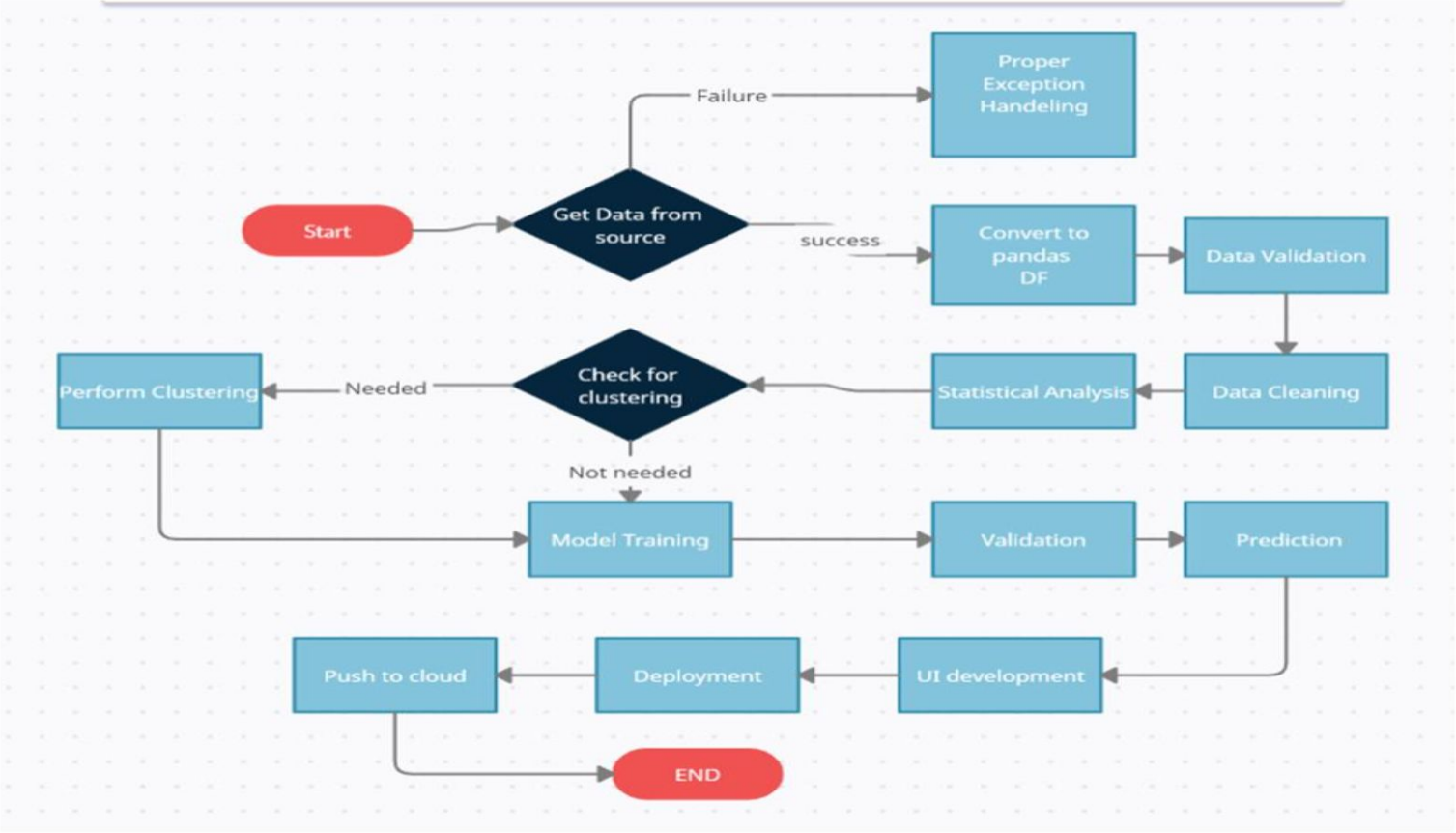
INTRODUCTION

- The random forest classifier is a supervised learning algorithm which we can use for regression and classification problems. It is among the most popular machine learning algorithms due to its high flexibility and ease of implementation.

- It is called Random Forest because it consists of multiple decision trees just as a forest has many trees. On top of that, it uses randomness to enhance its accuracy and combat overfitting, which can be a huge issue for such a sophisticated algorithm. These algorithms make decision trees based on a random selection of data samples and get predictions from every tree. After that, they select the best viable solution through votes.

- Random Forest Classifier being ensembled algorithm tends to give more accurate result. This is because it works on the principle i.e. number of weak estimators when combined forms strong estimator. Even if one or few decision trees are prone to noise, overall results would tend to be correct.
  Even with small number of estimators (=30), its gives us high accuracy as 97%.

MODEL TRAINING AND VALIDATION WORKFLOW



MODEL TRAINING AND VALIDATION

Data Collection:

Data was collected from Kaggle

Preprocessing

- Missing values imputed using Median strategy
- Outliers detection using boxplot and kdeplot and removed using percentile
- Label Encoding of Categorical Features
- Imbalanced data set was taken care by implementing Under-Sampling
- Feature selection using Chi_square and Random_Forest feature importance

MODEL TRAINING AND VALIDATION WORKFLOW

Model Creation and Evaluation

- Various classification algorithms like Logistic Regression, Random Forest, Decision Tree, Naïve Bayes, Support Vector Machine tested.
- Random Forest, Decision Tree and Logistic regression were given better results. Random Forest was chosen for the final model training and testing.
- Hyper parameter tuning was performed.
- Model performance evaluated based on accuracy, confusion matrix, classification report.

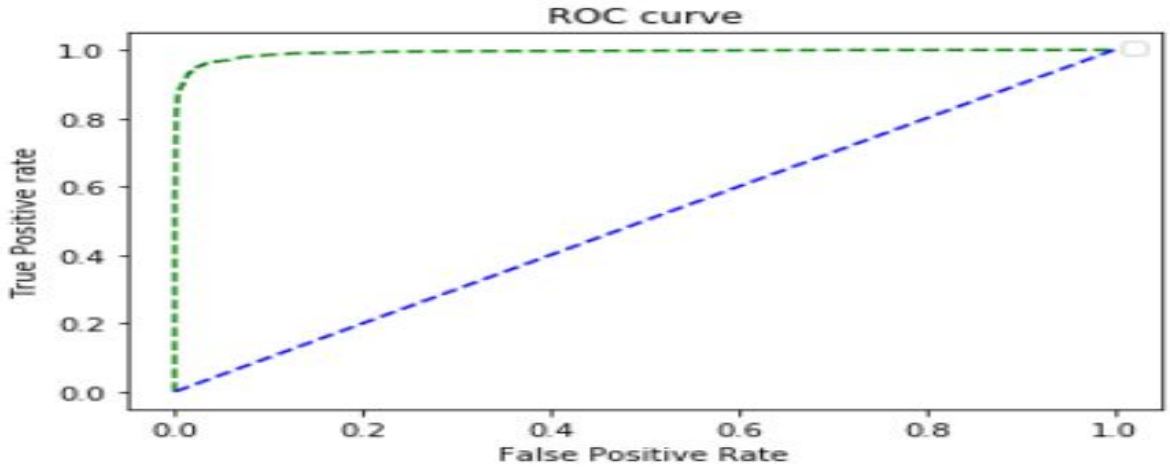
PREDICTION RESULT ON TEST DATA

```
--------CLassification Report-----------------------
              precision    recall  f1-score   support

           0       0.99      0.96      0.98      8420
           1       0.93      0.97      0.95      3939

    accuracy                           0.97     12359
   macro avg       0.96      0.97      0.96     12359
weighted avg       0.97      0.97      0.97     12359

--------Confusion Matrix------------------------
[[8123  297]
 [ 116 3823]]
----------Feature Imp------------------
```



DATABASE CONNECTION & DEPLOYMENT

Database Connection

- MongoDB Atlas database used for this project

Model Deployment

- The final model is deployed using on Heroku using Flask framework