Predicciones de salidas de colaboradores de RiotGames (Caso Hipotetico)

HR Employee Attrition Predicton on RiotGames Company (Hypothetical Case)

Contexto/Context

ESPAÑOL

Riot Games es una multinacional con miles de colaboradores repartidos por todo el mundo. La empresa anehela la contratación de los mejores talentos disponibles y en retenerlos el mayor tiempo posible. Por lo cual, se invierte una gran cantidad de recurso (y dinero) en retener a los empleados existentes a traves de diversas iniciativas. Sin embargo, la organización (gerente de operaciones) quiere reducir los costos de retener empleados. Para ello, nos proponenn limitar los incentivos unicamente a los empleados que corren riesgo de abandono.

Se le otorgo el desafío a la area de people de identificar patrones y caracteristicas de los empleados que abandonan la organización. Ademas, deben de utilizar la información levantada para predecir si un empleado esta en riesgo de abandono.

ENGLISH

Riot Games is a multinational company with thousands of employees around the world. The company strives to recruit the best talent available and to retain them for as long as possible. Therefore, a great deal of resources (and money) is invested in retaining existing employees through various initiatives. However, the organization (operations manager) wants to reduce the costs of retaining employees. To do so, we propose to limit incentives to only those employees who are at risk of leaving.

The people area was given the challenge of identifying patterns and characteristics of employees who leave the organization. In addition, they must use the information gathered to predict whether an employee is at risk of leaving.

Objetivo/Objective

ESPAÑOL

1. Identificar las distintas variables o factores que hacen a los colaboradores renunciar.
2. Construir un modelo que sea capaz de predecir si un colaborador va a renunciar o no.

ENGLISH

1. To identify the different factors that drive attrittion in riot games employees.
2. To build a model to predict if an employee wil attrite or not.

Let´s goooo

In [102]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

#Para crecer los datos usando puntaje Z
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

#Algortimos que hay que utilizar
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier

#Metricas para evluar el modelo
from sklearn.metrics import confusion_matrix, classification_report, precision_recall_curve

#Para afinar el modelo
from sklearn.model_selection import GridSearchCV

#Para ignorar las warnings
import warnings
warnings.filterwarnings("ignore")
```

Aqui agregamos la Data / Load the data

Debes subir tu excell con la base de datos que recolectaste de tu propia organización. En este caso utilizaremos una base ficticia de Riot Games.

In [103]:
```
df = pd.read_excel(r'C:\Users\marti\OneDrive\Escritorio\Importante\Data Science -MIT\HR_RiotGames_Employee_Attrition_Dataset.xlsx')
```

In [104]:
```
df.head(10)
```

Out[104]:

	EmployeeNumber	Attrition	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EnvironmentSatisfaction	...	StandardHours	StockOptionLevel	TotalWorkingYears	TrainingTimesLastYear	WorkLifeBala
0	1	Yes	41	Travel_Rarely	1102	Business Department	1	2	Programming	2	...	80	0	8	0	
1	2	No	49	Travel_Frequently	279	Technology Department	8	1	Programming	3	...	80	1	10	3	
2	3	Yes	37	Travel_Rarely	1373	Technology Department	2	2	Other	4	...	80	0	7	3	
3	4	No	33	Travel_Frequently	1392	Technology Department	3	4	Programming	4	...	80	0	8	3	
4	5	No	27	Travel_Rarely	591	Technology Department	2	1	Business	1	...	80	1	6	3	
5	6	No	32	Travel_Frequently	1005	Technology Department	2	2	Programming	4	...	80	0	8	2	
6	7	No	59	Travel_Rarely	1324	Technology Department	3	3	Business	3	...	80	3	12	3	
7	8	No	30	Travel_Rarely	1358	Technology Department	24	1	Programming	4	...	80	1	1	2	
8	9	No	38	Travel_Frequently	216	Technology Department	23	3	Programming	4	...	80	0	10	2	
9	10	No	36	Travel_Rarely	1299	Technology Department	27	3	Business	3	...	80	2	17	3	

10 rows × 35 columns

In [105]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2940 entries, 0 to 2939
Data columns (total 35 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 0   EmployeeNumber            2940 non-null   int64 
 1   Attrition                 2940 non-null   object
 2   Age                       2940 non-null   int64 
 3   BusinessTravel            2940 non-null   object
 4   DailyRate                 2940 non-null   int64 
 5   Department                2940 non-null   object
 6   DistanceFromHome          2940 non-null   int64 
 7   Education                 2940 non-null   int64 
 8   EducationField            2940 non-null   object
 9   EnvironmentSatisfaction   2940 non-null   int64 
 10  Gender                    2940 non-null   object
 11  HourlyRate                2940 non-null   int64 
 12  JobInvolvement            2940 non-null   int64 
 13  JobLevel                  2940 non-null   int64 
 14  JobRole                   2940 non-null   object
 15  JobSatisfaction           2940 non-null   int64 
 16  MaritalStatus             2940 non-null   object
 17  MonthlyIncome             2940 non-null   int64 
 18  MonthlyRate               2940 non-null   int64 
 19  NumCompaniesWorked        2940 non-null   int64 
 20  Over18                    2940 non-null   object
 21  OverTime                  2940 non-null   object
 22  PercentSalaryHike         2940 non-null   int64 
 23  PerformanceRating         2940 non-null   int64 
 24  RelationshipSatisfaction  2940 non-null   int64 
 25  StandardHours             2940 non-null   int64 
 26  StockOptionLevel          2940 non-null   int64 
 27  TotalWorkingYears         2940 non-null   int64 
 28  TrainingTimesLastYear     2940 non-null   int64 
 29  WorkLifeBalance           2940 non-null   int64 
 30  YearsAtCompany            2940 non-null   int64 
 31  YearsInCurrentRole        2940 non-null   int64 
 32  YearsSinceLastPromotion   2940 non-null   int64 
 33  YearsWithCurrManager      2940 non-null   int64 
 34  Gamers                    2940 non-null   object
dtypes: int64(25), object(10)
memory usage: 804.0+ KB
```

¿Que vemos? Nos permite saber que cosas son Int64 (numeros) y que cosas son Object (letras)

1. Hay un total de 2949 observaciones y 35 columnas
2. Todas las columnas son "non-null", por ende, no hay data que falte en el excel.

In [106]:
```
df.nunique ()
```

Out[106]:
```
EmployeeNumber              2940
Attrition                      2
Age                           43
BusinessTravel                 3
DailyRate                    886
Department                     3
DistanceFromHome              29
Education                      5
EducationField                 6
EnvironmentSatisfaction        4
Gender                         2
HourlyRate                    71
JobInvolvement                 4
JobLevel                       5
JobRole                        9
JobSatisfaction                4
MaritalStatus                  3
MonthlyIncome               1349
MonthlyRate                 1427
NumCompaniesWorked            10
Over18                         1
OverTime                       2
PercentSalaryHike             15
PerformanceRating              2
RelationshipSatisfaction       4
StandardHours                  1
StockOptionLevel               4
TotalWorkingYears             40
TrainingTimesLastYear          7
WorkLifeBalance                4
YearsAtCompany                37
YearsInCurrentRole            19
YearsSinceLastPromotion       16
YearsWithCurrManager          18
Gamers                         2
dtype: int64
```

¿Que vemos? Aqui se muestran el numero de "opciones" que tiene cada una de las categorias.

1. Employee Number es el codigo de cada colaborador, es algo unico. Asi que esta columna no nos va a gregar ningun valor.
2. Las columnas de Over18 y StandarHours solo tienenn un valor, asi que estas columnas tambien no nos agregaran valor.
3. Con estos numeros podemos identificar cuales columnas son continuas y cuales son categoricas.

In [107]:
```
#Columnas que seran eliminadas
df=df.drop(['EmployeeNumber','Over18','StandardHours'], axis=1)
```

In [108]:
```
#Creando columnas numericas
```

In [109]:
```
#Creando columnas numericas
num_cols=['DailyRate','Age','DistanceFromHome','MonthlyIncome','MonthlyRate','PercentSalaryHike','TotalWorkingYears','YearsAtCompany','NumCompaniesWorked','HourlyRate','YearsInCurrentRole','Y

#Creando columnas categoricas
cat_cols=['Attrition','OverTime','BusinessTravel','Department','Education','EducationField','JobSatisfaction','EnvironmentSatisfaction','WorkLifeBalance','StockOptionLevel','Gender','Performar
```

A explorar y analizar la data / Exploratory Data Analysis

ANALISIS A LAS COLUMNAS NUMERICAS Y VER QUE SACAMOS DE AHÍ

In [110]:
```
#Resumen estadistico...
df[num_cols].describe().T
```

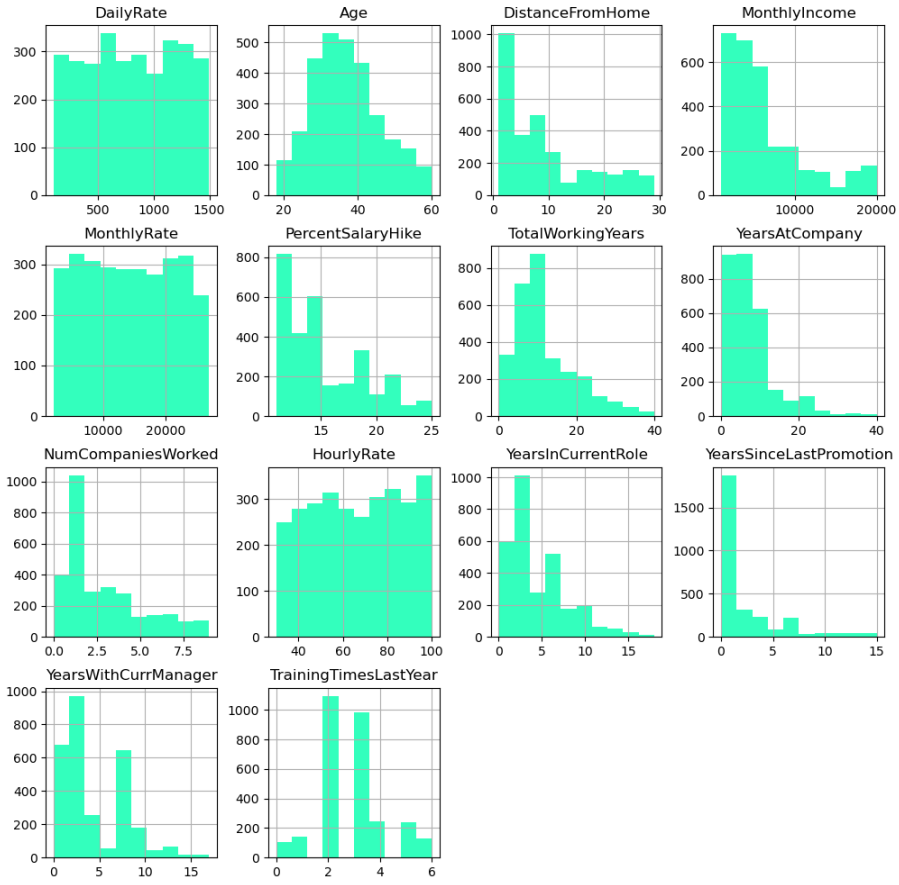
Out[110]:

	count	mean	std	min	25%	50%	75%	max
DailyRate	2940.0	802.485714	403.440447	102.0	465.0	802.0	1157.0	1499.0
Age	2940.0	36.923810	9.133819	18.0	30.0	36.0	43.0	60.0
DistanceFromHome	2940.0	9.192517	8.105485	1.0	2.0	7.0	14.0	29.0
MonthlyIncome	2940.0	6502.931293	4707.155770	1009.0	2911.0	4919.0	8380.0	19999.0
MonthlyRate	2940.0	14313.103401	7116.575021	2094.0	8045.0	14235.5	20462.0	26999.0
PercentSalaryHike	2940.0	15.209524	3.659315	11.0	12.0	14.0	18.0	25.0
TotalWorkingYears	2940.0	11.279592	7.779458	0.0	6.0	10.0	15.0	40.0
YearsAtCompany	2940.0	7.008163	6.125483	0.0	3.0	5.0	9.0	40.0
NumCompaniesWorked	2940.0	2.693197	2.497584	0.0	1.0	2.0	4.0	9.0
HourlyRate	2940.0	65.891156	20.325969	30.0	48.0	66.0	84.0	100.0
YearsInCurrentRole	2940.0	4.229252	3.622521	0.0	2.0	3.0	7.0	18.0
YearsSinceLastPromotion	2940.0	2.187755	3.221882	0.0	0.0	1.0	3.0	15.0
YearsWithCurrManager	2940.0	4.123129	3.567529	0.0	2.0	3.0	7.0	17.0
TrainingTimesLastYear	2940.0	2.799320	1.289051	0.0	2.0	3.0	3.0	6.0

Observaciones:

- **La edad media de los empleados esta alrededor de los 37 años.** Tienen una muy buena diversidad de edades en la empresa, ya que tienen un rango entre los 18 años y 60 años en los colaboradores.
- Al menos el **50% de los empleados viven en un radio de 7km de la organización** (consideremos para este caso que RiotGames cuenta con solo una sede, que es su casa matriz en Los Angeles). El colaborador mas lejano esta a 29km.
- El **sueldo mensual medio de un empleado de RiotGames es de 6500USD.** Los sueldos varian entre 1k-20k USD, lo cual es bastante razonable. Si se ven los cuartiles y el aumento de sueldo recibido en cada uno de estos, se puede observar una desproporcion entre el tercer cuartil (75%) y el valor maximo, donde el primero gana en promedio 8.400USD y el maximo gana un promedio de 20.000USD. En resumen, **los que mas ganan en la empresa tienen ingresos desproporcionados con el resto (lo cual es bastante comun en las organización).**
- La subida salarial media de un colaborador ronda en el 15% y **al menos el 50% de los colaboradores obtuvo un aumento salarial del 14% o menos.** Me quito el sombrero, demuestra que tienen un buen modelo de compensaciones en la organización y que se esta pendiente de esto.
- El **tiempo transcurrido desde que un colaborador obtuvo un ascenso es de ~2,19**. La mayoría de los empleados han sido ascendidos en el último año.
- La media de **años vinculado en la empresa los colaboradores es de 7.**

In [111]:
```
#Ahora veremos la distribución de los datos. Crearemos histogramas
df[num_cols].hist(figsize=(12,12), color='#33FFBD')
plt.show()
```



Observaciones:

- La distribución de la edad de los colaboradores se parece a una distribución normal. Donde la mayoria esta entre los 25 y 50.
- La mayoria de los colaboradores viven cerca de la casa matriz y cada vez va bajando mas la gente en relacion a la distancia.
- Los ingresos mensuales y el número total de años de trabajo están sesgados a la derecha, lo que indica que la mayoría de los trabajadores ocupan puestos de nivel inicial o medio en la empresa.
- El porcentaje de aumento salarial está desviado hacia la derecha, lo que significa que la mayoría de los empleados reciben un porcentaje de aumento salarial más bajo.
- Existe una buena proporcion en YearsAtCompany respecto a los colaboradores que llevan mas de 10 años. Lo que demuestra que hay varias personas leales a la compañia.
- La distribución de la variable YearsSinceLastPromotion indica que algunos empleados no han recibido un ascenso en 10-15 años y siguen trabajando en la empresa. Se supone que estos empleados tienen una gran experiencia laboral y ocupan puestos de alta dirección, como cofundadores, empleados de la C-suite, etc (pero abria que indagar estos perfiles).

ANALISIS A LAS COLUMNAS CATEGORICAS Y VER QUE SACAMOS DE AHÍ

Con esto podremos ver el porcentaje de las categorias y cuantas existen por "opcion".

In [112]:
```
for i in cat_cols:
    print(df[i].value_counts(normalize = True))
    print('*'*40)
```

```
No     0.838776
Yes    0.161224
Name: Attrition, dtype: float64
****************************************
No     0.717007
Yes    0.282993
Name: OverTime, dtype: float64
****************************************
Travel_Rarely        0.709524
Travel_Frequently    0.188435
Non-Travel           0.102041
Name: BusinessTravel, dtype: float64
****************************************
Technology Department    0.653741
Business Department      0.303401
Human Resources          0.042857
Name: Department, dtype: float64
****************************************
3    0.389116
4    0.270748
2    0.191837
1    0.115646
5    0.032653
Name: Education, dtype: float64
****************************************
Programming         0.412245
Business            0.315646
Marketing           0.108163
Technical Degree    0.089796
Other               0.055782
Human Resources     0.018367
Name: EducationField, dtype: float64
****************************************
4    0.312245
3    0.300680
1    0.196599
2    0.190476
Name: JobSatisfaction, dtype: float64
****************************************
3    0.308163
4    0.303401
2    0.195238
1    0.193197
Name: EnvironmentSatisfaction, dtype: float64
****************************************
3    0.607483
2    0.234014
4    0.104082
1    0.054422
Name: WorkLifeBalance, dtype: float64
****************************************
0    0.429252
1    0.405442
2    0.107483
3    0.057823
Name: StockOptionLevel, dtype: float64
****************************************
Male      0.6
Female    0.4
Name: Gender, dtype: float64
****************************************
3    0.846259
4    0.153741
Name: PerformanceRating, dtype: float64
****************************************
3    0.590476
2    0.255102
4    0.097959
1    0.056463
Name: JobInvolvement, dtype: float64
****************************************
1    0.369388
2    0.363265
3    0.148299
4    0.072109
5    0.046939
Name: JobLevel, dtype: float64
****************************************
Marketing Analyst         0.221769
Designer                  0.198639
Developer                 0.176190
Manufacturing Director    0.098639
Customer Experience       0.089116
Manager                   0.069388
Business Analyst          0.056463
Research Director         0.054422
Human Resources           0.035374
Name: JobRole, dtype: float64
****************************************
Married     0.457823
Single      0.319728
Divorced    0.222449
Name: MaritalStatus, dtype: float64
****************************************
3    0.312245
4    0.293878
2    0.206122
1    0.187755
Name: RelationshipSatisfaction, dtype: float64
****************************************
Yes    0.727551
No     0.272449
Name: Gamers, dtype: float64
****************************************
```

Observaciones:

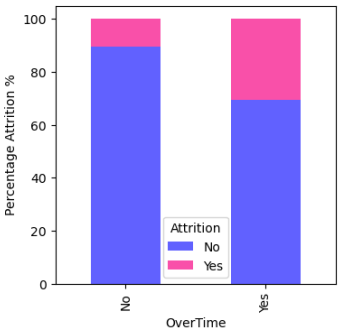
- **La tasa de bajas de colaboradores es del 16%.**
- Alrededor de **28% de los trabajadores trabajan horas extras**. Un gran numero que influye en el balance de calidad de vida personal-profesional (factor de estres).
- Aproximadamente un **73% de los colaboradores vienen del mundo de la programación y business**. Y el 65% de los colaboradores trabajan en el **departamento de tecnología**.
- Cerca del **40% de los trabajadores evaluan su satisfacción laboral o ambiente laboral como baja o media**. Lo cual es un gran porcentaje de la plantilla.
- 19% de los coolaboradores viaja frecuentemente por el trabajo y 71% de los trabajadores rara vez.
- Mas del 30% de los empleados muestran una implicancia baja o media. Y mas del 80% de los empleados no tiene stockoptions o tiene muy pocas.
- El **73% de los colaboradores de RiotGames tiene relación con el mundo gamers.** Lo cual podría implicar relación con el producto y cultura de la organización.
- **Ninguno de los empleados a recibido una puntuación menor a 3 (excelente) respecto a su evaluación de desempeño (rendimiento)**. El 85% dfe sus colaboradores es evaluado 3 (excelente) y el resto 4 (sobre lo esperado). Dos escenarios:
  - Los colaboradores de Riot Games son increiblemente buenos.
  - El proceso de evaluación es indulgente. Y falta capacitar a la organizacion y lideres de como generar una correcta evaluación de desempeño.

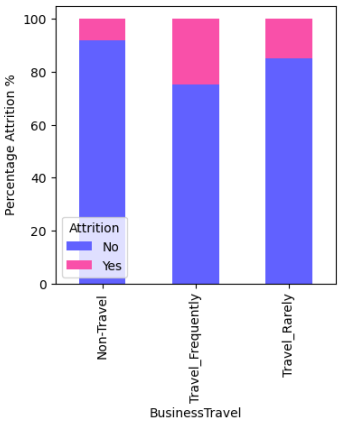
MixUp (Analisis columnas categoricas) / Multivariate Analysis

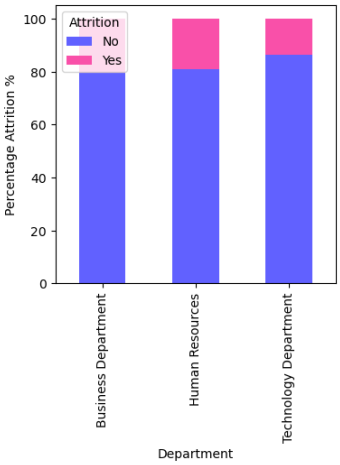
Ahora analizaremos el relacion entre la tasa de abandono (attrition) con todas las otras variables categoricas.

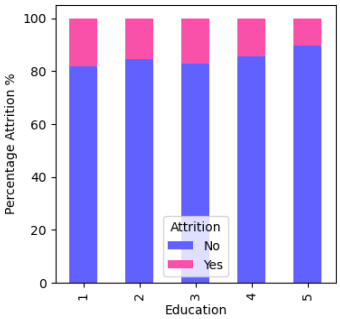
In [113]:

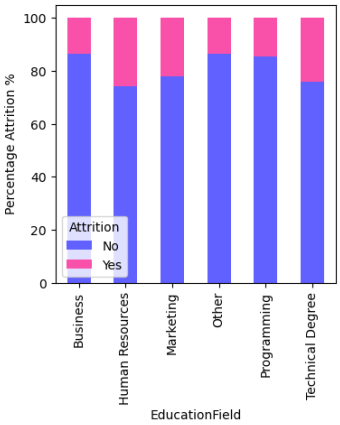
```
for i in cat_cols:
    if i !='Attrition':
        (pd.crosstab(df[i], df['Attrition'], normalize ='index')*100).plot(kind='bar', figsize=(4,4), stacked= True, color =['#6161FF','#f950a9'])
        plt.ylabel('Percentage Attrition %')
```

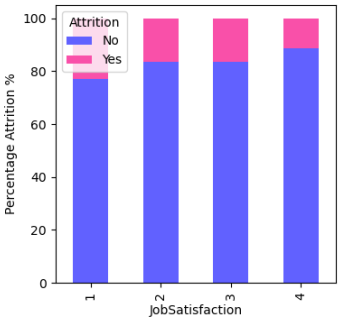


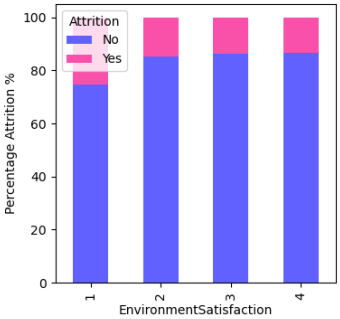


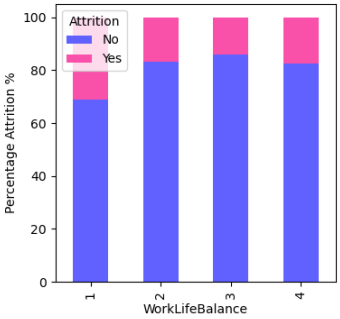


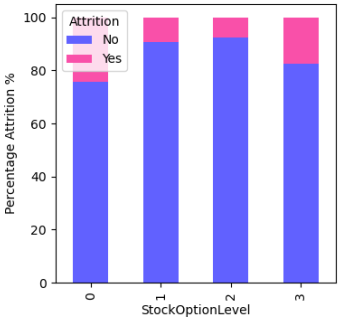


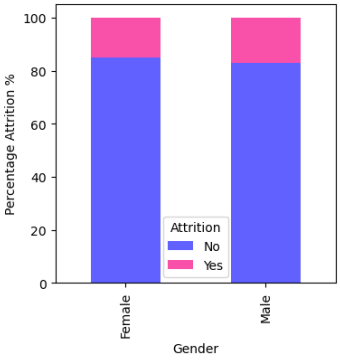

Percentage Attrition %
Attrition
No
Yes
Business
Human Resources
Marketing
Other
Programming
Technical Degree
EducationField

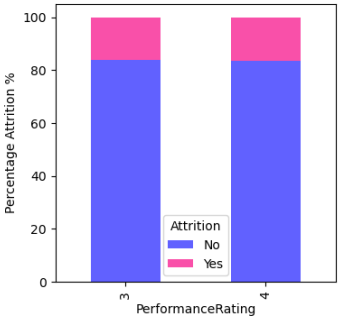

Percentage Attrition %
Attrition
No
Yes
JobSatisfaction

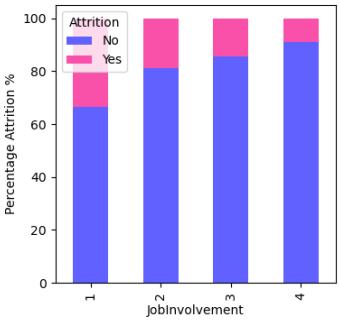

Percentage Attrition %
Attrition
No
Yes
EnvironmentSatisfaction

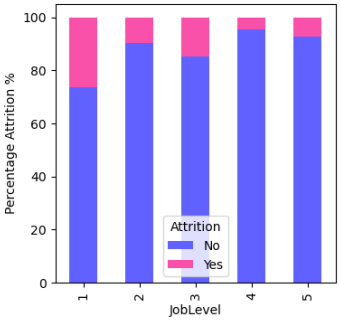

Percentage Attrition %
Attrition
No
Yes
WorkLifeBalance

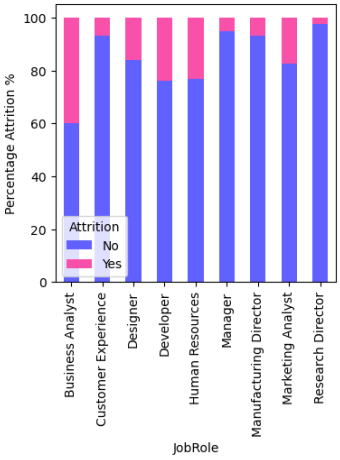

Percentage Attrition %
Attrition
No
Yes
StockOptionLevel

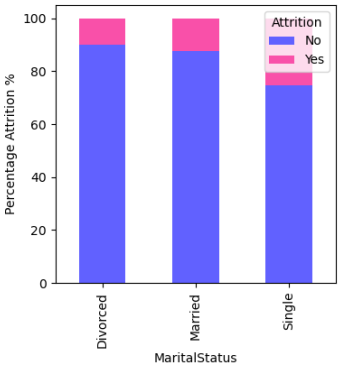

Percentage Attrition %
Attrition
No
Yes
Female
Male
Gender

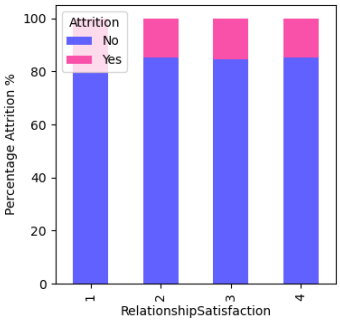

Percentage Attrition %
Attrition
No
Yes
3
4
PerformanceRating

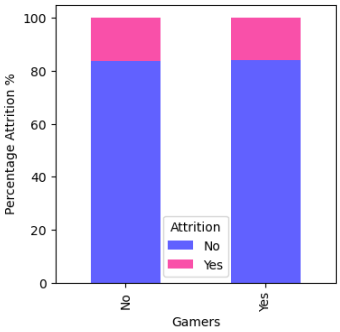

Percentage Attrition %
Attrition
No
Yes
1
2
3
4
JobInvolvement


Percentage Attrition %
Attrition
No
Yes
1
2
3
4
5
JobLevel


Percentage Attrition %
Attrition
No
Yes
Business Analyst
Customer Experience
Designer
Developer
Human Resources
Manager
Manufacturing Director
Marketing Analyst
Research Director
JobRole







Observaciones:

- **Los trabajadores que hacen horas extra tienen un 30% más de probabilidad de abandono**. Lo cual es alto si se compara con el 10% de las personas que no tienen overtime.
- Como se comento antes, **la mayoria trabaja en el departamento de tecnología y es este mismo departamento con el menor indice de rotación**. La probabilidad de que se produzcan bajas es del ~15%.
- **Los cargos HR, Technical degree y Marketin son los que muestran un mayor indice de rotación del personal**. Se podria hipotetizar que se prioriza la retención de cargos que generan mas ingreso a la compañia y se descuidan los cargos que generan un mayor "costo" a la organización. Ya que los devs y los de business son los que menos rotación tienen y a la vez son escenciales para desarrollar y crear los productos.
- Se demuestra un increible correlación entre el grado de implicancia y el nivel de bajas de colaboradores. Donde **entre menos implicancia mayor es el porcentaje de rotación**. Donde los que estan en el nivel mas bajo de implicancia tienen un porcentaje del 35% de abandono. Seria interesante evaluar la relación entre implicancia y stock options, ver si existe una relación entre estos factores y el nivel de rotación. Ya que los que no tienen stock options son los que mas rotan, pero despues les continua los que más tienen stock options ¿A quien les estamos dando las stock options? ¿Tiene relación con el grado de implicancia? ¿Tener más StockOptions significa tener más implicancia con el trabajo?
- **Los empleados con un nivel de empleo inferior (entry level) también sufren más bajas**, y los empleados con un nivel de empleo de 1 tienen casi un 25% de probabilidades de abandonar. Puede tratarse de empleados jóvenes que tienden a explorar más opciones en las etapas iniciales de sus carreras.
- Que sean del **mundo gamers no demuestra implicancia en el nivel de rotación del personal**. No es un factor evidentemente fuerte en el nivel de retencion.

MixUp (Analisis columnas numericas con Attrition)

Vamos a ver la relación entre la rotación y cada una de las variables numericas!

In [114]:
```
df.groupby(['Attrition'])[num_cols].mean()
```

Out[114]:

Attrition	DailyRate	Age	DistanceFromHome	MonthlyIncome	MonthlyRate	PercentSalaryHike	TotalWorkingYears	YearsAtCompany	NumCompaniesWorked	HourlyRate	YearsInCurrentRole	YearsSinceLastPromotion	YearsWithCurrManage
No	812.504461	37.561233	8.915653	6832.739659	14265.779400	15.231144	11.862936	7.369019	2.645580	65.952149	4.484185	2.234388	4.36739
Yes	750.362869	33.607595	10.632911	4787.092827	14559.308017	15.097046	8.244726	5.130802	2.940928	65.573840	2.902954	1.945148	2.85232

Observaciones:

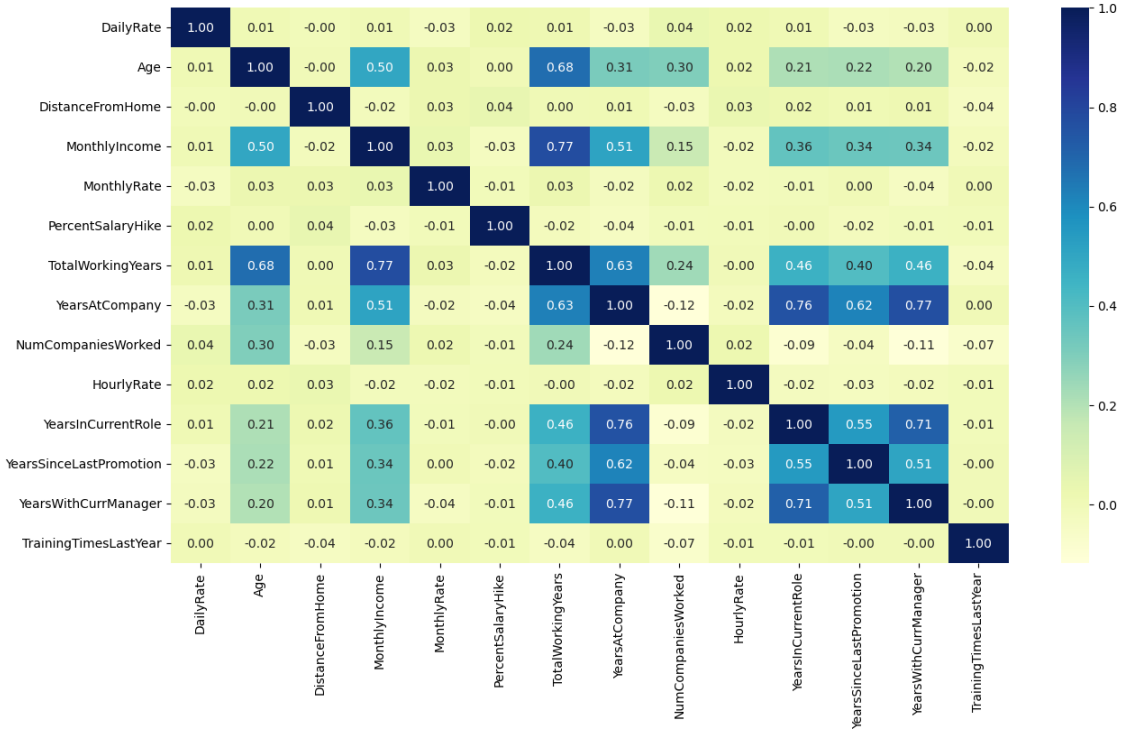
- **Los colaboradores que abandonan la empresa tienen casi un 30% menos de ingresos medios y un 30% menos de experiencia laboral de los que no lo hacen**. Se puede hipotetizar que lo hacen para explorar nuevas opciones y/o aumentar su salario con un cambio de empresa.
- La distancia es relevante? Los empleados que se dan de baja también tienden a vivir un 16% más lejos de la oficina que los queno se dan de baja. ¿Si se aplica teletrabajo como se vera afectado esto?

ANALISIS DE RELACIÓN ENTRE LAS DISTINTAS VARIABLES NUMERICAS

Vamos a ver como se correlacionan entre ellas. Donde haremos un mapa de calor y tendremos que identificar relaciones interesantes que puedan significar algo.

In [115]:
```
plt.figure(figsize=(15,8))

sns.heatmap(df[num_cols].corr(), annot = True, fmt = '0.2f', cmap = 'YlGnBu')
```

Out[115]: <Axes: >



Observaciones:

- En las correlación es importante ver que relación de variables se acerca al 1.0 (entre mas cercano a ese numero mayor es la correlación).
- **Experiencia laboral, sueldo mensual, años en Riot Games y años con el mismo líder estan altamente correlacionado entre si y con la edad de los empleados**, en donde el ultimo es bastante logico, ya que a medida que uno va teniendo mas años tiende a tener mas de las otras variables.
- **Los años en la empresa y los años en el puesto actual están correlacionados con los años transcurridos desde la última promoción**, lo que significa que la empresa no está dando promociones en el momento adecuado. Es decir, entre mas años llevo en el cargo actual, más tiempo ha pasado desde mi ultima promoción.

Resumen del analsis de datos

1. La edad media de los empleados ronda los 37 años. Tiene un rango elevado, de 18 años a 60, lo que indica una buena diversidad de edades en la organización.
2. Al menos el 50% de los empleados viven en un radio de 7 km de la organización. Sin embargo, hay algunos valores extremos, ya que el valor máximo es de 29 km.
3. El ingreso mensual medio de un empleado es de 6500 USD. Tiene un alto rango de valores de 1K-20K USD, lo que es de esperar para la distribución de ingresos de cualquier organización. Hay una gran diferencia entre el valor del tercer cuartil (alrededor de 8400 USD) y el valor máximo (casi 20000 USD), lo que demuestra que las personas que más ganan en la empresa tienen unos ingresos desproporcionadamente grandes en comparación con el resto de los empleados. De nuevo, esto es bastante común en la mayoría de las organizaciones. Sin embargo, se podria optar por algunas politicas o analisis del modelo de compensación como una buena practica.
4. La subida salarial media de un empleado ronda el 15%. Al menos el 50% de los empleados obtuvo un aumento salarial del 14% o menos, siendo el aumento salarial máximo del 25%.
5. El número medio de años que un empleado lleva vinculado a la empresa es de 7.
6. Por término medio, el número de años transcurridos desde que un empleado obtuvo un ascenso es de ~2,19. La mayoría de los empleados han sido ascendidos en el último año.
7. La distribución por edades se aproxima a una distribución normal, con la mayoría de los empleados entre 25 y 50 años.
8. DistanceFromHome también tiene una distribución sesgada a la derecha, lo que significa que la mayoría de los empleados viven cerca del trabajo, pero hay algunos que viven más lejos.
9. Los ingresos mensuales y el número total de años de trabajo están sesgados a la derecha, lo que indica que la mayoría de los trabajadores ocupan puestos de nivel inicial o medio en la empresa.
10. El porcentaje de aumento salarial está sesgado a la derecha, lo que significa que la mayoría de los empleados están recibiendo aumentos salariales de menor porcentaje.
11. La distribución de la variable YearsAtCompany muestra una buena proporción de trabajadores con más de 10 años, lo que indica un número significativo de empleados leales a la organización.
12. La distribución de YearsInCurrentRole tiene tres picos en 0, 2 y 7. . Hay pocos empleados que hayan permanecido en el mismo puesto durante 15 años o más.
13. La distribución de la variable YearsSinceLastPromotion indica que algunos empleados no han recibido un ascenso en 10-15 años y siguen trabajando en la empresa. Se supone que estos empleados tienen una gran experiencia laboral y ocupan puestos de alta dirección, como cofundadores, empleados de la C-suite, etc.
14. Las distribuciones de DailyRate, HourlyRate y MonthlyRate parecen uniformes y no aportan mucha información. Podría ser que la tasa diaria se refiera a los ingresos obtenidos por día extra trabajado, mientras que la tasa horaria podría referirse al mismo concepto aplicado a las horas extra trabajadas al día. Dado que estas tasas tienden a ser muy similares para varios empleados de un mismo departamento, eso explica la distribución uniforme que muestran.
15. La tasa de abandono de los empleados es del 16%.
16. Alrededor del 28% de los empleados hacen horas extraordinarias. Esta cifra parece estar en el lado más alto y podría indicar una vida laboral estresada de los empleados. Los empleados que hacen horas extras tienen más de un 30% de probabilidades de abandono, una cifra muy alta comparada con el 10% de probabilidades de abandono de los empleados que no hacen horas extras.
17. El 71% de los empleados ha viajado pocas veces, mientras que alrededor del 19% tiene que viajar con frecuencia.
18. Alrededor del 73% de los empleados tienen formación en el campo de las Programming y Business.
19. Más del 65% de los empleados trabajan en el departamento de Technology Department y Business Department. La probabilidad de desgaste es de ~15%.
20. Casi el 40% de los empleados tienen una satisfacción baja (1) o media (2) con el trabajo y el entorno de la organización, lo que indica que la moral de la empresa parece ser algo baja.
21. Más del 30% de los empleados muestran una implicación en el trabajo de baja (1) a media (2).
22. Más del 80% de los empleados no tienen opciones sobre acciones o tienen muy pocas.
23. En cuanto a la valoración del rendimiento, ninguno de los empleados ha recibido una valoración inferior a 3 (excelente). Alrededor del 85% de los empleados tiene una valoración del rendimiento igual a 3 (excelente), mientras que el resto tiene una valoración de 4 (sobresaliente). Esto podría significar que la mayoría de los empleados son de alto rendimiento, o lo más probable es que la organización sea muy indulgente con su proceso de evaluación del rendimiento. Recomendación de capacitación a lideres respecto a la nivelación de puntuación.
24. Los empleados que trabajan como Business Analyst tienen una tasa de abandono de alrededor del 40%, mientras que los de RR.HH. y los Developer tienen una tasa de abandono de alrededor del 25%. Los departamentos de business tiene tasa de toración mas alta que los cargos de tecnología, lo cual es bastante raro en este nivel de industria. Habria que analizar en profundidad la area de Businees (remuneración, líderazgo, distribución de proyectos, etc.).
25. Cuanto menor es la implicación en el trabajo del empleado, mayores parecen ser sus posibilidades de abandono, con un 35% de abandono entre los empleados con una implicación en el trabajo de 1 punto. Esto podría deberse a que los empleados con una menor implicación en el trabajo podrían sentirse excluidos o menos valorados y ya han empezado a explorar nuevas opciones, lo que conduce a una mayor tasa de abandono.
26. Los empleados con un nivel de empleo inferior también sufren más bajas, y los empleados con un nivel de empleo de 1 tienen casi un 25% de probabilidades de abandonar. Puede tratarse de empleados jóvenes que tienden a explorar más opciones en las etapas iniciales de sus carreras.
27. Una valoración baja de la conciliación de la vida laboral y personal lleva a los empleados a abandonar la empresa; ~30% de los que se encuentran en la categoría de valoración 1 presentan bajas.
28. Los empleados que abandonan la empresa tienen una media de ingresos casi un 30% inferior y un 30% menos de experiencia laboral que los que no lo hacen. Estos podrían ser los empleados que buscan explorar nuevas opciones y/o aumentar su salario con un cambio de empresa.
29. Los empleados que se dan de baja también tienden a vivir un 16% más lejos de la oficina que los que no se dan de baja. Los desplazamientos más largos para ir y volver del trabajo podrían significar que tienen que dedicar más tiempo/dinero cada día, y esto podría estar provocando insatisfacción laboral y el deseo de abandonar la organización.
30. La experiencia laboral total, los ingresos mensuales, los años en la empresa y los años con los jefes actuales están muy correlacionados entre sí y con la edad del empleado, lo cual es fácil de entender, ya que estas variables muestran un aumento con la edad para la mayoría de los empleados.
31. Los años en la empresa y los años en el puesto actual están correlacionados con los años transcurridos desde el último ascenso, lo que significa que la empresa no concede los ascensos en el momento adecuado.

TERMINA LA PARTE I (ANALISIS DE LOS DATOS) Y COMIENZA PARTE II (ALGORTIMO PARA PREDECIR SALIDAS)

Modelo de Predicción de Salidas / Model Bouilding - Approach

HEMOS ANALIZADO LA DATA, AHORA VAMOS A CONTRUIR EL MODELO PARA PREDECIR LAS SALIDAS DE LOS COLABORADORES DE RIOTGAMES

De aqui en adelante el contenido es mas complicado de explicar, pero hare el mejor esfuerzo.

1. Preparar la dara para modelar.
2. Dividir los datos en conjuntos de entrenamientos y de test.
3. Construir el modelo con los datos de entrenamiento.
4. Ajustar el modelo.
5. Probar los datos en el conjunto de test.

Preparando los datos para el modelo

Crando "dummy" variables para las variables categoricas. En general, es cambiar palabras por numeros.

In [116]:
```
# Creando la lista de columna en donde debemos crear dummy variables.
to_get_dummies_for = ['BusinessTravel', 'Department', 'Education', 'EducationField', 'EnvironmentSatisfaction', 'Gender', 'JobInvolvement', 'JobLevel', 'JobRole', 'MaritalStatus','Gamers']

# Creando las dummy variables
df = pd.get_dummies(data = df, columns = to_get_dummies_for, drop_first = True)

# Mapeando overtime and attrition
dict_OverTime = {'Yes': 1, 'No': 0}
dict_attrition = {'Yes': 1, 'No': 0}

df['OverTime'] = df.OverTime.map(dict_OverTime)
df['Attrition'] = df.Attrition.map(dict_attrition)
```

In [117]:
```
#Separando las variables independientes (X) entre las variables dependientes (Y)
Y = df.Attrition
X = df.drop(columns = ['Attrition'])
```

Ahora bien.... Scaling the data

En este conjunto las variables independientes de este conjunto de datos tienen escalas diferentes. Cuando las características tienen escalas diferentes entre sí, existe la posibilidad de que se dé una mayor ponderación a las características que tienen una mayor magnitud, y dominarán sobre otras características cuyos cambios de magnitud pueden ser menores, pero cuyos cambios porcentuales pueden ser igual de significativos o incluso mayores. Esto afectará al rendimiento del algoritmo de aprendizaje automático.

Lo solucionaremos algo que se llama "Feature Scaling" (escalado de caracteristicas. En otras palabras, escalar el conjunto de datos para dar a cada variable transformada una escala comparable.

En este problema, utilizaremos el método Standard Scaler, que centra y escala el conjunto de datos utilizando el Z-Score.

Estandariza las características restando la media y escalándola para tener una varianza unitaria.

La puntuación estándar de la muestra x se calcula como

z = (x - u) / s

donde u es la media de las muestras de entrenamiento (cero) y s es la desviación estándar de las muestras de entrenamiento.

In [118]:
```
# Scaling the data
sc = StandardScaler()

X_scaled = sc.fit_transform(X)

X_scaled = pd.DataFrame(X_scaled, columns = X.columns)
```

Dividiendo la data en 70% train y 30% test sets

Para que el nivel, o mejor dicho porcentaje, de "Attrition" se mantenga equilibrado en ambas muestras. Ocuparemos un muestreo estratificado para garantizar tener la misma frecencia.

En palabras mas simples, queremos que en ambas muestras (train y test) tengamos un porcentaje de 16% attrition.

In [119]:
```
# Splitting the data
x_train, x_test, y_train, y_test = train_test_split(X_scaled, Y, test_size = 0.3, random_state = 1, stratify = Y)
```

Criterio de evaluación del modelo

El modelo puede realizar dos tipos de predicciones erróneas:

1. Predecir que un empleado abandonará cuando en realidad no lo hace.
2. Predecir que un empleado no se dará de baja cuando en realidad se da de baja.

¿Qué caso es más importante?

La respuesta correcta seia la numero dos. Predecir que el empleado no abandonara cuando en realidad se da de baja. Ya que esto se consideraria gran fallo para cualquier predictor de abandono de empleados y, por lo tanto, es el caso más importante de predicciones erróneas. ¿Cómo reducir esta pérdida, es decir, la necesida de reducir falsos negativos?

La empresa querría maximizar los Recall (recuperación); cuanto mayor sea, mayores serán las posibilidades de minimizar los falsos negativos. Por lo tanto, el objetivo debe ser aumentar la recuperación (minimizar los falsos negativos) o, en otras palabras, identificar muy bien los verdaderos positivos (es decir, la clase 1), de modo que la empresa pueda ofrecer incentivos para controlar la tasa de abandono, especialmente para los mejores. Esto ayudaría a optimizar el coste global del proyecto para retener a los mejores talentos.

Además, vamos a crear una función para calcular e imprimir el informe de clasificación y la matriz de confusión para que no tengamos que reescribir el mismo código repetidamente para cada modelo.

Recall: medida de la capacidad de un modelo de aprendizaje automático para identificar todos los casos relevantes de una clase determinada. En este caso, la clase positiva serian los empleados que abandonan las empresa y que el modelo identifica correctamente.

In [120]:
```
def metrics_score(actual, predicted):

    print(classification_report(actual, predicted))

    cm = confusion_matrix(actual, predicted)

    plt.figure(figsize = (8, 5))

    sns.heatmap(cm, annot = True, fmt = '.2f', xticklabels = ['Not Attrite', 'Attrite'], yticklabels = ['Not Attrite', 'Attrite'])

    plt.ylabel('Actual')

    plt.xlabel('Predicted')

    plt.show()
```

Creando el Modelo

Vamos a crear 2 modelos diferentes:

1. Logistic Regression (Regresión)
2. K-Nearest Neighbors (K-NN) (Vecinos próximos)

Vamos por el primero... Modelo de regresion

In [121]:
```
# Fitting the Logistic regression model
lg = LogisticRegression()

lg.fit(x_train,y_train)
```

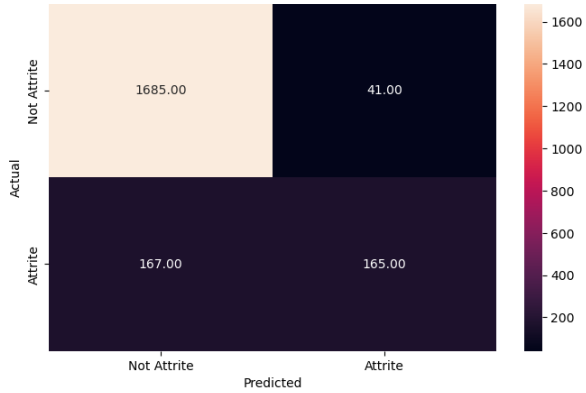
Out[121]:
```
LogisticRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

Revisemos el rendimiento de nuestro modelo

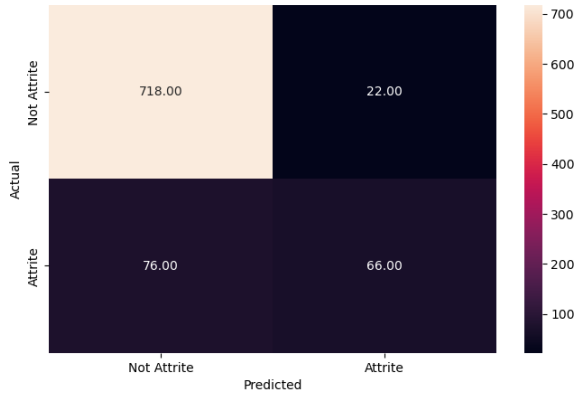
In [122]:
```
# Evaluando el rendimiento en nuestro training data
y_pred_train = lg.predict(x_train)
metrics_score(y_train, y_pred_train)
```

```
              precision    recall  f1-score   support

           0       0.91      0.98      0.94      1726
           1       0.80      0.50      0.61       332

    accuracy                           0.90      2058
   macro avg       0.86      0.74      0.78      2058
weighted avg       0.89      0.90      0.89      2058
```



In [123]:
```
# Evaluando el rendimiento en nuestro set data
y_pred_test = lg.predict(x_test)
metrics_score(y_test, y_pred_test)
```

```
              precision    recall  f1-score   support

           0       0.90      0.97      0.94       740
           1       0.75      0.46      0.57       142

    accuracy                           0.89       882
   macro avg       0.83      0.72      0.76       882
weighted avg       0.88      0.89      0.88       882
```



Observaciones:

1. La primera observación es que el modelo tiene una precisión de alrededor del 90% tanto en el conjunto de datos de entrenamiento como en el de prueba. Esto significa que el modelo predice correctamente si un empleado abandonará o no la empresa en el 90% de los casos.
2. La segunda observación es que la recuperación de la clase 1 (empleados que abandonan la empresa) sólo ronda el 50% en los datos de entrenamiento y el 46% en los datos de prueba. La recuperación mide la eficacia del modelo a la hora de identificar a los empleados que corren el riesgo de abandonar la empresa. Una recuperación baja significa que el modelo no es bueno para identificar a los empleados que corren el riesgo de abandonar la empresa.
3. La tercera observación es que, como la recuperación es baja, el modelo no funcionará bien a la hora de distinguir a los empleados que tienen más probabilidades de abandonar la empresa. Esto significa que no ayudará a reducir la tasa de abandono.
4. La cuarta observación es que, como podemos ver en la matriz de confusión, este modelo no es bueno para identificar a los empleados que corren el riesgo de abandonar la empresa. Una matriz de confusión es una tabla que muestra el rendimiento de un modelo comparando sus predicciones con los resultados reales.

En resumen, aunque el modelo tiene una alta precisión, su baja recuperación significa que no es bueno para identificar a los empleados que corren el riesgo de abandonar la empresa. Esto significa que puede no ser eficaz para reducir las bajas.

Comprobemos los coeficientes y averigüemos qué variables provocan el abandono y cuáles pueden ayudar a reducirlo.

In [124]:
```
# Visualizando el coeficiente de la regresion ligistica
cols = X.columns

coef_lg = lg.coef_

pd.DataFrame(coef_lg,columns = cols).T.sort_values(by = 0, ascending = False)
```

Out[124]:

	0
OverTime	0.954859
BusinessTravel_Travel_Frequently	0.720271
MaritalStatus_Single	0.623614
YearsSinceLastPromotion	0.551366
YearsAtCompany	0.534303
NumCompaniesWorked	0.496154
BusinessTravel_Travel_Rarely	0.446879
DistanceFromHome	0.386476
JobRole_Human Resources	0.386061
EducationField_Technical Degree	0.300268
MaritalStatus_Married	0.291952
JobLevel_5	0.272820
EducationField_Marketing	0.219969
Gender_Male	0.166946
Education_3	0.159227
EducationField_Human Resources	0.135193
Education_2	0.133188
Education_4	0.115034
JobRole_Marketing Analyst	0.096392
Education_5	0.091082
MonthlyRate	0.058916
EducationField_Programming	0.052074
HourlyRate	0.047247
EducationField_Other	0.035985
JobLevel_3	-0.001782
PerformanceRating	-0.034693
Gamers_Yes	-0.035068
Department_Technology Department	-0.048424
JobRole_Manufacturing Director	-0.072903
PercentSalaryHike	-0.074232
JobRole_Developer	-0.074448
DailyRate	-0.093721
StockOptionLevel	-0.103986
JobLevel_4	-0.169284
JobRole_Manager	-0.183561
JobRole_Customer Experience	-0.186324
WorkLifeBalance	-0.210766
TrainingTimesLastYear	-0.242308
Age	-0.271521
RelationshipSatisfaction	-0.313403
JobSatisfaction	-0.372680
YearsWithCurrManager	-0.391109
YearsInCurrentRole	-0.442057
EnvironmentSatisfaction_2	-0.445452
JobInvolvement_2	-0.480720
TotalWorkingYears	-0.490906
Department_Human Resources	-0.495187
JobRole_Designer	-0.496220
JobRole_Research Director	-0.498261
EnvironmentSatisfaction_3	-0.502891
MonthlyIncome	-0.615774
JobInvolvement_4	-0.637248
EnvironmentSatisfaction_4	-0.652362
JobLevel_2	-0.727879
JobInvolvement_3	-0.744691

Aqui podemos observar que los que estan mas arriba y son de numeros positivos corresponden a los que afectan positivamente al % de salidas de colaboradores en RiotGames.

Por el contrario, los que se encuentran abajo, con numeros negativos, son los que afectan negativamente al nivel de salidas de colaboradores.

Los coeficientes del modelo de regresión logística nos dan el logaritmo de probabilidades, que es difícil de interpretar en el mundo real. Podemos convertir el logaritmo de probabilidades en probabilidades tomando su exponencial.

In [125]:
```
odds = np.exp(lg.coef_[0]) # Transformando y encontrando las probabilidades

#Agregando las probabilidades de nuestra base de datos y ordenandolas
pd.DataFrame(odds, x_train.columns, columns = ['odds']).sort_values(by = 'odds', ascending = False)
```

Out[125]:

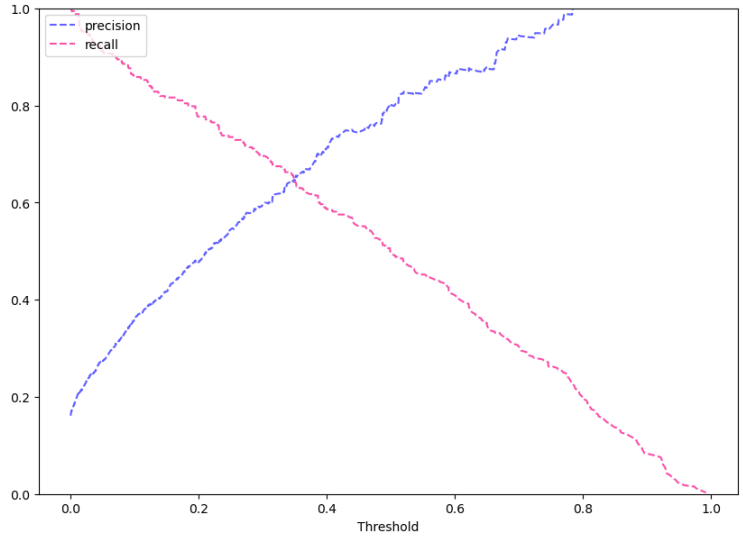
	odds
OverTime	2.598305
BusinessTravel_Travel_Frequently	2.054990
MaritalStatus_Single	1.865658
YearsSinceLastPromotion	1.735622
YearsAtCompany	1.706259
NumCompaniesWorked	1.642392
BusinessTravel_Travel_Rarely	1.563426
DistanceFromHome	1.471786
JobRole_Human Resources	1.471175
EducationField_Technical Degree	1.350220
MaritalStatus_Married	1.339039
JobLevel_5	1.313663
EducationField_Marketing	1.246038
Gender_Male	1.181691
Education_3	1.172605
EducationField_Human Resources	1.144757
Education_2	1.142464
Education_4	1.121911
JobRole_Marketing Analyst	1.101191
Education_5	1.095359
MonthlyRate	1.060686
EducationField_Programming	1.053453
HourlyRate	1.048381
EducationField_Other	1.036640
JobLevel_3	0.998220
PerformanceRating	0.965902
Gamers_Yes	0.965540
Department_Technology Department	0.952730
JobRole_Manufacturing Director	0.929691
PercentSalaryHike	0.928456
JobRole_Developer	0.928256
DailyRate	0.910536
StockOptionLevel	0.901238
JobLevel_4	0.844269
JobRole_Manager	0.832301
JobRole_Customer Experience	0.830004
WorkLifeBalance	0.809963
TrainingTimesLastYear	0.784814
Age	0.762219
RelationshipSatisfaction	0.730955
JobSatisfaction	0.688886
YearsWithCurrManager	0.676306
YearsInCurrentRole	0.642713
EnvironmentSatisfaction_2	0.640535
JobInvolvement_2	0.618338
TotalWorkingYears	0.612071
Department_Human Resources	0.609457
JobRole_Designer	0.608828
JobRole_Research Director	0.607587
EnvironmentSatisfaction_3	0.604780
MonthlyIncome	0.540223
JobInvolvement_4	0.528745
EnvironmentSatisfaction_4	0.520814
JobLevel_2	0.482932
JobInvolvement_3	0.474881

Observaciones:

- Las probabilidades de que un empleado que hace horas extras se desgaste son 2,6 veces superiores a las de uno que no hace horas extras, probablemente porque hacer horas extras no es sostenible durante mucho tiempo para ningún empleado, y puede provocar agotamiento e insatisfacción laboral.
- Las probabilidades de que un empleado que viaja con frecuencia se desgaste son el doble que las de un empleado que no viaja tan a menudo.
- Las probabilidades de que los empleados solteros abandonen el trabajo son aproximadamente 1,85 veces (un 85% más altas) que las de un empleado con otro estado civil.

Curva de precisión/recuperación de la regresión logística

In [126]:
```
y_scores_lg = lg.predict_proba(x_train) # predict_proba nos da la probabilidad de cada observacion belonging to each class


precisions_lg, recalls_lg, thresholds_lg = precision_recall_curve(y_train, y_scores_lg[:, 1])

# Plot values of precisions, recalls, and thresholds
plt.figure(figsize = (10, 7))

plt.plot(thresholds_lg, precisions_lg[:-1], 'b--', label = 'precision', color = '#6161ff')

plt.plot(thresholds_lg, recalls_lg[:-1], 'g--', label = 'recall', color = '#f950a9')

plt.xlabel('Threshold')

plt.legend(loc = 'upper left')

plt.ylim([0, 1])

plt.show()
```



Observación:

- Podemos ver que la precisión y el recuerdo están equilibrados para un umbral de alrededor de 0,35.
- El treshold determina el punto en el que el modelo clasifica una observación como positiva o negativa. Al ajustar este valor, puede controlar el equilibrio entre precisión y recuperación. Un umbral más bajo dará lugar a un mayor número de predicciones positivas, lo que aumentará la recuperación pero reducirá potencialmente la precisión. Un umbral más alto dará lugar a menos predicciones positivas, lo que aumentará la precisión pero podría reducir la recuperación.
- En este caso, la observación indica que cuando el umbral se fija en torno a 0,35, los valores de precisión y recuperación (Recall) se equilibran. Esto significa que, con este valor de umbral, el modelo es capaz de identificar correctamente una buena proporción de los casos positivos reales, al tiempo que mantiene un alto nivel de precisión en sus predicciones positivas.

Averigüemos el rendimiento del modelo en este umbral.

In [127]:
```
optimal_threshold1 = .35

y_pred_train = lg.predict_proba(x_train)

metrics_score(y_train, y_pred_train[:, 1] > optimal_threshold1)
```

```
              precision    recall  f1-score   support

           0       0.93      0.93      0.93      1726
           1       0.65      0.64      0.65       332

    accuracy                           0.89      2058
   macro avg       0.79      0.79      0.79      2058
weighted avg       0.89      0.89      0.89      2058
```

Observaciones:

- El rendimiento del modelo ha mejorado. La recuperación ha aumentado significativamente para la clase 1.

Comprobemos el rendimiento con los datos de prueba (set data).

In [128]:
```
optimal_threshold1 = .35

y_pred_test = lg.predict_proba(x_test)

metrics_score(y_test, y_pred_test[:, 1] > optimal_threshold1)
```

```
              precision    recall  f1-score   support

           0       0.93      0.92      0.92       740
           1       0.60      0.62      0.61       142

    accuracy                           0.87       882
   macro avg       0.76      0.77      0.77       882
weighted avg       0.87      0.87      0.87       882
```

Observaciones:

- El modelo ofrece un rendimiento similar en los conjuntos de datos de prueba y de entrenamiento, es decir, ofrece un rendimiento generalizado.
- La recuperación de los datos de prueba ha aumentado y, al mismo tiempo, la precisión ha disminuido ligeramente, como era de esperar al ajustar el umbral.
- La recuperación y la precisión medias del modelo son buenas, pero veamos si podemos obtener un rendimiento aún mejor utilizando otros algoritmos.

K-Nearest Neighbors (K-NN)

K-NN utiliza características de los datos de entrenamiento para predecir los valores de nuevos puntos de datos, lo que significa que al nuevo punto de datos se le asignará un valor basado en lo similar que es a los puntos de datos del conjunto de entrenamiento.

¿Que debemos hacer?

1. Seleccionar K
2. Calcular la distancia (Euclídea, Manhattan, etc.)
3. Encontrar los K vecinos más cercanos
4. Votar por mayoría las etiquetas

La "K" en el algoritmo K-NN es el número de vecinos más cercanos entre los que queremos votar. Generalmente, K es un número impar cuando el número de clases es par, para obtener un voto mayoritario. Supongamos que K=3. En ese caso, haremos un círculo con el nuevo punto de datos como centro tan grande como encerrar sólo los tres puntos de datos más cercanos en el plano.

Pero antes de construir realmente el modelo, necesitamos identificar el valor de K que se utilizará en K-NN. Para ello realizaremos los siguientes pasos.

- Para cada valor de K (de 1 a 15), dividir el conjunto de entrenamiento en un nuevo conjunto de entrenamiento y validación (30 veces).
- Escalar los datos de entrenamiento y los datos de validación
- Calcula la media del error en los conjuntos de entrenamiento y validación para cada valor de K.
- Represente gráficamente el error medio de entrenamiento frente al de validación para todos los K.
- Elegir el valor óptimo de K en el gráfico para que los dos errores sean comparables.

In [139]:

```
knn = KNeighborsClassifier()

# We select the optimal value of K for which the error rate is the least in the validation data
# Let us loop over a few values of K to determine the optimal value of K

train_error = []

test_error = []

knn_many_split = {}

error_df_knn = pd.DataFrame()

features = X.columns

for k in range(1, 15):
    train_error = []

    test_error = []

    lista = []

    knn = KNeighborsClassifier(n_neighbors = k)

    for i in range(30):
        x_train_new, x_val, y_train_new, y_val = train_test_split(x_train, y_train, test_size = 0.20)

        # Fitting K-NN on the training data
        knn.fit(x_train_new, y_train_new)

        # Calculating error on the training data and the validation data
        train_error.append(1 - knn.score(x_train_new, y_train_new))

        test_error.append(1 - knn.score(x_val, y_val))

    lista.append(sum(train_error)/len(train_error))

    lista.append(sum(test_error)/len(test_error))

    knn_many_split[k] = lista

knn_many_split
```

Out[139]:

```
{1: [0.0, 0.09555016181229771],
 2: [0.05516403402187121, 0.1457928802588997 2],
 3: [0.05838396111786149, 0.1580097087378641 6],
 4: [0.11814499797488862, 0.1521035598705502 1],
 5: [0.109437019036047, 0.1519417475728156],
 6: [0.13063993519643577, 0.1505663430420712 3],
 7: [0.1228230052652896, 0.1530744336569579 7],
 8: [0.13681652490887, 0.1542071197411003 2],
 9: [0.13140947752126367, 0.1545307443365696],
 10: [0.1445727014985824, 0.1502427184466020 3],
 11: [0.13997569866342646, 0.1506472491909385 5],
 12: [0.14746861077359255, 0.1557443365695793 4],
 13: [0.14400567031186715, 0.1459546925566343 5],
 14: [0.14868367760226814, 0.1548543689320388 8]}
```

PROYECTO EN PROCESO. TODAVÍA NO FINALIZADO.