INTRODUCTION

1.1 Introduction

Attrition is the departure of employees from the organization for any reason (voluntary or involuntary), including resignation, termination, death or retirement. Attrition rate is the rate at which employees leave an organization divided by the average number of employees at the organization over a given period of time. Employee attrition can happen for several reasons. These include unhappiness about employee benefits or the pay structure, a lack of employee development opportunities, and even poor conditions in the workplace.

1.2 Objective Of Project

The main objective of this research work is to develop a model that can help to predict whether an employee will leave the company or not. The essential idea is to measure the effectiveness of employee appraisal and satisfaction rates within the company, which can help to reduce the attrition rate of employees.

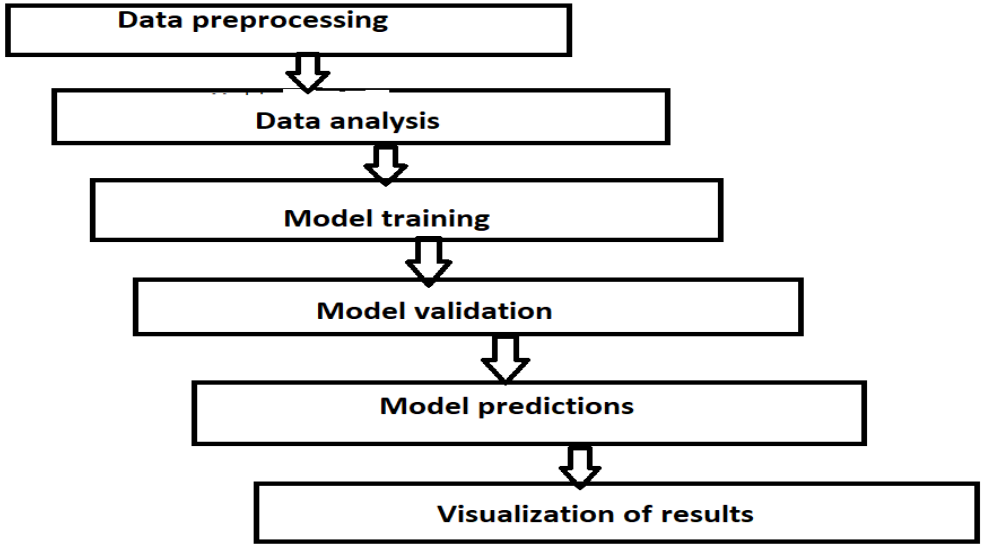
2.2 Requirement Analysis

- Data preprocessing
- Data analysis
- Model training
- Model validation
- Model predictions
- Visualization of results
- Model Deployment

SYSREM ANALYSIS & DESIGN

3.1 Type of SDLC Model Used

Waterfall Model:



3.2 UML Diagrams

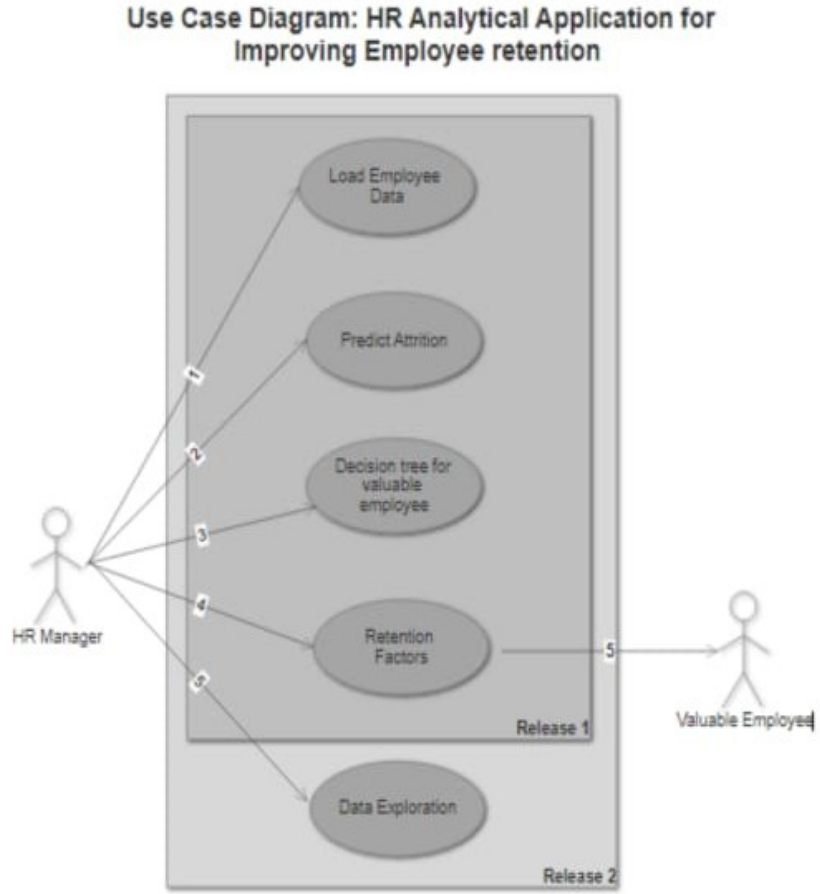


Figure 4.4 1: Use Case Diagram

4. SYSTEM IMPLEMENTATION

4.1 Implementation Details

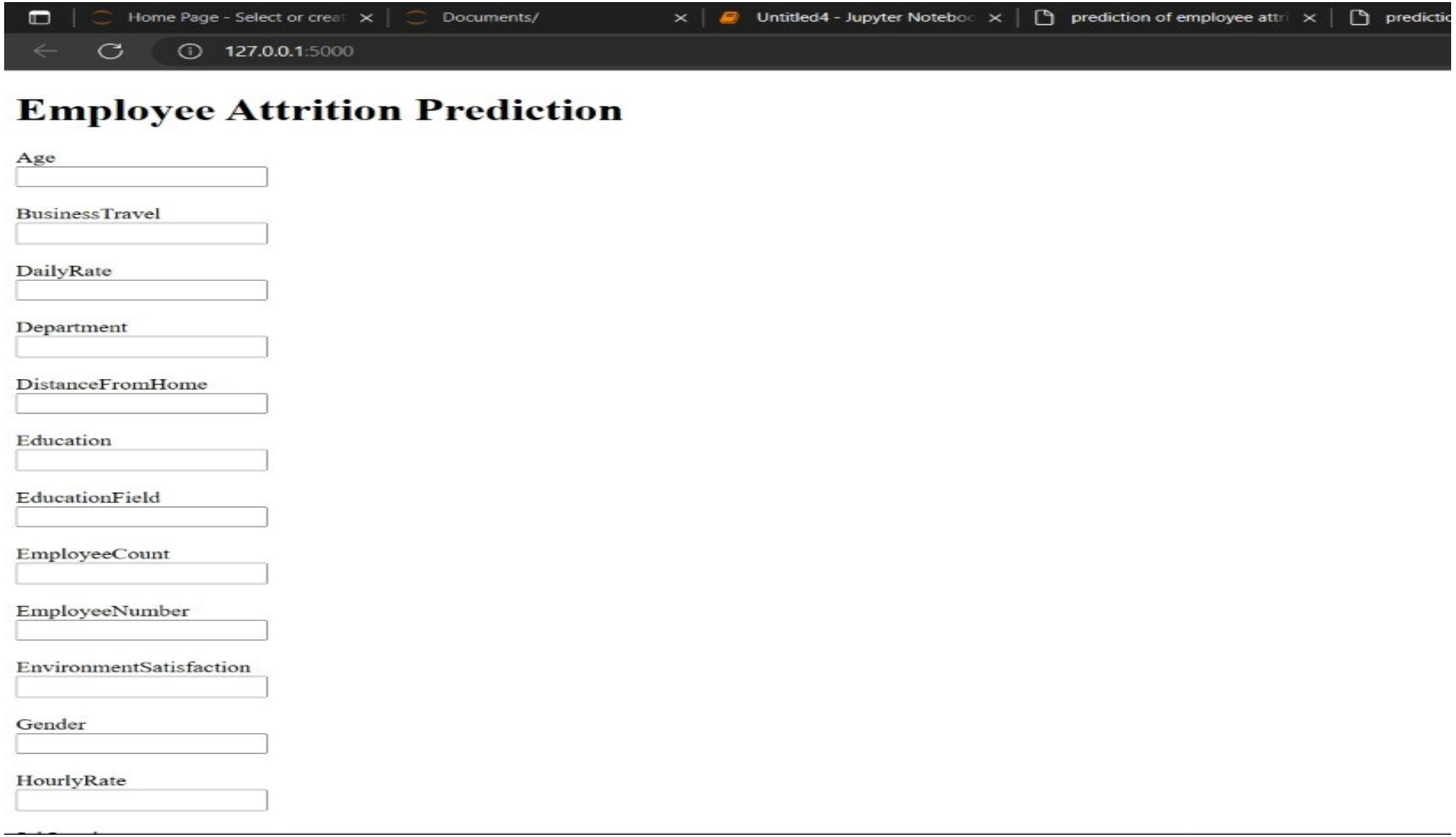
The Process for Classifications:

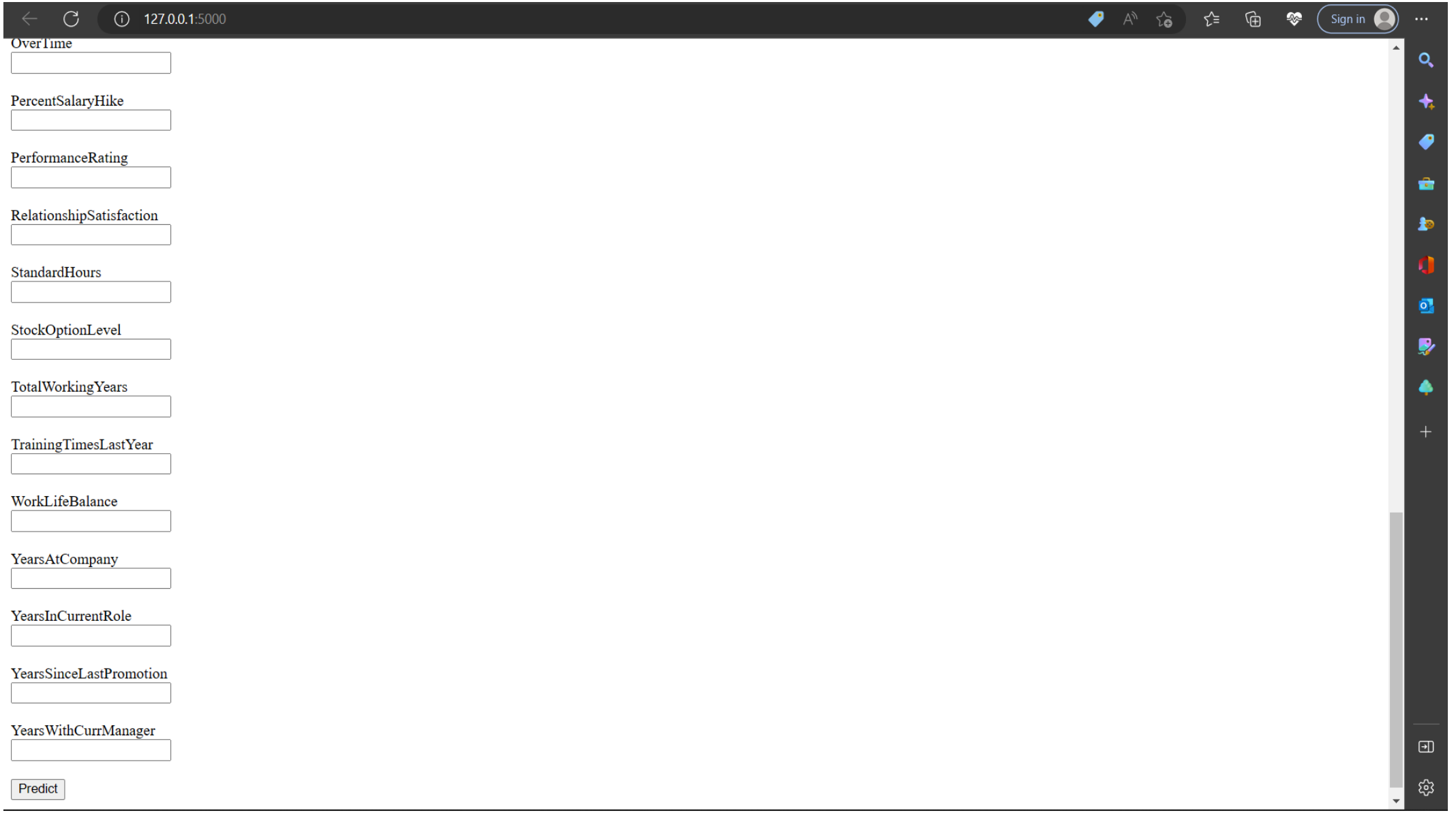
- Create an estimation sample and two validation samples by splitting the data into two groups.
- Set up the dependent variable, employee attrition (as a categorical 0-1 variable)
- Estimate the classification model using the estimation data, and interpret the results.
- Assess the accuracy of classification in the first validation sample, possibly repeating steps 2-5 a few times changing the classifier in different ways to increase performance.
- Finally, assess the accuracy of classification in the second validation sample. You should eventually use and report all relevant performance measures and plots on this second validation sample only.

CODING

```
from flask import Flask, render_template, request
import pickle
import numpy as np
import os
model = pickle.load(open("model.pkl", "rb"))
app =Flask(_name_, template_folder='template')
@app.route('/')
def index():
    return render_template('index1.html')
@app.route('/predict', methods=['POST'])
def predict_Attrition():
Age= int(request.form.get('Age'))
    BusinessTravel=str(request.form.get('BusinessTravel'))
    DailyRate=int(request.form.get('DailyRate'))
    Department=str(request.form.get('Department'))
    DistanceFromHome=int(request.form.get('DistanceFromHome'))
    Education=int(request.form.get('Education'))
    EducationField=str(request.form.get('EducationField'))
    EmployeeCount=int(request.form.get('EmployeeCount'))
```

```
EmployeeNumber=int(request.form.get('EmployeeNumber'))
EnvironmentSatisfaction=int(request.form.get('EnvironmentSatisfaction'))
Gender=str(request.form.get('Gender'))
HourlyRate=int(request.form.get('HourlyRate'))

X=np.array([[Age,BusinessTravel,DailyRate,Department,DistanceFromHome,Education,EducationField,Employ
eeCount,EmployeeNumber,EnvironmentSatisfaction,Gender,HourlyRate,]]).reshape(1,34)
    print(X)

result = model.predict(X)
return render_template('index1.html', prediction=result)

    if result[0] == 1:
 result="attrition"
    else:
result="not-Attrition“

if _name_ == "_main_":
    app.run(debug=True)
```


Home Page - Select or creat
Documents/
Untitled4 - Jupyter Noteboc
prediction of employee attri
predictic
127.0.0.1:5000
Employee Attrition Prediction
Age
BusinessTravel
DailyRate
Department
DistanceFromHome
Education
EducationField
EmployeeCount
EmployeeNumber
EnvironmentSatisfaction
Gender
HourlyRate


127.0.0.1:5000
Sign in
OverTime
PercentSalaryHike
PerformanceRating
RelationshipSatisfaction
StandardHours
StockOptionLevel
TotalWorkingYears
TrainingTimesLastYear
WorkLifeBalance
YearsAtCompany
YearsInCurrentRole
YearsSinceLastPromotion
YearsWithCurrManager
Predict

Documents/ | Untitled4 - Jupyter Notebook | New tab | Microsoft Edge

localhost:8888/notebooks/Documents/Untitled4.ipynb

jupyter Untitled4 Last Checkpoint: 12/10/2022 (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run | Code

In [1]:
```
!pip install hvplot
import hvplot#interactive visualization library
import pandas as pd#data cleaning and analysis
import numpy as np#array function
import matplotlib.pyplot as plt#graphical plotting library
import seaborn as sns# visualization
import hvplot.pandas

%matplotlib inline#for backkend function
sns.set_style("whitegrid")#background
plt.style.use("fivethirtyeight")#style

pd.set_option("display.float_format", "{:.2f}".format)
pd.set_option("display.max_columns", 80)
pd.set_option("display.max_rows", 80)
```

```
0)
Requirement already satisfied: pyct>=0.4.4 in c:\users\shital\anaconda3\lib\site-packages (from colorcet>=2->hvplot) (0.4.6)
Requirement already satisfied: panel>=0.8.0 in c:\users\shital\anaconda3\lib\site-packages (from holoviews>=1.11.0->hvplot)
(0.13.0)
Requirement already satisfied: pyviz-comms>=0.7.4 in c:\users\shital\anaconda3\lib\site-packages (from holoviews>=1.11.0->hvp
lot) (2.0.2)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\shital\anaconda3\lib\site-packages (from Jinja2>=2.9->bokeh>=1.0.
0->hvplot) (2.0.1)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in c:\users\shital\anaconda3\lib\site-packages (from packaging>=16.8-
>bokeh>=1.0.0->hvplot) (3.0.4)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\shital\anaconda3\lib\site-packages (from pandas->hvplot)
(2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\shital\anaconda3\lib\site-packages (from pandas->hvplot) (2021.3)
Requirement already satisfied: bleach in c:\users\shital\anaconda3\lib\site-packages (from panel>=0.8.0->holoviews>=1.11.0->h
vplot) (4.1.0)
Requirement already satisfied: markdown in c:\users\shital\anaconda3\lib\site-packages (from panel>=0.8.0->holoviews>=1.11.0-
>hvplot) (3.3.4)
Requirement already satisfied: requests in c:\users\shital\anaconda3\lib\site-packages (from panel>=0.8.0->holoviews>=1.11.0-
>hvplot) (2.27.1)
Requirement already satisfied: tqdm>=4.48.0 in c:\users\shital\anaconda3\lib\site-packages (from panel>=0.8.0->holoviews>=1.1
```

In [3]:
```
df = pd.read_csv("D:/shital/WA_Fn-UseC_-HR-Employee-Attrition.csv")
```

Home Page - Select or create a n | Documents/ | Untitled4 - Jupyter Notebook

localhost:8889/notebooks/Documents/Untitled4.ipynb

Sign in

jupyter Untitled4 Last Checkpoint: 12/10/2022 (autosaved)

Logout

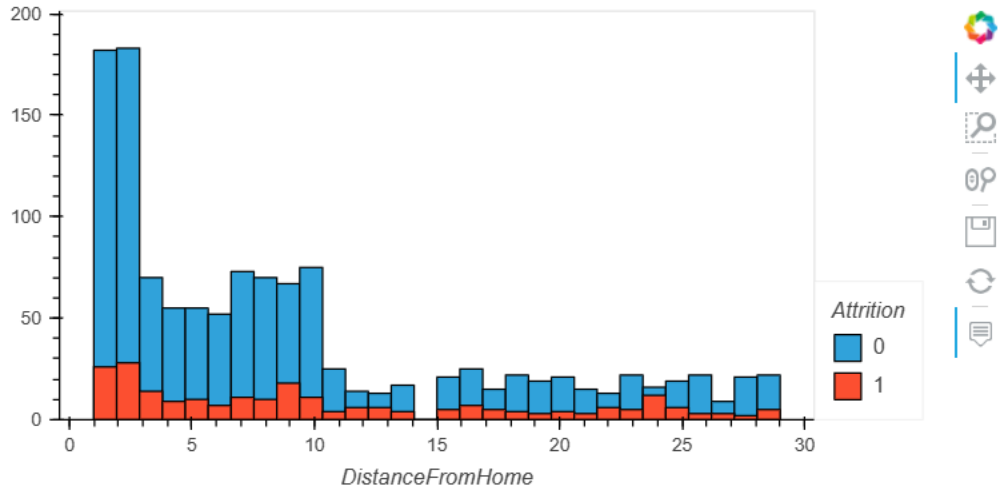
File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run | Code

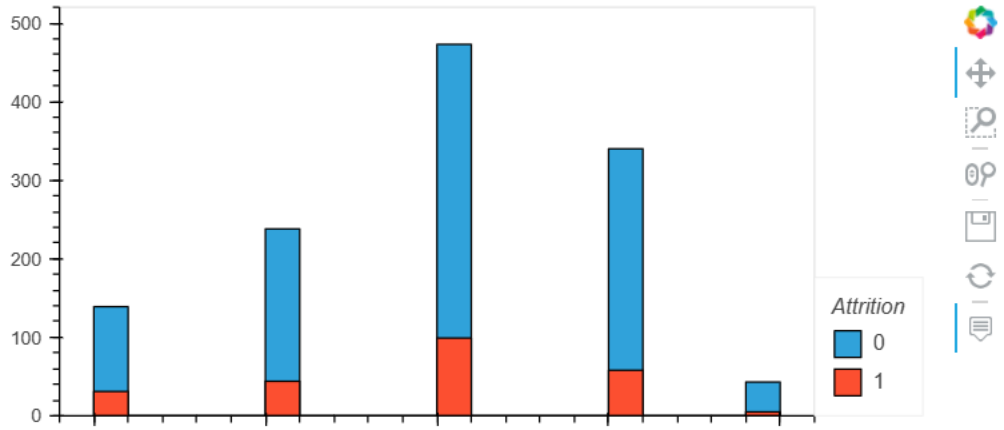
In [13]:
```
df.hvplot.hist(y='DistanceFromHome', by='Attrition', subplots=False, width=600, height=300, bins=30)
```

Out[13]:



In [14]:
```
df.hvplot.hist(y='Education', by='Attrition', subplots=False, width=600, height=300)
```

Out[14]:



Documents/ | Untitled4 - Jupyter Notebook | New tab | Microsoft Edge

localhost:8888/notebooks/Documents/Untitled4.ipynb

Sign in

jupyter **Untitled4** Last Checkpoint: 12/10/2022 (unsaved changes)

Logout

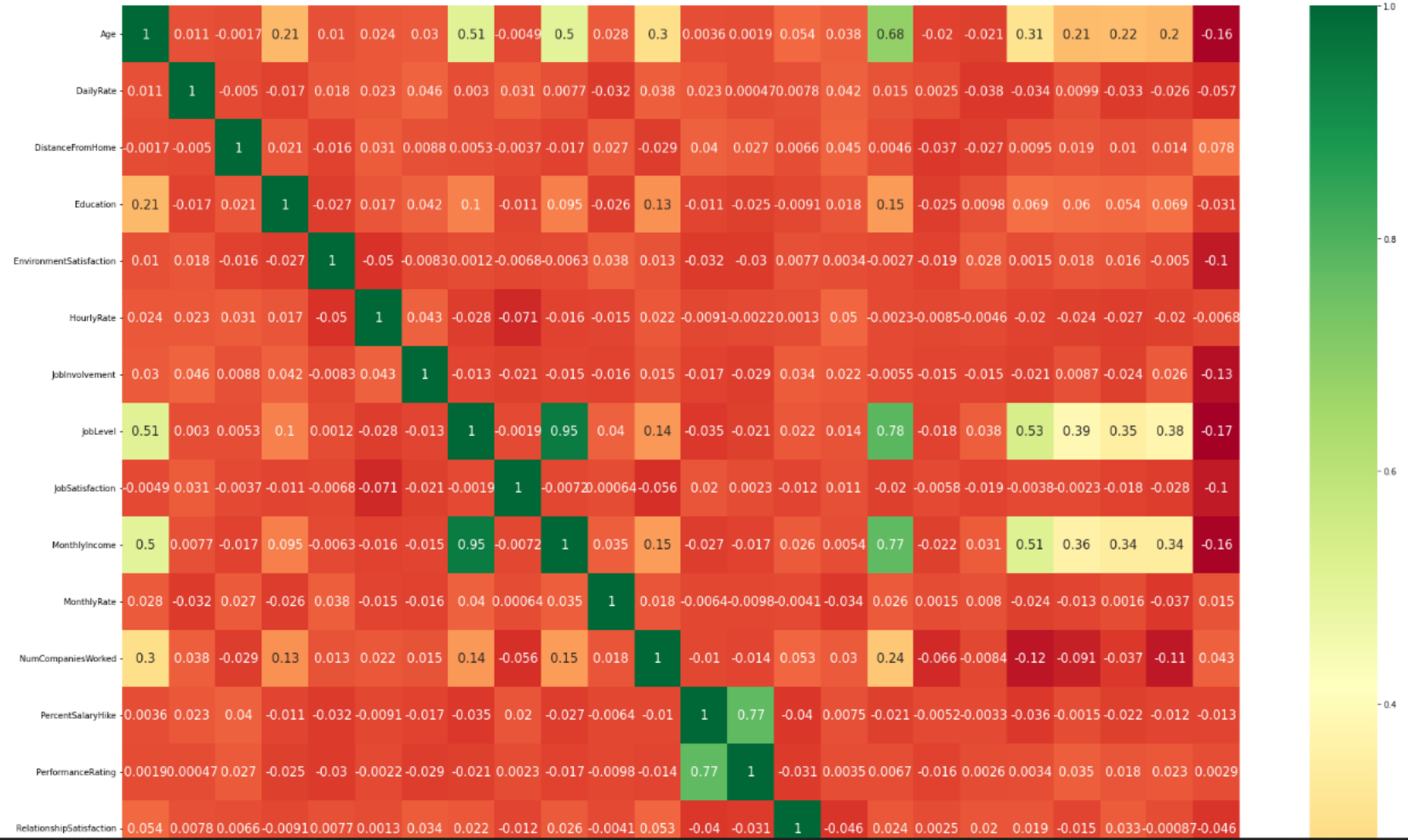
File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run | Code

In [28]:
```
plt.figure(figsize=(30, 30))
sns.heatmap(df.corr(), annot=True, cmap="RdYlGn", annot_kws={"size":15})
```

Out[28]: <AxesSubplot:>



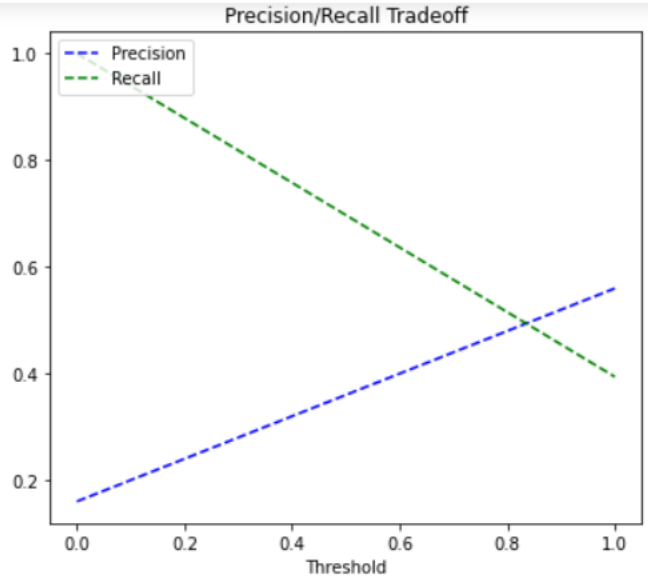
jupyter Untitled4 Last Checkpoint: 12/10/2022 (autosaved)

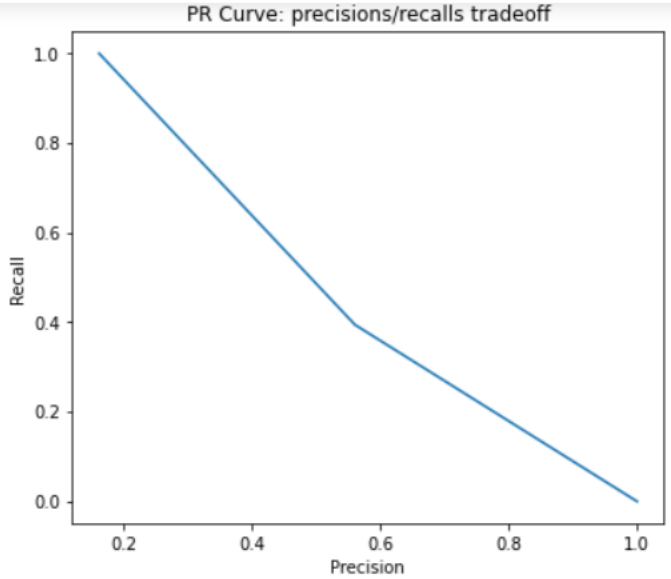
Logout

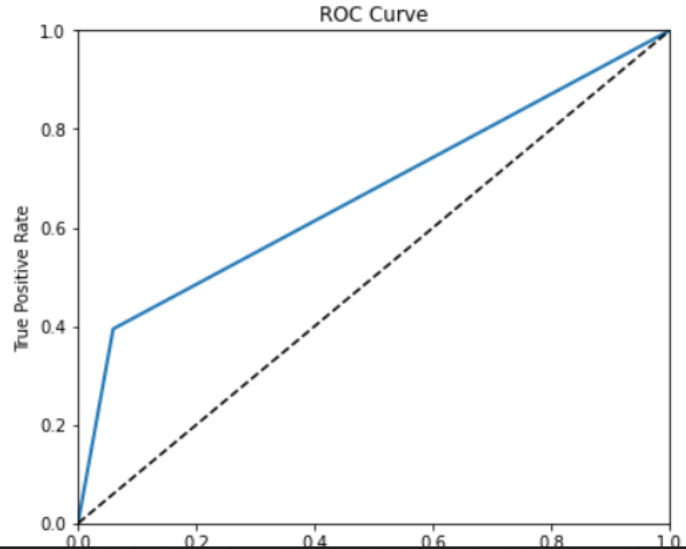
File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run | Code







jupyter Untitled4 Last Checkpoint: 12/10/2022 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run | Code

In [41]:
```
from sklearn.linear_model import LogisticRegression#apply on linearly seperable data

lr_clf = LogisticRegression(solver='liblinear', penalty='l1')
lr_clf.fit(X_train_std, y_train)

evaluate(lr_clf, X_train_std, X_test_std, y_train, y_test)
```

```
TRAINIG RESULTS:
===============================
CONFUSION MATRIX:
[[849  14]
 [ 59 107]]
ACCURACY SCORE:
0.9291
CLASSIFICATION REPORT:
                    0           1  accuracy    macro avg  weighted avg
precision    0.935022    0.884298  0.929057     0.909660      0.926839
recall       0.983778    0.644578  0.929057     0.814178      0.929057
f1-score     0.958780    0.745645  0.929057     0.852212      0.924397
support    863.000000  166.000000  0.929057  1029.000000   1029.000000
TESTING RESULTS:
===============================
CONFUSION MATRIX:
[[348  22]
 [ 43  28]]
ACCURACY SCORE:
0.8526
CLASSIFICATION REPORT:
                    0          1  accuracy   macro avg  weighted avg
precision    0.890026   0.560000  0.852608    0.725013      0.836892
recall       0.940541   0.394366  0.852608    0.667453      0.852608
f1-score     0.914586   0.462810  0.852608    0.688698      0.841851
support    370.000000  71.000000  0.852608  441.000000    441.000000
```

In [42]:
```
from sklearn.metrics import precision_recall_curve, roc_curve

def plot_precision_recall_vs_threshold(precisions, recalls, thresholds):
    plt.plot(thresholds, precisions[:-1], "b--", label="Precision")
    plt.plot(thresholds, recalls[:-1], "g--", label="Recall")
```

Home Page - Select or create a n | Documents/ | Untitled4 - Jupyter Notebook

localhost:8889/notebooks/Documents/Untitled4.ipynb

jupyter Untitled4 Last Checkpoint: 12/10/2022 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run | Code

In [45]:
```
param_grid = dict(
    n_estimators= [10, 50, 90],
    max_features= ['auto', 'sqrt'],
    max_depth= [2, 3, 5, 10, 15, None],
    min_samples_split= [2, 5, 10],
    min_samples_leaf= [1, 2, 4],
    bootstrap= [True, False]
)

rf_clf = RandomForestClassifier(random_state=42)
search = GridSearchCV(rf_clf, param_grid=param_grid, scoring='roc_auc', cv=5, verbose=1, n_jobs=-1)
search.fit(X_train, y_train)

rf_clf = RandomForestClassifier(**search.best_params_, random_state=42)
rf_clf.fit(X_train, y_train)
evaluate(rf_clf, X_train, X_test, y_train, y_test)
```

```
Fitting 5 folds for each of 648 candidates, totalling 3240 fits
TRAINIG RESULTS:
===============================
CONFUSION MATRIX:
[[863   0]
 [ 49 117]]
ACCURACY SCORE:
0.9524
CLASSIFICATION REPORT:
                    0           1  accuracy    macro avg  weighted avg
precision    0.946272    1.000000  0.952381     0.973136      0.954939
recall       1.000000    0.704819  0.952381     0.852410      0.952381
f1-score     0.972394    0.826855  0.952381     0.899625      0.948916
support    863.000000  166.000000  0.952381  1029.000000   1029.000000
TESTING RESULTS:
===============================
CONFUSION MATRIX:
[[363   7]
 [ 63   8]]
ACCURACY SCORE:
0.8413
CLASSIFICATION REPORT:
                    0           1  accuracy    macro avg  weighted avg
```

RESULT

5.1 Comparative result to existing system

This phase evaluated the qualities of the adopted models. The results of the decisions made inthe prediction phase were collected, for each algorithm, in the relative “confusion matrix”. This is a matrix where the values predicted by the classifier are shown in the columns and the real values of each instance of the test-set are shown in rows. These metrics,summarised and this are based on the number of errors and correct answers formulated by the classifier.