

Name- Akash Kumar  
Internship Program- Data Science With Machine Learning and Python  
Batch- Jan 2022 – Mar 2022  
Certificate Code- TCRIB2R43  
Date of submission- 3 APRIL 2022



## (HR Employee Attrition Data Analysis)

A Case-Study Submitted for the requirement of  
**Technical Coding Research Innovation**

For the Internship Project work done during  
**DATA SCIENCE WITH MACHINE LEARNING AND PYTHON  
INTERNSHIP PROGRAM**

by  
AKASH KUMAR (TCRIB2R43)

Rutuja Doiphode  
**CO-FOUNDER &CEO**  
TCR innovation.

Name- Akash Kumar  
Internship Program- Data Science With Machine Learning and Python  
Batch- Jan 2022 – Mar 2022  
Certificate Code- TCRIB2R43  
Date of submission- 3 APRIL 2022

# Preparation of Papers in Two-Column Format for Conference Proceedings Published by IEEE<sup>1</sup>

Bart Simpson and Homer Simpson  
Department of Nuclear Power Engineering  
University of Springfield  
Springfield, Nostate 12345, USA  
{bart.simpson & homer.simpson}@uspringfield.edu

Monkey King, Bajie Zhu and Seng Tang  
Department of Intelligent Robotics  
University of Huaguoshan  
Huaguoshan, Jilishijie Province, China  
monkey.king@uhuaguoshan.edu.cn

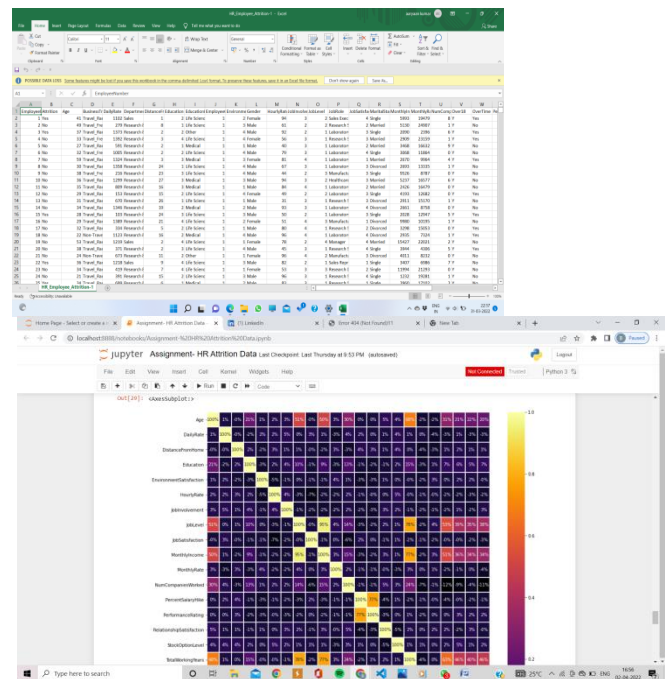
*Aim-we need to predict whether a given employee will leave the organization or not*

## INTRODUCTION:

we will be predicting whether an employee will leave the organization or not.our target column is attrition.

- 1.Import the relevant packages
- 2.Download and explore the datasheet
- 3.Perform EDA,apply dataset the processing
- 4.Predict the target column.

## HR DATASHEET



<sup>1</sup> This work is partially supported by NSF Grant #2003168 to H. Simpson and CNSF Grant #9972988 to M. King.

Name- Akash Kumar  
 Internship Program- Data Science With Machine Learning and Python  
 Batch- Jan 2022 – Mar 2022  
 Certificate Code- TCRIB2R43  
 Date of submission- 3 APRIL 2022

```

Building and testing model by applying Random Forest ML algorithm

In [ ]: rfc = RandomForestClassifier(n_estimators=100, random_state=0)
rfc.fit(X_train, y_train)
y_pred = rfc.predict(X_test)

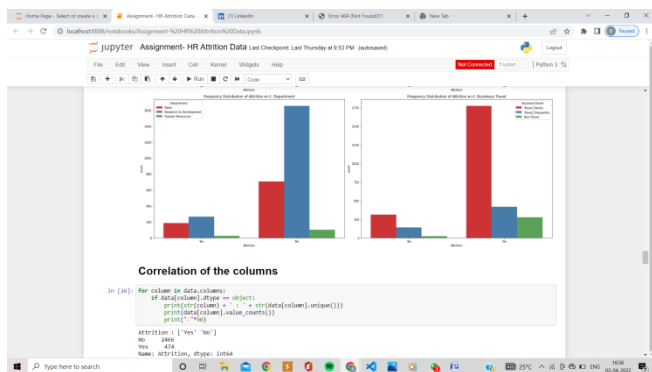
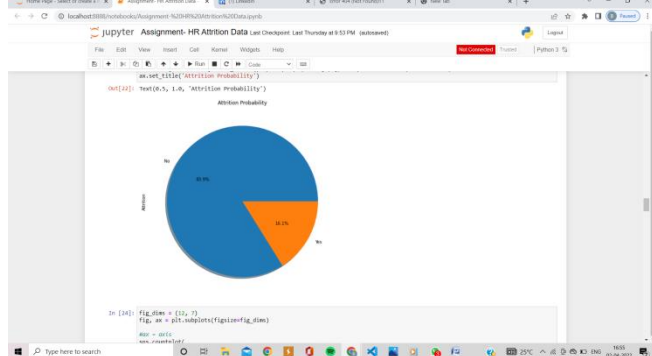
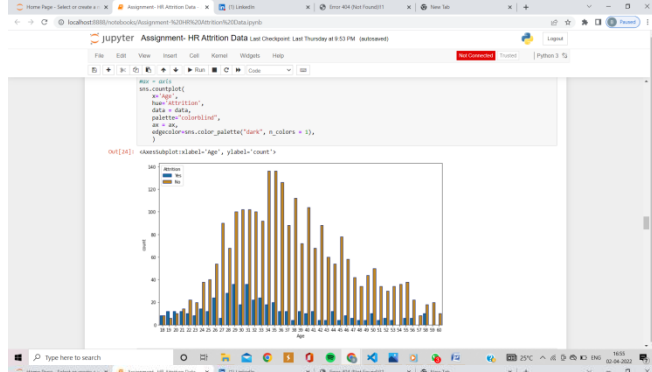
In [ ]: score = accuracy_score(y_test, y_pred)
print("RandomForest Classifier score: ", np.abs(score)*100)

In [ ]: y_pred

final result
Accuracy score = 95.23

prepared by
Vikash kumar

In [ ]:
  
```



Correlation of the columns

```

In [24]: for column in data.columns:
if data[column].dtype == object:
print(column, " = ", str(data[column].unique()))
print(data[column].value_counts())
print("\n")

attrition | "Yes" "No"
Yes
4314
No
4314
Name: attrition, dtype: object
  
```

EmployeeID	BusinessAge	YearsSinceLastPromotion	YearsSinceLastSalaryIncrease	YearsSinceLastJobChange	YearsSinceLastEducationChange	YearsSinceLastMaritalStatusChange	YearsSinceLastReligionChange	YearsSinceLastEthnicityChange	YearsSinceLastGenderChange	YearsSinceLastSexualOrientationChange	YearsSinceLastDisabilityChange	YearsSinceLastAncestryChange	YearsSinceLastLanguageChange	YearsSinceLastNativeLanguageChange	YearsSinceLastCountryChange	YearsSinceLastCityChange	YearsSinceLastStateChange	YearsSinceLastZipChange	YearsSinceLastHouseholdSizeChange	YearsSinceLastHouseholdIncomeChange	YearsSinceLastHouseholdAssetsChange	YearsSinceLastHouseholdDebtChange	YearsSinceLastHouseholdChildrenChange	YearsSinceLastHouseholdPetsChange	YearsSinceLastHouseholdVehiclesChange	YearsSinceLastHouseholdUtilitiesChange	YearsSinceLastHouseholdInsuranceChange	YearsSinceLastHouseholdHealthcareChange	YearsSinceLastHouseholdEducationChange	YearsSinceLastHouseholdEmploymentChange	YearsSinceLastHouseholdIncomeSourceChange	YearsSinceLastHouseholdIncomeLevelChange	YearsSinceLastHouseholdIncomeTypeChange	YearsSinceLastHouseholdIncomeFrequencyChange	YearsSinceLastHouseholdIncomeStabilityChange	YearsSinceLastHouseholdIncomeGrowthChange	YearsSinceLastHouseholdIncomeRiskChange	YearsSinceLastHouseholdIncomeVolatilityChange	YearsSinceLastHouseholdIncomeCorrelationChange	YearsSinceLastHouseholdIncomeCausalityChange	YearsSinceLastHouseholdIncomePredictabilityChange	YearsSinceLastHouseholdIncomeControllabilityChange	YearsSinceLastHouseholdIncomeFlexibilityChange	YearsSinceLastHouseholdIncomeAdaptabilityChange	YearsSinceLastHouseholdIncomeResilienceChange	YearsSinceLastHouseholdIncomeSustainabilityChange	YearsSinceLastHouseholdIncomeEthicalityChange	YearsSinceLastHouseholdIncomeLegitimacyChange	YearsSinceLastHouseholdIncomeTransparencyChange	YearsSinceLastHouseholdIncomeAccountabilityChange	YearsSinceLastHouseholdIncomeResponsibilityChange	YearsSinceLastHouseholdIncomeIntegrityChange	YearsSinceLastHouseholdIncomeHonestyChange	YearsSinceLastHouseholdIncomeFairnessChange	YearsSinceLastHouseholdIncomeJusticeChange	YearsSinceLastHouseholdIncomeEquityChange	YearsSinceLastHouseholdIncomeBalanceChange	YearsSinceLastHouseholdIncomeHarmonyChange	YearsSinceLastHouseholdIncomePeaceChange	YearsSinceLastHouseholdIncomeLoveChange	YearsSinceLastHouseholdIncomeCompassionChange	YearsSinceLastHouseholdIncomeKindnessChange	YearsSinceLastHouseholdIncomeGenerosityChange	YearsSinceLastHouseholdIncomeGratitudeChange	YearsSinceLastHouseholdIncomeOptimismChange	YearsSinceLastHouseholdIncomePositivityChange	YearsSinceLastHouseholdIncomeHopeChange	YearsSinceLastHouseholdIncomeFaithChange	YearsSinceLastHouseholdIncomeTrustChange	YearsSinceLastHouseholdIncomeRespectChange	YearsSinceLastHouseholdIncomeToleranceChange	YearsSinceLastHouseholdIncomePatienceChange	YearsSinceLastHouseholdIncomeHumilityChange	YearsSinceLastHouseholdIncomeModestyChange	YearsSinceLastHouseholdIncomeSimplicityChange	YearsSinceLastHouseholdIncomeClarityChange	YearsSinceLastHouseholdIncomeFocusChange	YearsSinceLastHouseholdIncomeDeterminationChange	YearsSinceLastHouseholdIncomePerseveranceChange	YearsSinceLastHouseholdIncomeCourageChange	YearsSinceLastHouseholdIncomeStrengthChange	YearsSinceLastHouseholdIncomePowerChange	YearsSinceLastHouseholdIncomeInfluenceChange	YearsSinceLastHouseholdIncomeAuthorityChange	YearsSinceLastHouseholdIncomeLeadershipChange	YearsSinceLastHouseholdIncomeMentorshipChange	YearsSinceLastHouseholdIncomeCoachingChange	YearsSinceLastHouseholdIncomeTeachingChange	YearsSinceLastHouseholdIncomeLearningChange	YearsSinceLastHouseholdIncomeGrowthMindsetChange	YearsSinceLastHouseholdIncomeGrowthOrientationChange	YearsSinceLastHouseholdIncomeGrowthPotentialChange	YearsSinceLastHouseholdIncomeGrowthOpportunityChange	YearsSinceLastHouseholdIncomeGrowthChallengeChange	YearsSinceLastHouseholdIncomeGrowthObstacleChange	YearsSinceLastHouseholdIncomeGrowthBarrierChange	YearsSinceLastHouseholdIncomeGrowthLimitationChange	YearsSinceLastHouseholdIncomeGrowthConstraintChange	YearsSinceLastHouseholdIncomeGrowthRestrictionChange	YearsSinceLastHouseholdIncomeGrowthProhibitionChange	YearsSinceLastHouseholdIncomeGrowthBanChange	YearsSinceLastHouseholdIncomeGrowthNoChange	YearsSinceLastHouseholdIncomeGrowthUnchangeChange	YearsSinceLastHouseholdIncomeGrowthNoChangeChange	YearsSinceLastHouseholdIncomeGrowthUnchangeNoChangeChange
------------	-------------	-------------------------	------------------------------	-------------------------	-------------------------------	-----------------------------------	------------------------------	-------------------------------	----------------------------	---------------------------------------	--------------------------------	------------------------------	------------------------------	------------------------------------	-----------------------------	--------------------------	---------------------------	-------------------------	-----------------------------------	-------------------------------------	-------------------------------------	-----------------------------------	---------------------------------------	-----------------------------------	---------------------------------------	----------------------------------------	----------------------------------------	-----------------------------------------	----------------------------------------	-----------------------------------------	-------------------------------------------	------------------------------------------	-----------------------------------------	----------------------------------------------	----------------------------------------------	-------------------------------------------	-----------------------------------------	-----------------------------------------------	------------------------------------------------	----------------------------------------------	---------------------------------------------------	----------------------------------------------------	------------------------------------------------	-------------------------------------------------	-----------------------------------------------	---------------------------------------------------	-----------------------------------------------	-----------------------------------------------	-------------------------------------------------	---------------------------------------------------	---------------------------------------------------	----------------------------------------------	--------------------------------------------	---------------------------------------------	--------------------------------------------	-------------------------------------------	--------------------------------------------	--------------------------------------------	------------------------------------------	-----------------------------------------	-----------------------------------------------	---------------------------------------------	-----------------------------------------------	----------------------------------------------	---------------------------------------------	-----------------------------------------------	-----------------------------------------	------------------------------------------	------------------------------------------	--------------------------------------------	----------------------------------------------	---------------------------------------------	---------------------------------------------	--------------------------------------------	-----------------------------------------------	--------------------------------------------	------------------------------------------	--------------------------------------------------	-------------------------------------------------	--------------------------------------------	---------------------------------------------	------------------------------------------	----------------------------------------------	----------------------------------------------	-----------------------------------------------	-----------------------------------------------	---------------------------------------------	---------------------------------------------	---------------------------------------------	--------------------------------------------------	------------------------------------------------------	----------------------------------------------------	------------------------------------------------------	----------------------------------------------------	---------------------------------------------------	--------------------------------------------------	-----------------------------------------------------	-----------------------------------------------------	------------------------------------------------------	------------------------------------------------------	----------------------------------------------	---------------------------------------------	---------------------------------------------------	---------------------------------------------------	-----------------------------------------------------------

# SOURCE CODE:

```

import numpy as np
import pandas as pd
import seaborn as sns
import category_encoders as ce
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import RobustScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
  
```

Name- Akash Kumar  
Internship Program- Data Science With Machine Learning and Python  
Batch- Jan 2022 – Mar 2022  
Certificate Code- TCRIB2R43  
Date of submission- 3 APRIL 2022

```
data = pd.read_csv(r'C:\Users\HP\Downloads\HR_Employee_Attrition-1\HR_Employee_Attrition-1.csv')
data.head()

data.tail()

data.shape

data.columns

data.isnull().sum()

data.nunique()

data.describe().T

data.dtypes.sort_index()

duplicate_data = data[data.duplicated()]
duplicate_data

data.describe()

categorical = [i for i in data.columns if data[i].dtype == 'O']
print(categorical)

data[categorical].head()

attrition_count = pd.DataFrame(data['Attrition'].value_counts())
attrition_count

f, ax = plt.subplots(figsize=(8,10))
ax = data['Attrition'].value_counts().plot.pie(explode=[0,0], autopct = '%1.1f%%', shadow=True)
ax.set_title('Attrition Probability')

fig_dims = (12, 7)
fig, ax = plt.subplots(figsize=fig_dims)

#ax = axis
sns.countplot(x='Age', hue='Attrition', data = data, palette="colorblind",
```

Name- Akash Kumar  
Internship Program- Data Science With Machine Learning and Python  
Batch- Jan 2022 – Mar 2022  
Certificate Code- TCRIB2R43  
Date of submission- 3 APRIL 2022

```
f, ax = plt.subplots(2,2, figsize=(20,15))
```

```
ax[0,0] = sns.countplot(x='Attrition', hue=  
'EducationField', data=data, ax = ax[0,0], palette='Set1' )  
ax[0,0].set_title("Frequency Distribution of Attrition w.r.t.  
Education Field")
```

```
ax[1,0] = sns.countplot(x='Attrition', hue= 'Department',  
data=data, ax = ax[1,0], palette='Set1' )  
ax[1,0].set_title("Frequency Distribution of Attrition w.r.t.  
Department")
```

```
ax[0,1] = sns.countplot(x='Attrition', hue= 'Education',  
data=data, ax = ax[0,1], palette='Set1' )  
ax[0,1].set_title("Frequency Distribution of Attrition w.r.t.  
Education")
```

```
ax[1,1] = sns.countplot(x='Attrition', hue=  
'BusinessTravel', data=data, ax = ax[1,1], palette='Set1' )  
ax[1,1].set_title("Frequency Distribution of Attrition w.r.t.  
Bussiness Travel")
```

```
f.tight_layout()
```

```
for column in data.columns:  
    if data[column].dtype == object:  
        print(str(column) + ' : ' + str(data[column].unique()))  
        print(data[column].value_counts())  
        print("-"*90)
```

```
#Remove unneeded columns
```

```
#Remove the column EmployeeNumber  
data = data.drop('EmployeeNumber', axis = 1) # A number  
assignment
```

```
#Remove the column StandardHours  
data = data.drop('StandardHours', axis = 1) #Contains only  
value 80
```

```
#Remove the column EmployeeCount  
data = data.drop('EmployeeCount', axis = 1) #Contains  
only the value 1
```

```
#Remove the column EmployeeCount  
data = data.drop('Over18', axis = 1) #Contains only the  
value 'Yes'
```

Name- Akash Kumar  
Internship Program- Data Science With Machine Learning and Python  
Batch- Jan 2022 – Mar 2022  
Certificate Code- TCRIB2R43  
Date of submission- 3 APRIL 2022

```
plt.figure(figsize=(15,15))
sns.heatmap(
    data.corr(),
    annot=True,
    fmt='.0%',
    linewidths=1,
    cmap='inferno'
)

X_train, X_test, Y_train, Y_test =
train_test_split(x, y,
test_size=0.2,random_state=42 )

print(X_train.shape, X_test.shape)

encoder = ce.OrdinalEncoder(cols =
['BusinessTravel', 'Department',
'EducationField', 'Gender', 'JobRole',
'MaritalStatus', 'OverTime'])
X_train = encoder.fit_transform(X_train)
X_test = encoder.fit_transform(X_test)

cols = X_train.columns

scaler = RobustScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.fit_transform(X_test)
X_train = pd.DataFrame(X_train, columns=
[cols])
X_test = pd.DataFrame(X_test,
columns=[cols])

rfc =
RandomForestClassifier(n_estimators=100,
random_state=0)
rfc.fit(X_train, Y_train)
y_pred = rfc.predict(X_test)
```

Name- Akash Kumar  
Internship Program- Data Science With Machine Learning and Python  
Batch- Jan 2022 – Mar 2022  
Certificate Code- TCRIB2R43  
Date of submission- 3 APRIL 2022

```
score = accuracy_score(Y_test, y_pred)
print('randomforest classifier score: ',
np.abs(score)*100)
```

**y\_pred**

**final result**

**Accuracy score = 95.23**