{

"cells": [

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"<h1>EXPLORATORY DATA ANALYSIS</h1>"

]

},

{

"cell\_type": "code",

"execution\_count": 1,

"metadata": {},

"outputs": [],

"source": [

"import numpy as np\n",

"import pandas as pd\n",

"\n",

"pd.set\_option('display.max\_rows', 2000)\n",

"pd.set\_option('display.max\_columns', 500)\n",

"pd.set\_option('display.width', 1000)\n",

"\n",

"\n",

"%matplotlib inline\n",

"import matplotlib.pyplot as plt\n",

"import seaborn as sns\n"

]

},

{

"cell\_type": "code",

"execution\_count": 2,

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Age</th>\n",

" <th>Attrition</th>\n",

" <th>BusinessTravel</th>\n",

" <th>DailyRate</th>\n",

" <th>Department</th>\n",

" <th>DistanceFromHome</th>\n",

" <th>Education</th>\n",

" <th>EducationField</th>\n",

" <th>EmployeeCount</th>\n",

" <th>EmployeeNumber</th>\n",

" <th>EnvironmentSatisfaction</th>\n",

" <th>Gender</th>\n",

" <th>HourlyRate</th>\n",

" <th>JobInvolvement</th>\n",

" <th>JobLevel</th>\n",

" <th>JobRole</th>\n",

" <th>JobSatisfaction</th>\n",

" <th>MaritalStatus</th>\n",

" <th>MonthlyIncome</th>\n",

" <th>MonthlyRate</th>\n",

" <th>NumCompaniesWorked</th>\n",

" <th>Over18</th>\n",

" <th>OverTime</th>\n",

" <th>PercentSalaryHike</th>\n",

" <th>PerformanceRating</th>\n",

" <th>RelationshipSatisfaction</th>\n",

" <th>StandardHours</th>\n",

" <th>StockOptionLevel</th>\n",

" <th>TotalWorkingYears</th>\n",

" <th>TrainingTimesLastYear</th>\n",

" <th>WorkLifeBalance</th>\n",

" <th>YearsAtCompany</th>\n",

" <th>YearsInCurrentRole</th>\n",

" <th>YearsSinceLastPromotion</th>\n",

" <th>YearsWithCurrManager</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>41</td>\n",

" <td>Yes</td>\n",

" <td>Travel\_Rarely</td>\n",

" <td>1102</td>\n",

" <td>Sales</td>\n",

" <td>1</td>\n",

" <td>2</td>\n",

" <td>Life Sciences</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>2</td>\n",

" <td>Female</td>\n",

" <td>94</td>\n",

" <td>3</td>\n",

" <td>2.0</td>\n",

" <td>Sales Executive</td>\n",

" <td>4.0</td>\n",

" <td>Single</td>\n",

" <td>5993.0</td>\n",

" <td>19479.0</td>\n",

" <td>8.0</td>\n",

" <td>Y</td>\n",

" <td>Yes</td>\n",

" <td>11.0</td>\n",

" <td>3.0</td>\n",

" <td>1.0</td>\n",

" <td>80.0</td>\n",

" <td>0.0</td>\n",

" <td>8.0</td>\n",

" <td>0.0</td>\n",

" <td>1.0</td>\n",

" <td>6.0</td>\n",

" <td>4.0</td>\n",

" <td>0.0</td>\n",

" <td>5.0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>49</td>\n",

" <td>No</td>\n",

" <td>Travel\_Frequently</td>\n",

" <td>279</td>\n",

" <td>Research &amp; Development</td>\n",

" <td>8</td>\n",

" <td>1</td>\n",

" <td>Life Sciences</td>\n",

" <td>1</td>\n",

" <td>2</td>\n",

" <td>3</td>\n",

" <td>Male</td>\n",

" <td>61</td>\n",

" <td>2</td>\n",

" <td>2.0</td>\n",

" <td>Research Scientist</td>\n",

" <td>2.0</td>\n",

" <td>Married</td>\n",

" <td>5130.0</td>\n",

" <td>24907.0</td>\n",

" <td>1.0</td>\n",

" <td>Y</td>\n",

" <td>No</td>\n",

" <td>23.0</td>\n",

" <td>4.0</td>\n",

" <td>4.0</td>\n",

" <td>80.0</td>\n",

" <td>1.0</td>\n",

" <td>10.0</td>\n",

" <td>3.0</td>\n",

" <td>3.0</td>\n",

" <td>10.0</td>\n",

" <td>7.0</td>\n",

" <td>1.0</td>\n",

" <td>7.0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>37</td>\n",

" <td>Yes</td>\n",

" <td>Travel\_Rarely</td>\n",

" <td>1373</td>\n",

" <td>Research &amp; Development</td>\n",

" <td>2</td>\n",

" <td>2</td>\n",

" <td>Other</td>\n",

" <td>1</td>\n",

" <td>4</td>\n",

" <td>4</td>\n",

" <td>Male</td>\n",

" <td>92</td>\n",

" <td>2</td>\n",

" <td>1.0</td>\n",

" <td>Laboratory Technician</td>\n",

" <td>3.0</td>\n",

" <td>Single</td>\n",

" <td>2090.0</td>\n",

" <td>2396.0</td>\n",

" <td>6.0</td>\n",

" <td>Y</td>\n",

" <td>Yes</td>\n",

" <td>15.0</td>\n",

" <td>3.0</td>\n",

" <td>2.0</td>\n",

" <td>80.0</td>\n",

" <td>0.0</td>\n",

" <td>7.0</td>\n",

" <td>3.0</td>\n",

" <td>3.0</td>\n",

" <td>0.0</td>\n",

" <td>0.0</td>\n",

" <td>0.0</td>\n",

" <td>0.0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>33</td>\n",

" <td>No</td>\n",

" <td>Travel\_Frequently</td>\n",

" <td>1392</td>\n",

" <td>Research &amp; Development</td>\n",

" <td>3</td>\n",

" <td>4</td>\n",

" <td>Life Sciences</td>\n",

" <td>1</td>\n",

" <td>5</td>\n",

" <td>4</td>\n",

" <td>Female</td>\n",

" <td>56</td>\n",

" <td>3</td>\n",

" <td>1.0</td>\n",

" <td>Research Scientist</td>\n",

" <td>3.0</td>\n",

" <td>Married</td>\n",

" <td>2909.0</td>\n",

" <td>23159.0</td>\n",

" <td>1.0</td>\n",

" <td>Y</td>\n",

" <td>Yes</td>\n",

" <td>11.0</td>\n",

" <td>3.0</td>\n",

" <td>3.0</td>\n",

" <td>80.0</td>\n",

" <td>0.0</td>\n",

" <td>8.0</td>\n",

" <td>3.0</td>\n",

" <td>3.0</td>\n",

" <td>8.0</td>\n",

" <td>7.0</td>\n",

" <td>3.0</td>\n",

" <td>0.0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>27</td>\n",

" <td>No</td>\n",

" <td>Travel\_Rarely</td>\n",

" <td>591</td>\n",

" <td>Research &amp; Development</td>\n",

" <td>2</td>\n",

" <td>1</td>\n",

" <td>Medical</td>\n",

" <td>1</td>\n",

" <td>7</td>\n",

" <td>1</td>\n",

" <td>Male</td>\n",

" <td>40</td>\n",

" <td>3</td>\n",

" <td>1.0</td>\n",

" <td>Laboratory Technician</td>\n",

" <td>2.0</td>\n",

" <td>Married</td>\n",

" <td>3468.0</td>\n",

" <td>16632.0</td>\n",

" <td>9.0</td>\n",

" <td>Y</td>\n",

" <td>No</td>\n",

" <td>12.0</td>\n",

" <td>3.0</td>\n",

" <td>4.0</td>\n",

" <td>80.0</td>\n",

" <td>1.0</td>\n",

" <td>6.0</td>\n",

" <td>3.0</td>\n",

" <td>3.0</td>\n",

" <td>2.0</td>\n",

" <td>2.0</td>\n",

" <td>2.0</td>\n",

" <td>2.0</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" Age Attrition BusinessTravel DailyRate Department DistanceFromHome Education EducationField EmployeeCount EmployeeNumber EnvironmentSatisfaction Gender HourlyRate JobInvolvement JobLevel JobRole JobSatisfaction MaritalStatus MonthlyIncome MonthlyRate NumCompaniesWorked Over18 OverTime PercentSalaryHike PerformanceRating RelationshipSatisfaction StandardHours StockOptionLevel TotalWorkingYears TrainingTimesLastYear WorkLifeBalance YearsAtCompany YearsInCurrentRole YearsSinceLastPromotion YearsWithCurrManager\n",

"0 41 Yes Travel\_Rarely 1102 Sales 1 2 Life Sciences 1 1 2 Female 94 3 2.0 Sales Executive 4.0 Single 5993.0 19479.0 8.0 Y Yes 11.0 3.0 1.0 80.0 0.0 8.0 0.0 1.0 6.0 4.0 0.0 5.0\n",

"1 49 No Travel\_Frequently 279 Research & Development 8 1 Life Sciences 1 2 3 Male 61 2 2.0 Research Scientist 2.0 Married 5130.0 24907.0 1.0 Y No 23.0 4.0 4.0 80.0 1.0 10.0 3.0 3.0 10.0 7.0 1.0 7.0\n",

"2 37 Yes Travel\_Rarely 1373 Research & Development 2 2 Other 1 4 4 Male 92 2 1.0 Laboratory Technician 3.0 Single 2090.0 2396.0 6.0 Y Yes 15.0 3.0 2.0 80.0 0.0 7.0 3.0 3.0 0.0 0.0 0.0 0.0\n",

"3 33 No Travel\_Frequently 1392 Research & Development 3 4 Life Sciences 1 5 4 Female 56 3 1.0 Research Scientist 3.0 Married 2909.0 23159.0 1.0 Y Yes 11.0 3.0 3.0 80.0 0.0 8.0 3.0 3.0 8.0 7.0 3.0 0.0\n",

"4 27 No Travel\_Rarely 591 Research & Development 2 1 Medical 1 7 1 Male 40 3 1.0 Laboratory Technician 2.0 Married 3468.0 16632.0 9.0 Y No 12.0 3.0 4.0 80.0 1.0 6.0 3.0 3.0 2.0 2.0 2.0 2.0"

]

},

"execution\_count": 2,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"data = pd.read\_csv(r\"C:\\Users\\itspark\\Documents\\Analytics\\dataset/hr\_employee\_retention.csv\")\n",

"data.head()"

]

},

{

"cell\_type": "code",

"execution\_count": 3,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"(1469, 35)"

]

},

"execution\_count": 3,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"data.shape"

]

},

{

"cell\_type": "code",

"execution\_count": 4,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"No 1232\n",

"Yes 237\n",

"Name: Attrition, dtype: int64"

]

},

"execution\_count": 4,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"data.Attrition.value\_counts()"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"<h2>DISTRIBUTION OF EMPLOYEE ATTRITION IN THE COMPANY</h2> \n"

]

},

{

"cell\_type": "code",

"execution\_count": 5,

"metadata": {},

"outputs": [

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 1080x432 with 1 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"labels = 'Attrition NO','Attrition YES'\n",

"data['Attrition'].astype(str).value\_counts().plot(kind='pie',\n",

" figsize=(15, 6),\n",

" autopct='%1.1f%%', \n",

" startangle=90, \n",

" shadow=True, \n",

" labels=None, \n",

" )\n",

"\n",

"plt.title('Distribution of Employee Attrition in the Company ', y=1.12) \n",

"plt.axis('equal') \n",

"# add legend\n",

"plt.legend(labels=labels, loc='upper left') \n",

" # show plot\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\*\*From the Pie Chart, we can infer that out of 1470 employees, 16% of the employees left their job due to some reasons whereas other 84% of the employees preferred to continue their job at the company\*\* "

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"<h2>ANALYSIS OF THE FEATURE NAMES</h2> "

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\*\*Removing Useless Features\*\*\n",

"\n",

"1. \*StandardHours\*\n",

"1. \*EmployeeCount\*\n",

"1. \*Over18\*\n",

"1. \*EmployeeNumber\*\n",

"1. \*StockOptionLevel\*"

]

},

{

"cell\_type": "code",

"execution\_count": 6,

"metadata": {

"scrolled": true

},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Age</th>\n",

" <th>Attrition</th>\n",

" <th>BusinessTravel</th>\n",

" <th>DailyRate</th>\n",

" <th>Department</th>\n",

" <th>DistanceFromHome</th>\n",

" <th>Education</th>\n",

" <th>EducationField</th>\n",

" <th>EnvironmentSatisfaction</th>\n",

" <th>Gender</th>\n",

" <th>HourlyRate</th>\n",

" <th>JobInvolvement</th>\n",

" <th>JobLevel</th>\n",

" <th>JobRole</th>\n",

" <th>JobSatisfaction</th>\n",

" <th>MaritalStatus</th>\n",

" <th>MonthlyIncome</th>\n",

" <th>MonthlyRate</th>\n",

" <th>NumCompaniesWorked</th>\n",

" <th>OverTime</th>\n",

" <th>PercentSalaryHike</th>\n",

" <th>PerformanceRating</th>\n",

" <th>RelationshipSatisfaction</th>\n",

" <th>TotalWorkingYears</th>\n",

" <th>TrainingTimesLastYear</th>\n",

" <th>WorkLifeBalance</th>\n",

" <th>YearsAtCompany</th>\n",

" <th>YearsInCurrentRole</th>\n",

" <th>YearsSinceLastPromotion</th>\n",

" <th>YearsWithCurrManager</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>41</td>\n",

" <td>Yes</td>\n",

" <td>Travel\_Rarely</td>\n",

" <td>1102</td>\n",

" <td>Sales</td>\n",

" <td>1</td>\n",

" <td>2</td>\n",

" <td>Life Sciences</td>\n",

" <td>2</td>\n",

" <td>Female</td>\n",

" <td>94</td>\n",

" <td>3</td>\n",

" <td>2.0</td>\n",

" <td>Sales Executive</td>\n",

" <td>4.0</td>\n",

" <td>Single</td>\n",

" <td>5993.0</td>\n",

" <td>19479.0</td>\n",

" <td>8.0</td>\n",

" <td>Yes</td>\n",

" <td>11.0</td>\n",

" <td>3.0</td>\n",

" <td>1.0</td>\n",

" <td>8.0</td>\n",

" <td>0.0</td>\n",

" <td>1.0</td>\n",

" <td>6.0</td>\n",

" <td>4.0</td>\n",

" <td>0.0</td>\n",

" <td>5.0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>49</td>\n",

" <td>No</td>\n",

" <td>Travel\_Frequently</td>\n",

" <td>279</td>\n",

" <td>Research &amp; Development</td>\n",

" <td>8</td>\n",

" <td>1</td>\n",

" <td>Life Sciences</td>\n",

" <td>3</td>\n",

" <td>Male</td>\n",

" <td>61</td>\n",

" <td>2</td>\n",

" <td>2.0</td>\n",

" <td>Research Scientist</td>\n",

" <td>2.0</td>\n",

" <td>Married</td>\n",

" <td>5130.0</td>\n",

" <td>24907.0</td>\n",

" <td>1.0</td>\n",

" <td>No</td>\n",

" <td>23.0</td>\n",

" <td>4.0</td>\n",

" <td>4.0</td>\n",

" <td>10.0</td>\n",

" <td>3.0</td>\n",

" <td>3.0</td>\n",

" <td>10.0</td>\n",

" <td>7.0</td>\n",

" <td>1.0</td>\n",

" <td>7.0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>37</td>\n",

" <td>Yes</td>\n",

" <td>Travel\_Rarely</td>\n",

" <td>1373</td>\n",

" <td>Research &amp; Development</td>\n",

" <td>2</td>\n",

" <td>2</td>\n",

" <td>Other</td>\n",

" <td>4</td>\n",

" <td>Male</td>\n",

" <td>92</td>\n",

" <td>2</td>\n",

" <td>1.0</td>\n",

" <td>Laboratory Technician</td>\n",

" <td>3.0</td>\n",

" <td>Single</td>\n",

" <td>2090.0</td>\n",

" <td>2396.0</td>\n",

" <td>6.0</td>\n",

" <td>Yes</td>\n",

" <td>15.0</td>\n",

" <td>3.0</td>\n",

" <td>2.0</td>\n",

" <td>7.0</td>\n",

" <td>3.0</td>\n",

" <td>3.0</td>\n",

" <td>0.0</td>\n",

" <td>0.0</td>\n",

" <td>0.0</td>\n",

" <td>0.0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>33</td>\n",

" <td>No</td>\n",

" <td>Travel\_Frequently</td>\n",

" <td>1392</td>\n",

" <td>Research &amp; Development</td>\n",

" <td>3</td>\n",

" <td>4</td>\n",

" <td>Life Sciences</td>\n",

" <td>4</td>\n",

" <td>Female</td>\n",

" <td>56</td>\n",

" <td>3</td>\n",

" <td>1.0</td>\n",

" <td>Research Scientist</td>\n",

" <td>3.0</td>\n",

" <td>Married</td>\n",

" <td>2909.0</td>\n",

" <td>23159.0</td>\n",

" <td>1.0</td>\n",

" <td>Yes</td>\n",

" <td>11.0</td>\n",

" <td>3.0</td>\n",

" <td>3.0</td>\n",

" <td>8.0</td>\n",

" <td>3.0</td>\n",

" <td>3.0</td>\n",

" <td>8.0</td>\n",

" <td>7.0</td>\n",

" <td>3.0</td>\n",

" <td>0.0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>27</td>\n",

" <td>No</td>\n",

" <td>Travel\_Rarely</td>\n",

" <td>591</td>\n",

" <td>Research &amp; Development</td>\n",

" <td>2</td>\n",

" <td>1</td>\n",

" <td>Medical</td>\n",

" <td>1</td>\n",

" <td>Male</td>\n",

" <td>40</td>\n",

" <td>3</td>\n",

" <td>1.0</td>\n",

" <td>Laboratory Technician</td>\n",

" <td>2.0</td>\n",

" <td>Married</td>\n",

" <td>3468.0</td>\n",

" <td>16632.0</td>\n",

" <td>9.0</td>\n",

" <td>No</td>\n",

" <td>12.0</td>\n",

" <td>3.0</td>\n",

" <td>4.0</td>\n",

" <td>6.0</td>\n",

" <td>3.0</td>\n",

" <td>3.0</td>\n",

" <td>2.0</td>\n",

" <td>2.0</td>\n",

" <td>2.0</td>\n",

" <td>2.0</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" Age Attrition BusinessTravel DailyRate Department DistanceFromHome Education EducationField EnvironmentSatisfaction Gender HourlyRate JobInvolvement JobLevel JobRole JobSatisfaction MaritalStatus MonthlyIncome MonthlyRate NumCompaniesWorked OverTime PercentSalaryHike PerformanceRating RelationshipSatisfaction TotalWorkingYears TrainingTimesLastYear WorkLifeBalance YearsAtCompany YearsInCurrentRole YearsSinceLastPromotion YearsWithCurrManager\n",

"0 41 Yes Travel\_Rarely 1102 Sales 1 2 Life Sciences 2 Female 94 3 2.0 Sales Executive 4.0 Single 5993.0 19479.0 8.0 Yes 11.0 3.0 1.0 8.0 0.0 1.0 6.0 4.0 0.0 5.0\n",

"1 49 No Travel\_Frequently 279 Research & Development 8 1 Life Sciences 3 Male 61 2 2.0 Research Scientist 2.0 Married 5130.0 24907.0 1.0 No 23.0 4.0 4.0 10.0 3.0 3.0 10.0 7.0 1.0 7.0\n",

"2 37 Yes Travel\_Rarely 1373 Research & Development 2 2 Other 4 Male 92 2 1.0 Laboratory Technician 3.0 Single 2090.0 2396.0 6.0 Yes 15.0 3.0 2.0 7.0 3.0 3.0 0.0 0.0 0.0 0.0\n",

"3 33 No Travel\_Frequently 1392 Research & Development 3 4 Life Sciences 4 Female 56 3 1.0 Research Scientist 3.0 Married 2909.0 23159.0 1.0 Yes 11.0 3.0 3.0 8.0 3.0 3.0 8.0 7.0 3.0 0.0\n",

"4 27 No Travel\_Rarely 591 Research & Development 2 1 Medical 1 Male 40 3 1.0 Laboratory Technician 2.0 Married 3468.0 16632.0 9.0 No 12.0 3.0 4.0 6.0 3.0 3.0 2.0 2.0 2.0 2.0"

]

},

"execution\_count": 6,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"# Drop useless features - StandardHours,EmployeeCount,Over18,EmployeeNumber,StockOptionLevel\n",

"data.drop(columns=['StandardHours','EmployeeCount','Over18','EmployeeNumber','StockOptionLevel'],inplace=True)\n",

"data.head()"

]

},

{

"cell\_type": "code",

"execution\_count": 7,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"(1469, 30)"

]

},

"execution\_count": 7,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"data.shape"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\*\*Analysis of the Rating Features\*\*\n",

"- JobSatisfaction\n",

"- EnvironmentSatisfaction\n",

"- RelationshipSatisfaction\n",

"- JobInvolvement\n",

"- WorkLifeBalance\n",

"- PerformanceRating"

]

},

{

"cell\_type": "code",

"execution\_count": 8,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"4.0 459\n",

"3.0 441\n",

"1.0 289\n",

"2.0 280\n",

"Name: JobSatisfaction, dtype: int64"

]

},

"execution\_count": 8,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"data['JobSatisfaction'].value\_counts()"

]

},

{

"cell\_type": "code",

"execution\_count": 9,

"metadata": {},

"outputs": [

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 1080x432 with 4 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"fig = plt.figure() \n",

"\n",

"ax1 = fig.add\_subplot(221) \n",

"ax2 = fig.add\_subplot(222) \n",

"ax3 = fig.add\_subplot(223) \n",

"ax4 = fig.add\_subplot(224) \n",

"\n",

"labels = 'Low','Medium','High','Very High'\n",

"\n",

"data['JobSatisfaction'].astype(str).value\_counts().plot(kind='pie',\n",

" figsize=(15, 6),\n",

" autopct='%1.1f%%', \n",

" startangle=90, \n",

" shadow=True, \n",

" labels=None,ax=ax1) # add to subplot 2\n",

"ax1.set\_title ('Rating of Job Satisfaction by Employees')\n",

"fig.legend(labels=labels,loc='center')\n",

"\n",

"data['EnvironmentSatisfaction'].astype(str).value\_counts().plot(kind='pie',\n",

" figsize=(15, 6),\n",

" autopct='%1.1f%%', \n",

" startangle=90, \n",

" shadow=True, \n",

" labels=None,ax=ax2) \n",

"ax2.set\_title('Rating of Environmental Satisfaction by Employees')\n",

"\n",

"data['RelationshipSatisfaction'].astype(str).value\_counts().plot(kind='pie',\n",

" figsize=(15, 6),\n",

" autopct='%1.1f%%', \n",

" startangle=90, \n",

" shadow=True, \n",

" labels=None,ax=ax3)\n",

"ax3.set\_title('Rating of Relationship Satisfaction by Employees')\n",

"\n",

"data['JobInvolvement'].astype(str).value\_counts().plot(kind='pie',\n",

" figsize=(15, 6),\n",

" autopct='%1.1f%%', \n",

" startangle=90, \n",

" shadow=True, \n",

" labels=None,ax=ax4) \n",

"ax4.set\_title('Rating of Job Involvement by Employees')\n",

"\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\*\*From the subplot, we can infer that more than 60% of the employees are :\*\*\n",

"- Not Satisfied in their Job\n",

"- Not Satisfied with their Work Environmnet\n",

"- Not Satisfied in their Relationship\n",

"- Not Getting involved in their job"

]

},

{

"cell\_type": "code",

"execution\_count": 10,

"metadata": {},

"outputs": [

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 1080x432 with 2 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"fig2 = plt.figure() \n",

"\n",

"ax5 = fig2.add\_subplot(121) \n",

"ax6 = fig2.add\_subplot(122) \n",

" \n",

"labels\_list1 = 'Bad','Good','Better','Best' \n",

"labels\_list2 = 'Low','Good','Excellent','Outstanding'\n",

"\n",

"data['WorkLifeBalance'].astype(str).value\_counts().plot(kind='pie',\n",

" figsize=(15, 6),\n",

" autopct='%1.1f%%', \n",

" startangle=90, \n",

" shadow=True, \n",

" labels=None,ax=ax5) # add to subplot 2\n",

"ax5.set\_title ('Rating of Work-Life Balance by Employees')\n",

"ax5.legend(labels=labels\_list1,loc='upper right')\n",

"\n",

"data['PerformanceRating'].astype(str).value\_counts().plot(kind='pie',\n",

" figsize=(15, 6),\n",

" autopct='%1.1f%%', \n",

" startangle=90, \n",

" shadow=True, \n",

" labels=None,ax=ax6) \n",

"ax6.set\_title('Performance Rating of the Employees')\n",

"ax6.legend(labels=labels\_list2,loc='upper right')\n",

"\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\*\*From the above piecharts, we can see that:\*\*\n",

"- Almost 60% of the employees have rated their Work-life Balance as Bad \n",

"- Almost 85% of the employees have a low performance rating "

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# Analysis of Business Travel Feature"

]

},

{

"cell\_type": "code",

"execution\_count": 11,

"metadata": {},

"outputs": [

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"props = data.groupby(\"BusinessTravel\")['Attrition'].value\_counts(normalize=False).unstack()\n",

"\n",

"props.plot(kind='bar', alpha=1, stacked='False')\n",

"\n",

"plt.title('Business Travel VS Attrition')\n",

"plt.ylabel('Number of Employee')\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\*\*From the above data it is clear that Employees who travel rarely have more attrition rate followed by Employees who travel frequently\*\*\n",

" \n",

"- Best way to reduce this attrition is to conduct monthly survey and to assign travel according to the Employees' business travel interest"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# Analysis of Work Experience "

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"- YearsAtCompany\n",

"- YearsInCurrentRole\n",

"- YearsSinceLastPromotion\n",

"- YearsWithCurrManager\n",

"- TotalWorkingYears"

]

},

{

"cell\_type": "code",

"execution\_count": 12,

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>YearsAtCompany</th>\n",

" <th>YearsInCurrentRole</th>\n",

" <th>YearsSinceLastPromotion</th>\n",

" <th>YearsWithCurrManager</th>\n",

" <th>TotalWorkingYears</th>\n",

" <th>Attrition</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>6.0</td>\n",

" <td>4.0</td>\n",

" <td>0.0</td>\n",

" <td>5.0</td>\n",

" <td>8.0</td>\n",

" <td>Yes</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>10.0</td>\n",

" <td>7.0</td>\n",

" <td>1.0</td>\n",

" <td>7.0</td>\n",

" <td>10.0</td>\n",

" <td>No</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>0.0</td>\n",

" <td>0.0</td>\n",

" <td>0.0</td>\n",

" <td>0.0</td>\n",

" <td>7.0</td>\n",

" <td>Yes</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>8.0</td>\n",

" <td>7.0</td>\n",

" <td>3.0</td>\n",

" <td>0.0</td>\n",

" <td>8.0</td>\n",

" <td>No</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>2.0</td>\n",

" <td>2.0</td>\n",

" <td>2.0</td>\n",

" <td>2.0</td>\n",

" <td>6.0</td>\n",

" <td>No</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" YearsAtCompany YearsInCurrentRole YearsSinceLastPromotion YearsWithCurrManager TotalWorkingYears Attrition\n",

"0 6.0 4.0 0.0 5.0 8.0 Yes\n",

"1 10.0 7.0 1.0 7.0 10.0 No\n",

"2 0.0 0.0 0.0 0.0 7.0 Yes\n",

"3 8.0 7.0 3.0 0.0 8.0 No\n",

"4 2.0 2.0 2.0 2.0 6.0 No"

]

},

"execution\_count": 12,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"we = data[['YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion', 'YearsWithCurrManager', 'TotalWorkingYears', 'Attrition']]\n",

"we.head()"

]

},

{

"cell\_type": "code",

"execution\_count": 13,

"metadata": {},

"outputs": [

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 720x432 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"yac = data.groupby(\"YearsAtCompany\")['Attrition'].value\_counts(normalize=False).unstack()\n",

"\n",

"yac.plot(kind='bar', stacked='False',figsize=(10,6))\n",

"\n",

"plt.title('Years At Company of Employee')\n",

"plt.ylabel('Number of Employees')\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\*\*It is observed that the newly arriving employees quit their jobs most,so more concern should be given to the freshers and their cause of leaving the company should be figured out\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": 14,

"metadata": {},

"outputs": [

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 1440x432 with 2 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"ycr = data.groupby(\"YearsInCurrentRole\")['Attrition'].value\_counts(normalize=False).unstack()\n",

"ysp = data.groupby(\"YearsSinceLastPromotion\")['Attrition'].value\_counts(normalize=False).unstack()\n",

"\n",

"\n",

"fig = plt.figure() # create figure\n",

"\n",

"ax0 = fig.add\_subplot(121) # add subplot 1 (1 row, 2 columns, first plot)\n",

"ax1 = fig.add\_subplot(122) # add subplot 2 (1 row, 2 columns, second plot). See tip below\*\*\n",

"\n",

"# Subplot 1: Box plot\n",

"ycr.plot(kind='bar', stacked='False',figsize=(20,6), ax=ax0) # add to subplot 1\n",

"ax0.set\_title('Same Role')\n",

"ax0.set\_xlabel('Years In Current Role')\n",

"ax0.set\_ylabel('Number of Employees')\n",

"\n",

"# Subplot 2: Line plot\n",

"ysp.plot(kind='bar', stacked='False',figsize=(20,6), ax=ax1) # add to subplot 2\n",

"ax1.set\_title ('Last Promotion')\n",

"ax1.set\_ylabel('Number of Employees')\n",

"ax1.set\_xlabel('Years Since Last Promotion')\n",

"\n",

"plt.show()\n"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\*\*From the above two plots, it is very clear that Employees who are in same post or not getting promoted tend to leave the company most. It is a major concern, since experienced Employees quiting their jobs would affect the company most\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": 15,

"metadata": {},

"outputs": [

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 720x432 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"ycm = data.groupby(\"YearsWithCurrManager\")['Attrition'].value\_counts(normalize=False).unstack()\n",

"\n",

"ycm.plot(kind='bar', stacked='False',figsize=(10,6))\n",

"\n",

"plt.title('Years with Current Manager')\n",

"plt.ylabel('Number of Employee')\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\*\*It is clear that in the starting of relation of Manager and Employee's are not so happy. It is important that the Manager communication with the employee from the starting itself trying to understand them soon to reduce the increase in Attrition\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": 16,

"metadata": {},

"outputs": [

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 576x360 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"twy = data.groupby(\"TotalWorkingYears\")['Attrition'].value\_counts(normalize=False).unstack()\n",

"\n",

"twy.plot(kind='bar', stacked='False',figsize=(8,5))\n",

"\n",

"plt.title('Total Working Years of Experience')\n",

"plt.ylabel('Number of Employee')\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\*\*It is observed that freshers leave the company very likely so it's important that company creates a new policy to handle freshers so they don't leave the company from the start\*\*"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# Analysis of Monthly Income"

]

},

{

"cell\_type": "code",

"execution\_count": 17,

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Yes</th>\n",

" <th>No</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>5993.0</td>\n",

" <td>5130.0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>2090.0</td>\n",

" <td>2909.0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>2028.0</td>\n",

" <td>3468.0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>3407.0</td>\n",

" <td>3068.0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>2960.0</td>\n",

" <td>2670.0</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" Yes No\n",

"0 5993.0 5130.0\n",

"1 2090.0 2909.0\n",

"2 2028.0 3468.0\n",

"3 3407.0 3068.0\n",

"4 2960.0 2670.0"

]

},

"execution\_count": 17,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"mi = data[data['Attrition']=='Yes']['MonthlyIncome']\n",

"mi = mi.reset\_index()\n",

"mi.drop(['index'], axis=1, inplace=True)\n",

"\n",

"\n",

"mn = data[data['Attrition']=='No']['MonthlyIncome']\n",

"mn = mn.reset\_index()\n",

"mn.drop(['index'], axis=1, inplace=True)\n",

"\n",

"mi['mn'] = mn\n",

"mi.rename(columns={'MonthlyIncome':'Yes', 'mn':'No'}, inplace=True)\n",

"mi.head()\n"

]

},

{

"cell\_type": "code",

"execution\_count": 18,

"metadata": {},

"outputs": [

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 720x504 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"mi.plot(kind='box', figsize=(10, 7))\n",

"\n",

"plt.title('Box plot of Monthly Income vs Attrition')\n",

"plt.ylabel('Monthly Income')\n",

"\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\*\*Employees who left their jobs tend to have low average monthly income than those who continued their job in the company\*\* "

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# Over Time Employee Analysis"

]

},

{

"cell\_type": "code",

"execution\_count": 19,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"<AxesSubplot:ylabel='Frequency'>"

]

},

"execution\_count": 19,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"dot = data[['OverTime', 'MonthlyIncome', 'Attrition']]\n",

"oyay = dot[(data['OverTime']=='Yes') & (data['Attrition']=='Yes')]\n",

"oyay = oyay.sort\_values(by = 'MonthlyIncome', ascending=False, axis=0) #sorting to get the top values\n",

"count, bin\_edges = np.histogram(oyay['MonthlyIncome'])\n",

"\n",

"oyay.plot(kind='hist', xticks=bin\_edges)"

]

},

{

"cell\_type": "code",

"execution\_count": 20,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"<AxesSubplot:ylabel='Frequency'>"

]

},

"execution\_count": 20,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"oyan = dot[(data['OverTime']=='Yes') & (data['Attrition']=='No')]\n",

"count, bin\_edges = np.histogram(oyan['MonthlyIncome'])\n",

"\n",

"oyan.plot(kind='hist', xticks=bin\_edges)"

]

},

{

"cell\_type": "code",

"execution\_count": 21,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"<AxesSubplot:ylabel='Frequency'>"

]

},

"execution\_count": 21,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"onay = dot[(data['OverTime']=='No') & (data['Attrition']=='Yes')]\n",

"count, bin\_edges = np.histogram(onay['MonthlyIncome'])\n",

"\n",

"onay.plot(kind='hist', xticks=bin\_edges)"

]

},

{

"cell\_type": "code",

"execution\_count": 22,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"<AxesSubplot:ylabel='Frequency'>"

]

},

"execution\_count": 22,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"onan = dot[(data['OverTime']=='No') & (data['Attrition']=='No')]\n",

"count, bin\_edges = np.histogram(onan['MonthlyIncome'])\n",

"\n",

"onan.plot(kind='hist',alpha =0.4, xticks=bin\_edges)"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# Analysis on Department"

]

},

{

"cell\_type": "code",

"execution\_count": 23,

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Department</th>\n",

" <th>Attrition</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>Sales</td>\n",

" <td>Yes</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>Research &amp; Development</td>\n",

" <td>No</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>Research &amp; Development</td>\n",

" <td>Yes</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>Research &amp; Development</td>\n",

" <td>No</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>Research &amp; Development</td>\n",

" <td>No</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" Department Attrition\n",

"0 Sales Yes\n",

"1 Research & Development No\n",

"2 Research & Development Yes\n",

"3 Research & Development No\n",

"4 Research & Development No"

]

},

"execution\_count": 23,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dpt = data[['Department','Attrition']]\n",

"dpt.head()"

]

},

{

"cell\_type": "code",

"execution\_count": 24,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"Research & Development 960\n",

"Sales 446\n",

"Human Resources 63\n",

"Name: Department, dtype: int64"

]

},

"execution\_count": 24,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dpt['Department'].value\_counts()"

]

},

{

"cell\_type": "code",

"execution\_count": 25,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"<matplotlib.legend.Legend at 0x29b069e4100>"

]

},

"execution\_count": 25,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 1080x432 with 1 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"dpt['Department'].value\_counts().plot(kind='pie',\n",

" figsize=(15, 6),\n",

" autopct='%1.1f%%', \n",

" startangle=90, \n",

" shadow=True, \n",

" labels=None) \n",

"plt.axis('equal') \n",

"plt.legend(labels=dpt['Department'].unique(), loc='upper left') "

]

},

{

"cell\_type": "code",

"execution\_count": 26,

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th>Department</th>\n",

" <th>Human Resources</th>\n",

" <th>Research &amp; Development</th>\n",

" <th>Sales</th>\n",

" </tr>\n",

" <tr>\n",

" <th>Attrition</th>\n",

" <th></th>\n",

" <th></th>\n",

" <th></th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>No</th>\n",

" <td>51</td>\n",

" <td>827</td>\n",

" <td>354</td>\n",

" </tr>\n",

" <tr>\n",

" <th>Yes</th>\n",

" <td>12</td>\n",

" <td>133</td>\n",

" <td>92</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

"Department Human Resources Research & Development Sales\n",

"Attrition \n",

"No 51 827 354\n",

"Yes 12 133 92"

]

},

"execution\_count": 26,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dpm = data.groupby(\"Department\")['Attrition'].value\_counts(normalize=False).unstack()\n",

"dpm = dpm.transpose()\n",

"dpm"

]

},

{

"cell\_type": "code",

"execution\_count": 27,

"metadata": {},

"outputs": [

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"labels = ['Human Resources', 'Research & Development', 'Sales',]\n",

"sizes = [63, 961, 446]\n",

"labels\_attrition = ['Yes','No','Yes','No','Yes','No']\n",

"sizes\_attrition = [12,51,133,828,92,354]\n",

"colors = ['#ff6666', '#ffcc99', '#99ff99']\n",

"\n",

"colors\_attrition = ['#0a0e77','#9e0723', '#0a0e77','#9e0723', '#0a0e77','#9e0723', '#0a0e77','#9e0723']\n",

" \n",

"# Plot\n",

"plt.pie(sizes, autopct='%1.1f%%', pctdistance=.87, labels=labels, colors=colors, startangle=90,frame=True)\n",

"plt.pie(sizes\_attrition,colors=colors\_attrition,radius=0.75,startangle=90)\n",

"centre\_circle = plt.Circle((0,0),0.5,color='black', fc='white',linewidth=0.5)\n",

"fig6 = plt.gcf()\n",

"fig6.gca().add\_artist(centre\_circle)\n",

"\n",

"#legend\n",

"import matplotlib.patches as mpatches\n",

"pur = mpatches.Patch(color='#0a0e77', label='Yes')\n",

"pin = mpatches.Patch(color='#9e0723', label='No')\n",

"plt.legend(handles=[pur, pin], loc='upper left')\n",

"\n",

"plt.axis('equal')\n",

"plt.tight\_layout()\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# Analysis Gender"

]

},

{

"cell\_type": "code",

"execution\_count": 28,

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Gender</th>\n",

" <th>DistanceFromHome</th>\n",

" <th>Attrition</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>Female</td>\n",

" <td>1</td>\n",

" <td>Yes</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>Male</td>\n",

" <td>8</td>\n",

" <td>No</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>Male</td>\n",

" <td>2</td>\n",

" <td>Yes</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>Female</td>\n",

" <td>3</td>\n",

" <td>No</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>Male</td>\n",

" <td>2</td>\n",

" <td>No</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" Gender DistanceFromHome Attrition\n",

"0 Female 1 Yes\n",

"1 Male 8 No\n",

"2 Male 2 Yes\n",

"3 Female 3 No\n",

"4 Male 2 No"

]

},

"execution\_count": 28,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"gda = data[['Gender', 'DistanceFromHome', 'Attrition']]\n",

"gda.head()"

]

},

{

"cell\_type": "code",

"execution\_count": 29,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"Male 881\n",

"Female 588\n",

"Name: Gender, dtype: int64"

]

},

"execution\_count": 29,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"gda['Gender'].value\_counts()"

]

},

{

"cell\_type": "code",

"execution\_count": 30,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"<matplotlib.legend.Legend at 0x29b069c9580>"

]

},

"execution\_count": 30,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 1080x432 with 1 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"gda['Gender'].value\_counts().plot(kind='pie',\n",

" figsize=(15, 6),\n",

" autopct='%1.1f%%', \n",

" startangle=90, \n",

" shadow=True, \n",

" labels=None) \n",

"plt.axis('equal') \n",

"plt.legend(labels=['Male', 'Female'], loc='upper left') "

]

},

{

"cell\_type": "code",

"execution\_count": 31,

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th>Gender</th>\n",

" <th>Female</th>\n",

" <th>Male</th>\n",

" </tr>\n",

" <tr>\n",

" <th>Attrition</th>\n",

" <th></th>\n",

" <th></th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>No</th>\n",

" <td>501</td>\n",

" <td>731</td>\n",

" </tr>\n",

" <tr>\n",

" <th>Yes</th>\n",

" <td>87</td>\n",

" <td>150</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

"Gender Female Male\n",

"Attrition \n",

"No 501 731\n",

"Yes 87 150"

]

},

"execution\_count": 31,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"fma = gda.groupby(\"Gender\")['Attrition'].value\_counts(normalize=False).unstack()\n",

"fma = fma.transpose()\n",

"fma"

]

},

{

"cell\_type": "code",

"execution\_count": 32,

"metadata": {},

"outputs": [

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"labels = ['Male', 'Female']\n",

"sizes = [882,588]\n",

"labels\_attrition = ['Yes','No','Yes','No']\n",

"sizes\_attrition = [150,732,87,501]\n",

"colors = ['#ff6666', '#ffcc99']\n",

"\n",

"colors\_attrition = ['#c2c2f0','#ffb3e6', '#c2c2f0','#ffb3e6']\n",

" \n",

"# Plot\n",

"plt.pie(sizes, labels=labels, colors=colors, startangle=90,frame=True)\n",

"plt.pie(sizes\_attrition,colors=colors\_attrition,radius=0.75,startangle=90)\n",

"centre\_circle = plt.Circle((0,0),0.5,color='black', fc='white',linewidth=0.5)\n",

"fig6 = plt.gcf()\n",

"fig6.gca().add\_artist(centre\_circle)\n",

"\n",

"#legend\n",

"import matplotlib.patches as mpatches\n",

"pur = mpatches.Patch(color='#c2c2f0', label='Yes')\n",

"pin = mpatches.Patch(color='#ffb3e6', label='No')\n",

"plt.legend(handles=[pur, pin], loc='center')\n",

"\n",

"plt.axis('equal')\n",

"plt.tight\_layout()\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# Analysis of Marital Status"

]

},

{

"cell\_type": "code",

"execution\_count": 33,

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>MaritalStatus</th>\n",

" <th>Attrition</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>Single</td>\n",

" <td>Yes</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>Married</td>\n",

" <td>No</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>Single</td>\n",

" <td>Yes</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>Married</td>\n",

" <td>No</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>Married</td>\n",

" <td>No</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" MaritalStatus Attrition\n",

"0 Single Yes\n",

"1 Married No\n",

"2 Single Yes\n",

"3 Married No\n",

"4 Married No"

]

},

"execution\_count": 33,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"ms = data[['MaritalStatus', 'Attrition']]\n",

"ms.head()"

]

},

{

"cell\_type": "code",

"execution\_count": 34,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"Married 672\n",

"Single 470\n",

"Divorced 327\n",

"Name: MaritalStatus, dtype: int64"

]

},

"execution\_count": 34,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"ms['MaritalStatus'].value\_counts()"

]

},

{

"cell\_type": "code",

"execution\_count": 35,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"<matplotlib.legend.Legend at 0x29b07248d30>"

]

},

"execution\_count": 35,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 1080x432 with 1 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"ms['MaritalStatus'].value\_counts().plot(kind='pie',\n",

" figsize=(15, 6),\n",

" autopct='%1.1f%%', \n",

" startangle=90, \n",

" shadow=True, \n",

" labels=None) \n",

"plt.axis('equal') \n",

"plt.legend(labels=['Married', 'Single', 'Divorced'], loc='upper left') "

]

},

{

"cell\_type": "code",

"execution\_count": 36,

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th>MaritalStatus</th>\n",

" <th>Divorced</th>\n",

" <th>Married</th>\n",

" <th>Single</th>\n",

" </tr>\n",

" <tr>\n",

" <th>Attrition</th>\n",

" <th></th>\n",

" <th></th>\n",

" <th></th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>No</th>\n",

" <td>294</td>\n",

" <td>588</td>\n",

" <td>350</td>\n",

" </tr>\n",

" <tr>\n",

" <th>Yes</th>\n",

" <td>33</td>\n",

" <td>84</td>\n",

" <td>120</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

"MaritalStatus Divorced Married Single\n",

"Attrition \n",

"No 294 588 350\n",

"Yes 33 84 120"

]

},

"execution\_count": 36,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"msa = ms.groupby(\"MaritalStatus\")['Attrition'].value\_counts(normalize=False).unstack()\n",

"msa = msa.transpose()\n",

"msa"

]

},

{

"cell\_type": "code",

"execution\_count": 37,

"metadata": {},

"outputs": [

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"labels = ['Married', 'Single', 'Divorced']\n",

"sizes = [673, 470, 327]\n",

"labels\_attrition = ['Yes','No','Yes','No','Yes','No']\n",

"sizes\_attrition = [84,589,120,350,33,294]\n",

"colors = ['#ff6666', '#ffcc99', '#99ff99']\n",

"\n",

"colors\_attrition = ['#c2c2f0','#ffb3e6', '#c2c2f0','#ffb3e6', '#c2c2f0','#ffb3e6']\n",

" \n",

"# Plot\n",

"plt.pie(sizes, labels=labels, colors=colors, startangle=90,frame=True)\n",

"plt.pie(sizes\_attrition,colors=colors\_attrition,radius=0.75,startangle=90)\n",

"centre\_circle = plt.Circle((0,0),0.5,color='black', fc='white',linewidth=0.5)\n",

"fig6 = plt.gcf()\n",

"fig6.gca().add\_artist(centre\_circle)\n",

"\n",

"#legend\n",

"import matplotlib.patches as mpatches\n",

"pur = mpatches.Patch(color='#c2c2f0', label='Yes')\n",

"pin = mpatches.Patch(color='#ffb3e6', label='No')\n",

"plt.legend(handles=[pur, pin], loc='center')\n",

"\n",

"plt.axis('equal')\n",

"plt.tight\_layout()\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# Model preprocessing"

]

},

{

"cell\_type": "code",

"execution\_count": 38,

"metadata": {},

"outputs": [],

"source": [

"from sklearn import preprocessing\n",

"import math\n",

"from sklearn.model\_selection import train\_test\_split\n",

"from sklearn import metrics "

]

},

{

"cell\_type": "code",

"execution\_count": 39,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"Index(['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department', 'DistanceFromHome', 'Education', 'EducationField', 'EnvironmentSatisfaction', 'Gender', 'HourlyRate', 'JobInvolvement', 'JobLevel', 'JobRole', 'JobSatisfaction', 'MaritalStatus', 'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked', 'OverTime', 'PercentSalaryHike', 'PerformanceRating', 'RelationshipSatisfaction', 'TotalWorkingYears', 'TrainingTimesLastYear', 'WorkLifeBalance', 'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion', 'YearsWithCurrManager'], dtype='object')"

]

},

"execution\_count": 39,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"data.columns"

]

},

{

"cell\_type": "code",

"execution\_count": 40,

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Age</th>\n",

" <th>Attrition</th>\n",

" <th>BusinessTravel</th>\n",

" <th>DailyRate</th>\n",

" <th>Department</th>\n",

" <th>DistanceFromHome</th>\n",

" <th>Education</th>\n",

" <th>EducationField</th>\n",

" <th>EnvironmentSatisfaction</th>\n",

" <th>Gender</th>\n",

" <th>HourlyRate</th>\n",

" <th>JobInvolvement</th>\n",

" <th>JobLevel</th>\n",

" <th>JobRole</th>\n",

" <th>JobSatisfaction</th>\n",

" <th>MaritalStatus</th>\n",

" <th>MonthlyIncome</th>\n",

" <th>MonthlyRate</th>\n",

" <th>NumCompaniesWorked</th>\n",

" <th>OverTime</th>\n",

" <th>PercentSalaryHike</th>\n",

" <th>PerformanceRating</th>\n",

" <th>RelationshipSatisfaction</th>\n",

" <th>TotalWorkingYears</th>\n",

" <th>TrainingTimesLastYear</th>\n",

" <th>WorkLifeBalance</th>\n",

" <th>YearsAtCompany</th>\n",

" <th>YearsInCurrentRole</th>\n",

" <th>YearsSinceLastPromotion</th>\n",

" <th>YearsWithCurrManager</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>41</td>\n",

" <td>Yes</td>\n",

" <td>Travel\_Rarely</td>\n",

" <td>1102</td>\n",

" <td>Sales</td>\n",

" <td>1</td>\n",

" <td>2</td>\n",

" <td>Life Sciences</td>\n",

" <td>2</td>\n",

" <td>Female</td>\n",

" <td>94</td>\n",

" <td>3</td>\n",

" <td>2.0</td>\n",

" <td>Sales Executive</td>\n",

" <td>4.0</td>\n",

" <td>Single</td>\n",

" <td>5993.0</td>\n",

" <td>19479.0</td>\n",

" <td>8.0</td>\n",

" <td>Yes</td>\n",

" <td>11.0</td>\n",

" <td>3.0</td>\n",

" <td>1.0</td>\n",

" <td>8.0</td>\n",

" <td>0.0</td>\n",

" <td>1.0</td>\n",

" <td>6.0</td>\n",

" <td>4.0</td>\n",

" <td>0.0</td>\n",

" <td>5.0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>49</td>\n",

" <td>No</td>\n",

" <td>Travel\_Frequently</td>\n",

" <td>279</td>\n",

" <td>Research &amp; Development</td>\n",

" <td>8</td>\n",

" <td>1</td>\n",

" <td>Life Sciences</td>\n",

" <td>3</td>\n",

" <td>Male</td>\n",

" <td>61</td>\n",

" <td>2</td>\n",

" <td>2.0</td>\n",

" <td>Research Scientist</td>\n",

" <td>2.0</td>\n",

" <td>Married</td>\n",

" <td>5130.0</td>\n",

" <td>24907.0</td>\n",

" <td>1.0</td>\n",

" <td>No</td>\n",

" <td>23.0</td>\n",

" <td>4.0</td>\n",

" <td>4.0</td>\n",

" <td>10.0</td>\n",

" <td>3.0</td>\n",

" <td>3.0</td>\n",

" <td>10.0</td>\n",

" <td>7.0</td>\n",

" <td>1.0</td>\n",

" <td>7.0</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" Age Attrition BusinessTravel DailyRate Department DistanceFromHome Education EducationField EnvironmentSatisfaction Gender HourlyRate JobInvolvement JobLevel JobRole JobSatisfaction MaritalStatus MonthlyIncome MonthlyRate NumCompaniesWorked OverTime PercentSalaryHike PerformanceRating RelationshipSatisfaction TotalWorkingYears TrainingTimesLastYear WorkLifeBalance YearsAtCompany YearsInCurrentRole YearsSinceLastPromotion YearsWithCurrManager\n",

"0 41 Yes Travel\_Rarely 1102 Sales 1 2 Life Sciences 2 Female 94 3 2.0 Sales Executive 4.0 Single 5993.0 19479.0 8.0 Yes 11.0 3.0 1.0 8.0 0.0 1.0 6.0 4.0 0.0 5.0\n",

"1 49 No Travel\_Frequently 279 Research & Development 8 1 Life Sciences 3 Male 61 2 2.0 Research Scientist 2.0 Married 5130.0 24907.0 1.0 No 23.0 4.0 4.0 10.0 3.0 3.0 10.0 7.0 1.0 7.0"

]

},

"execution\_count": 40,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"data.head(2)"

]

},

{

"cell\_type": "code",

"execution\_count": 42,

"metadata": {},

"outputs": [],

"source": [

"le = preprocessing.LabelEncoder()\n",

"categorial\_variables = ['Attrition','BusinessTravel','Department','EducationField',\n",

" 'Gender','JobRole','MaritalStatus','OverTime']"

]

},

{

"cell\_type": "code",

"execution\_count": 43,

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Age</th>\n",

" <th>Attrition</th>\n",

" <th>BusinessTravel</th>\n",

" <th>DailyRate</th>\n",

" <th>Department</th>\n",

" <th>DistanceFromHome</th>\n",

" <th>Education</th>\n",

" <th>EducationField</th>\n",

" <th>EnvironmentSatisfaction</th>\n",

" <th>Gender</th>\n",

" <th>HourlyRate</th>\n",

" <th>JobInvolvement</th>\n",

" <th>JobLevel</th>\n",

" <th>JobRole</th>\n",

" <th>JobSatisfaction</th>\n",

" <th>MaritalStatus</th>\n",

" <th>MonthlyIncome</th>\n",

" <th>MonthlyRate</th>\n",

" <th>NumCompaniesWorked</th>\n",

" <th>OverTime</th>\n",

" <th>PercentSalaryHike</th>\n",

" <th>PerformanceRating</th>\n",

" <th>RelationshipSatisfaction</th>\n",

" <th>TotalWorkingYears</th>\n",

" <th>TrainingTimesLastYear</th>\n",

" <th>WorkLifeBalance</th>\n",

" <th>YearsAtCompany</th>\n",

" <th>YearsInCurrentRole</th>\n",

" <th>YearsSinceLastPromotion</th>\n",

" <th>YearsWithCurrManager</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>41</td>\n",

" <td>1</td>\n",

" <td>2</td>\n",

" <td>1102</td>\n",

" <td>2</td>\n",

" <td>1</td>\n",

" <td>2</td>\n",

" <td>1</td>\n",

" <td>2</td>\n",

" <td>0</td>\n",

" <td>94</td>\n",

" <td>3</td>\n",

" <td>2.0</td>\n",

" <td>7</td>\n",

" <td>4.0</td>\n",

" <td>2</td>\n",

" <td>5993.0</td>\n",

" <td>19479.0</td>\n",

" <td>8.0</td>\n",

" <td>1</td>\n",

" <td>11.0</td>\n",

" <td>3.0</td>\n",

" <td>1.0</td>\n",

" <td>8.0</td>\n",

" <td>0.0</td>\n",

" <td>1.0</td>\n",

" <td>6.0</td>\n",

" <td>4.0</td>\n",

" <td>0.0</td>\n",

" <td>5.0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>49</td>\n",

" <td>0</td>\n",

" <td>1</td>\n",

" <td>279</td>\n",

" <td>1</td>\n",

" <td>8</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>3</td>\n",

" <td>1</td>\n",

" <td>61</td>\n",

" <td>2</td>\n",

" <td>2.0</td>\n",

" <td>6</td>\n",

" <td>2.0</td>\n",

" <td>1</td>\n",

" <td>5130.0</td>\n",

" <td>24907.0</td>\n",

" <td>1.0</td>\n",

" <td>0</td>\n",

" <td>23.0</td>\n",

" <td>4.0</td>\n",

" <td>4.0</td>\n",

" <td>10.0</td>\n",

" <td>3.0</td>\n",

" <td>3.0</td>\n",

" <td>10.0</td>\n",

" <td>7.0</td>\n",

" <td>1.0</td>\n",

" <td>7.0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>37</td>\n",

" <td>1</td>\n",

" <td>2</td>\n",

" <td>1373</td>\n",

" <td>1</td>\n",

" <td>2</td>\n",

" <td>2</td>\n",

" <td>4</td>\n",

" <td>4</td>\n",

" <td>1</td>\n",

" <td>92</td>\n",

" <td>2</td>\n",

" <td>1.0</td>\n",

" <td>2</td>\n",

" <td>3.0</td>\n",

" <td>2</td>\n",

" <td>2090.0</td>\n",

" <td>2396.0</td>\n",

" <td>6.0</td>\n",

" <td>1</td>\n",

" <td>15.0</td>\n",

" <td>3.0</td>\n",

" <td>2.0</td>\n",

" <td>7.0</td>\n",

" <td>3.0</td>\n",

" <td>3.0</td>\n",

" <td>0.0</td>\n",

" <td>0.0</td>\n",

" <td>0.0</td>\n",

" <td>0.0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>33</td>\n",

" <td>0</td>\n",

" <td>1</td>\n",

" <td>1392</td>\n",

" <td>1</td>\n",

" <td>3</td>\n",

" <td>4</td>\n",

" <td>1</td>\n",

" <td>4</td>\n",

" <td>0</td>\n",

" <td>56</td>\n",

" <td>3</td>\n",

" <td>1.0</td>\n",

" <td>6</td>\n",

" <td>3.0</td>\n",

" <td>1</td>\n",

" <td>2909.0</td>\n",

" <td>23159.0</td>\n",

" <td>1.0</td>\n",

" <td>1</td>\n",

" <td>11.0</td>\n",

" <td>3.0</td>\n",

" <td>3.0</td>\n",

" <td>8.0</td>\n",

" <td>3.0</td>\n",

" <td>3.0</td>\n",

" <td>8.0</td>\n",

" <td>7.0</td>\n",

" <td>3.0</td>\n",

" <td>0.0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>27</td>\n",

" <td>0</td>\n",

" <td>2</td>\n",

" <td>591</td>\n",

" <td>1</td>\n",

" <td>2</td>\n",

" <td>1</td>\n",

" <td>3</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>40</td>\n",

" <td>3</td>\n",

" <td>1.0</td>\n",

" <td>2</td>\n",

" <td>2.0</td>\n",

" <td>1</td>\n",

" <td>3468.0</td>\n",

" <td>16632.0</td>\n",

" <td>9.0</td>\n",

" <td>0</td>\n",

" <td>12.0</td>\n",

" <td>3.0</td>\n",

" <td>4.0</td>\n",

" <td>6.0</td>\n",

" <td>3.0</td>\n",

" <td>3.0</td>\n",

" <td>2.0</td>\n",

" <td>2.0</td>\n",

" <td>2.0</td>\n",

" <td>2.0</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" Age Attrition BusinessTravel DailyRate Department DistanceFromHome Education EducationField EnvironmentSatisfaction Gender HourlyRate JobInvolvement JobLevel JobRole JobSatisfaction MaritalStatus MonthlyIncome MonthlyRate NumCompaniesWorked OverTime PercentSalaryHike PerformanceRating RelationshipSatisfaction TotalWorkingYears TrainingTimesLastYear WorkLifeBalance YearsAtCompany YearsInCurrentRole YearsSinceLastPromotion YearsWithCurrManager\n",

"0 41 1 2 1102 2 1 2 1 2 0 94 3 2.0 7 4.0 2 5993.0 19479.0 8.0 1 11.0 3.0 1.0 8.0 0.0 1.0 6.0 4.0 0.0 5.0\n",

"1 49 0 1 279 1 8 1 1 3 1 61 2 2.0 6 2.0 1 5130.0 24907.0 1.0 0 23.0 4.0 4.0 10.0 3.0 3.0 10.0 7.0 1.0 7.0\n",

"2 37 1 2 1373 1 2 2 4 4 1 92 2 1.0 2 3.0 2 2090.0 2396.0 6.0 1 15.0 3.0 2.0 7.0 3.0 3.0 0.0 0.0 0.0 0.0\n",

"3 33 0 1 1392 1 3 4 1 4 0 56 3 1.0 6 3.0 1 2909.0 23159.0 1.0 1 11.0 3.0 3.0 8.0 3.0 3.0 8.0 7.0 3.0 0.0\n",

"4 27 0 2 591 1 2 1 3 1 1 40 3 1.0 2 2.0 1 3468.0 16632.0 9.0 0 12.0 3.0 4.0 6.0 3.0 3.0 2.0 2.0 2.0 2.0"

]

},

"execution\_count": 43,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"for i in categorial\_variables:\n",

" data[i] = le.fit\_transform(data[i])\n",

"data.head(5)"

]

},

{

"cell\_type": "code",

"execution\_count": 44,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"(1469, 29)"

]

},

"execution\_count": 44,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"target = data['Attrition']\n",

"train = data.drop('Attrition',axis = 1)\n",

"train.shape"

]

},

{

"cell\_type": "code",

"execution\_count": 45,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"0 1\n",

"Name: Attrition, dtype: int32"

]

},

"execution\_count": 45,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"target.head(1)"

]

},

{

"cell\_type": "code",

"execution\_count": 46,

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Age</th>\n",

" <th>BusinessTravel</th>\n",

" <th>DailyRate</th>\n",

" <th>Department</th>\n",

" <th>DistanceFromHome</th>\n",

" <th>Education</th>\n",

" <th>EducationField</th>\n",

" <th>EnvironmentSatisfaction</th>\n",

" <th>Gender</th>\n",

" <th>HourlyRate</th>\n",

" <th>JobInvolvement</th>\n",

" <th>JobLevel</th>\n",

" <th>JobRole</th>\n",

" <th>JobSatisfaction</th>\n",

" <th>MaritalStatus</th>\n",

" <th>MonthlyIncome</th>\n",

" <th>MonthlyRate</th>\n",

" <th>NumCompaniesWorked</th>\n",

" <th>OverTime</th>\n",

" <th>PercentSalaryHike</th>\n",

" <th>PerformanceRating</th>\n",

" <th>RelationshipSatisfaction</th>\n",

" <th>TotalWorkingYears</th>\n",

" <th>TrainingTimesLastYear</th>\n",

" <th>WorkLifeBalance</th>\n",

" <th>YearsAtCompany</th>\n",

" <th>YearsInCurrentRole</th>\n",

" <th>YearsSinceLastPromotion</th>\n",

" <th>YearsWithCurrManager</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>41</td>\n",

" <td>2</td>\n",

" <td>1102</td>\n",

" <td>2</td>\n",

" <td>1</td>\n",

" <td>2</td>\n",

" <td>1</td>\n",

" <td>2</td>\n",

" <td>0</td>\n",

" <td>94</td>\n",

" <td>3</td>\n",

" <td>2.0</td>\n",

" <td>7</td>\n",

" <td>4.0</td>\n",

" <td>2</td>\n",

" <td>5993.0</td>\n",

" <td>19479.0</td>\n",

" <td>8.0</td>\n",

" <td>1</td>\n",

" <td>11.0</td>\n",

" <td>3.0</td>\n",

" <td>1.0</td>\n",

" <td>8.0</td>\n",

" <td>0.0</td>\n",

" <td>1.0</td>\n",

" <td>6.0</td>\n",

" <td>4.0</td>\n",

" <td>0.0</td>\n",

" <td>5.0</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" Age BusinessTravel DailyRate Department DistanceFromHome Education EducationField EnvironmentSatisfaction Gender HourlyRate JobInvolvement JobLevel JobRole JobSatisfaction MaritalStatus MonthlyIncome MonthlyRate NumCompaniesWorked OverTime PercentSalaryHike PerformanceRating RelationshipSatisfaction TotalWorkingYears TrainingTimesLastYear WorkLifeBalance YearsAtCompany YearsInCurrentRole YearsSinceLastPromotion YearsWithCurrManager\n",

"0 41 2 1102 2 1 2 1 2 0 94 3 2.0 7 4.0 2 5993.0 19479.0 8.0 1 11.0 3.0 1.0 8.0 0.0 1.0 6.0 4.0 0.0 5.0"

]

},

"execution\_count": 46,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"train.head(1)"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# Now we will Implement classifiers in scikit-learn:\n",

"\n",

"Logistic Regression\n",

"\n",

"SVM\n",

"\n",

"KNN\n",

"\n",

"Decision Tree\n",

"\n",

"K Means Clustering"

]

},

{

"cell\_type": "code",

"execution\_count": 47,

"metadata": {},

"outputs": [],

"source": [

"train\_accuracy = []\n",

"test\_accuracy = []\n",

"models = ['Logistic Regression','SVM','KNN','Decision Tree','K Means Clustering']"

]

},

{

"cell\_type": "code",

"execution\_count": 48,

"metadata": {},

"outputs": [],

"source": [

"#Defining a function which will give us train and test accuracy for each classifier.\n",

"def train\_test\_error(y\_train,y\_test):\n",

" train\_error = ((y\_train==Y\_train).sum())/len(y\_train)\*100\n",

" test\_error = ((y\_test==Y\_test).sum())/len(Y\_test)\*100\n",

" train\_accuracy.append(train\_error)\n",

" test\_accuracy.append(test\_error)\n",

" print('{}'.format(train\_error) + \" is the train accuracy\")\n",

" print('{}'.format(test\_error) + \" is the test accuracy\")"

]

},

{

"cell\_type": "code",

"execution\_count": 49,

"metadata": {},

"outputs": [],

"source": [

"X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(train, target, test\_size=0.25, random\_state=42)"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# Logistic Regression"

]

},

{

"cell\_type": "code",

"execution\_count": 50,

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"83.10626702997274 is the train accuracy\n",

"86.68478260869566 is the test accuracy\n"

]

},

{

"name": "stderr",

"output\_type": "stream",

"text": [

"C:\\Users\\itspark\\anaconda3\\lib\\site-packages\\sklearn\\linear\_model\\\_logistic.py:814: ConvergenceWarning: lbfgs failed to converge (status=1):\n",

"STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.\n",

"\n",

"Increase the number of iterations (max\_iter) or scale the data as shown in:\n",

" https://scikit-learn.org/stable/modules/preprocessing.html\n",

"Please also refer to the documentation for alternative solver options:\n",

" https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression\n",

" n\_iter\_i = \_check\_optimize\_result(\n"

]

}

],

"source": [

"from sklearn.linear\_model import LogisticRegression\n",

"log\_reg = LogisticRegression()\n",

"log\_reg.fit(X\_train,Y\_train)\n",

"train\_predict = log\_reg.predict(X\_train)\n",

"test\_predict = log\_reg.predict(X\_test)\n",

"y\_prob = log\_reg.predict(train)\n",

"y\_pred = np.where(y\_prob > 0.5, 1, 0)\n",

"train\_test\_error(train\_predict , test\_predict)"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# SVM CLASSIFIER"

]

},

{

"cell\_type": "code",

"execution\_count": 51,

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"82.83378746594006 is the train accuracy\n",

"86.95652173913044 is the test accuracy\n"

]

}

],

"source": [

"from sklearn import svm\n",

"SVM = svm.SVC(probability=True)\n",

"SVM.fit(X\_train,Y\_train)\n",

"train\_predict = SVM.predict(X\_train)\n",

"test\_predict = SVM.predict(X\_test)\n",

"train\_test\_error(train\_predict , test\_predict)"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# KNN CLASSIFIER"

]

},

{

"cell\_type": "code",

"execution\_count": 52,

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"100.0 is the train accuracy\n",

"86.68478260869566 is the test accuracy\n"

]

}

],

"source": [

"from sklearn import neighbors\n",

"n\_neighbors = 15\n",

"knn = neighbors.KNeighborsClassifier(n\_neighbors, weights='distance')\n",

"knn.fit(X\_train,Y\_train)\n",

"train\_predict = knn.predict(X\_train)\n",

"test\_predict = knn.predict(X\_test)\n",

"train\_test\_error(train\_predict , test\_predict)"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# DICISION TREE CLASSIFIER"

]

},

{

"cell\_type": "code",

"execution\_count": 53,

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"100.0 is the train accuracy\n",

"79.34782608695652 is the test accuracy\n"

]

}

],

"source": [

"from sklearn import tree\n",

"dec = tree.DecisionTreeClassifier()\n",

"dec.fit(X\_train,Y\_train)\n",

"train\_predict = dec.predict(X\_train)\n",

"test\_predict = dec.predict(X\_test)\n",

"train\_test\_error(train\_predict , test\_predict)"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# K-MEANS CLUSTERING"

]

},

{

"cell\_type": "code",

"execution\_count": 54,

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"51.58946412352407 is the train accuracy\n",

"48.369565217391305 is the test accuracy\n"

]

}

],

"source": [

"from sklearn.cluster import KMeans\n",

"kms = KMeans(n\_clusters=2, random\_state=1)\n",

"kms.fit(X\_train,Y\_train)\n",

"train\_predict = kms.predict(X\_train)\n",

"test\_predict = kms.predict(X\_test)\n",

"train\_test\_error(train\_predict,test\_predict)"

]

},

{

"cell\_type": "code",

"execution\_count": 55,

"metadata": {},

"outputs": [],

"source": [

"results = pd.DataFrame({\"Test Accuracy\" : test\_accuracy , \"Train Accuracy\" : train\_accuracy} , index = models)"

]

},

{

"cell\_type": "code",

"execution\_count": 56,

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Test Accuracy</th>\n",

" <th>Train Accuracy</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>Logistic Regression</th>\n",

" <td>86.684783</td>\n",

" <td>83.106267</td>\n",

" </tr>\n",

" <tr>\n",

" <th>SVM</th>\n",

" <td>86.956522</td>\n",

" <td>82.833787</td>\n",

" </tr>\n",

" <tr>\n",

" <th>KNN</th>\n",

" <td>86.684783</td>\n",

" <td>100.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>Decision Tree</th>\n",

" <td>79.347826</td>\n",

" <td>100.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>K Means Clustering</th>\n",

" <td>48.369565</td>\n",

" <td>51.589464</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" Test Accuracy Train Accuracy\n",

"Logistic Regression 86.684783 83.106267\n",

"SVM 86.956522 82.833787\n",

"KNN 86.684783 100.000000\n",

"Decision Tree 79.347826 100.000000\n",

"K Means Clustering 48.369565 51.589464"

]

},

"execution\_count": 56,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"results"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

}

],

"metadata": {

"kernelspec": {

"display\_name": "Python 3 (ipykernel)",

"language": "python",

"name": "python3"

},

"language\_info": {

"codemirror\_mode": {

"name": "ipython",

"version": 3

},

"file\_extension": ".py",

"mimetype": "text/x-python",

"name": "python",

"nbconvert\_exporter": "python",

"pygments\_lexer": "ipython3",

"version": "3.9.12"

}

},

"nbformat": 4,

"nbformat\_minor": 2

}