APÉNDICE
IBM HR EMPLOYEE ATTRITION

CoderHouse Data Science

7 de Diciembre, 2021

Del Signore, Vittoria
Faya, Tamara
Lamas, Guadalupe Ayelén

Tabla de contenido

1 Apéndice:

Se resumen a continuación los resultados del notebook.

1.1 Análisis Univariado

Se estudiaron las variables por medio de histogramas, para comprender cómo es la distribución de cada una de ellas y poder detectar anomalías. Este análisis sólo puede realizarse para las variables numéricas, de tipo ordinales. Los gráficos obtenidos se muestran en la Figura 6.1.1.

Se observó que las variables DailyRate, MonthlyRate y HourlyRate muestran datos tipo "meseta", mientras que el resto de variables (Age, DistanceFromHome, MonthlyIncome, PercentSalaryHike, TotalWorkingYears, YearsAtCompany, YearsInCurrentRole, YearsSinceLastPromotion, YearsWithCurrManager) son asimetricos a derecha, teniendo en algunos casos más de un pico.

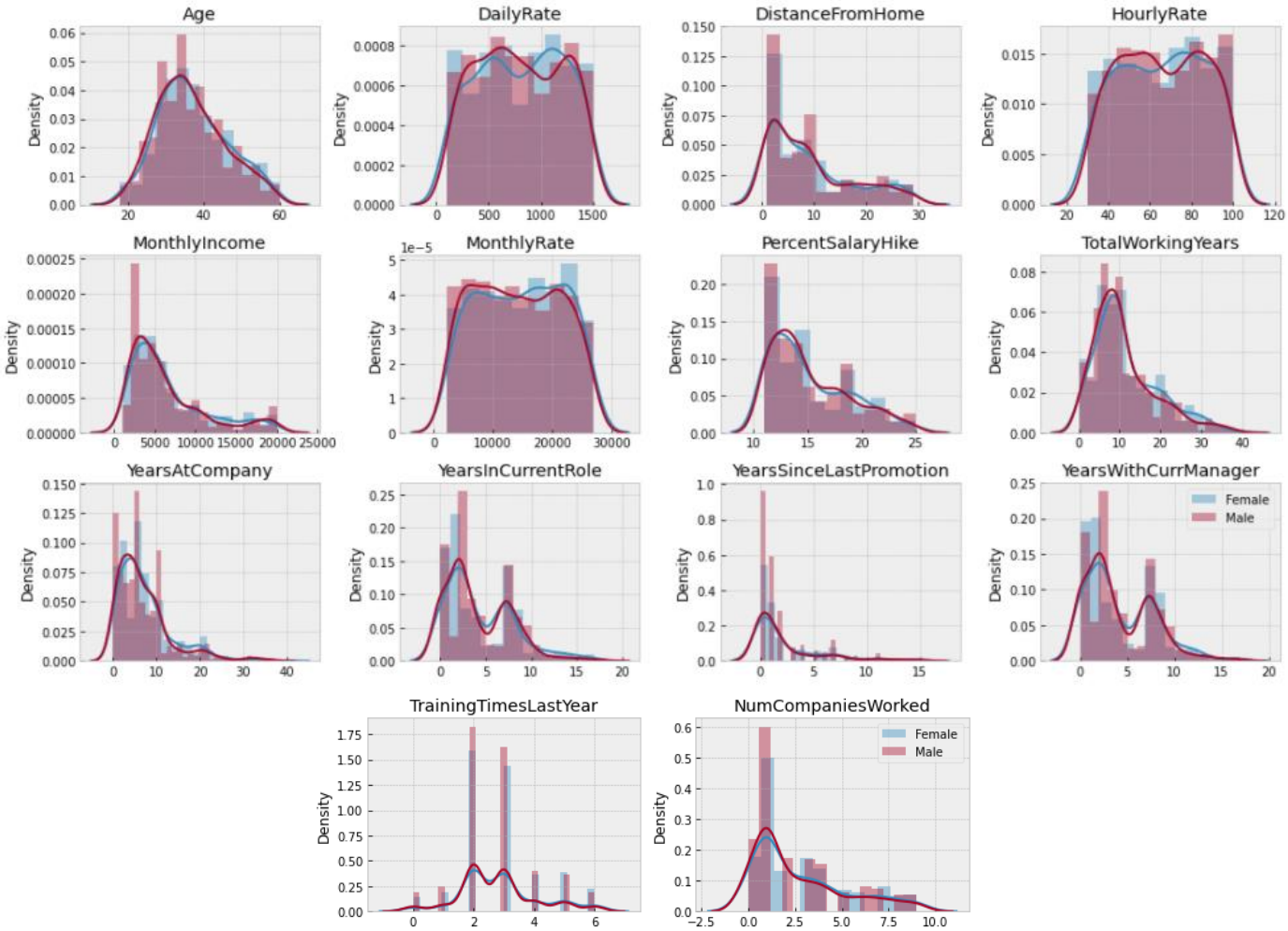
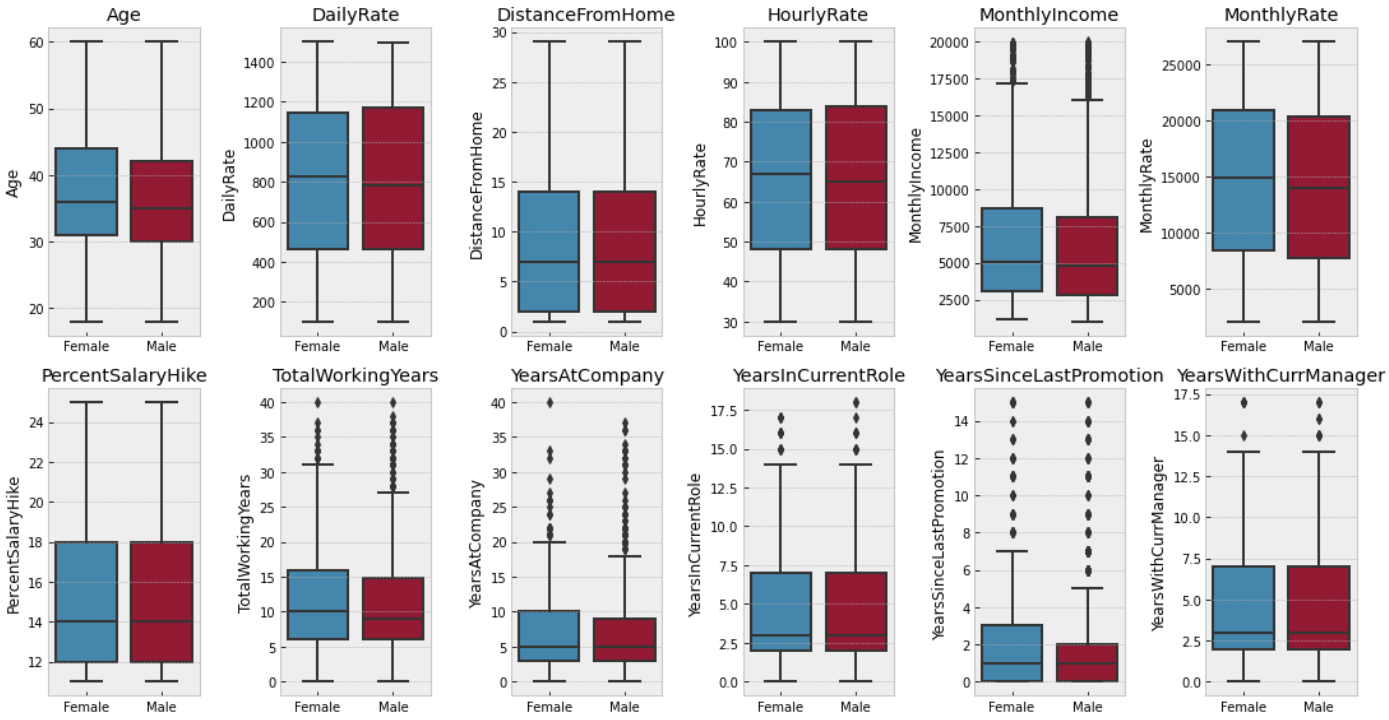


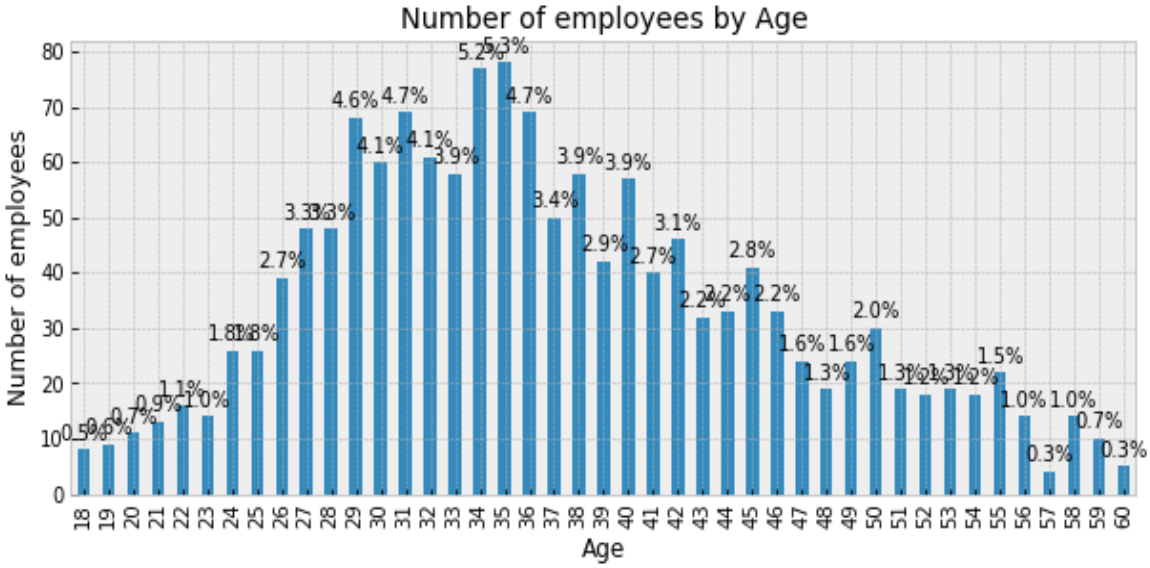
Figura 6.1.1 Distribuciones de las variables ordinales del data set.

Se estudió la presencia de outliers mediante gráficos tipo boxplot de estas variables y si bien parecían tener outliers, se consideró que responden a valores razonables del dataset, por lo que se decidió no excluirlos del estudio.

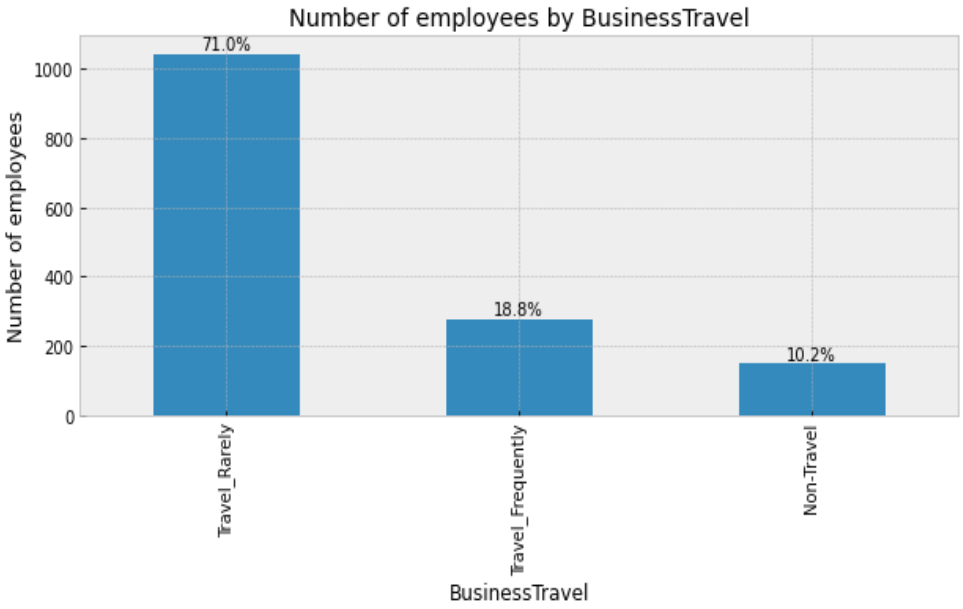
Se realizaron gráficos de caja y bigote para las mismas variables, pero ahora discriminando por género. En general, el comportamiento es similar para ambos géneros:



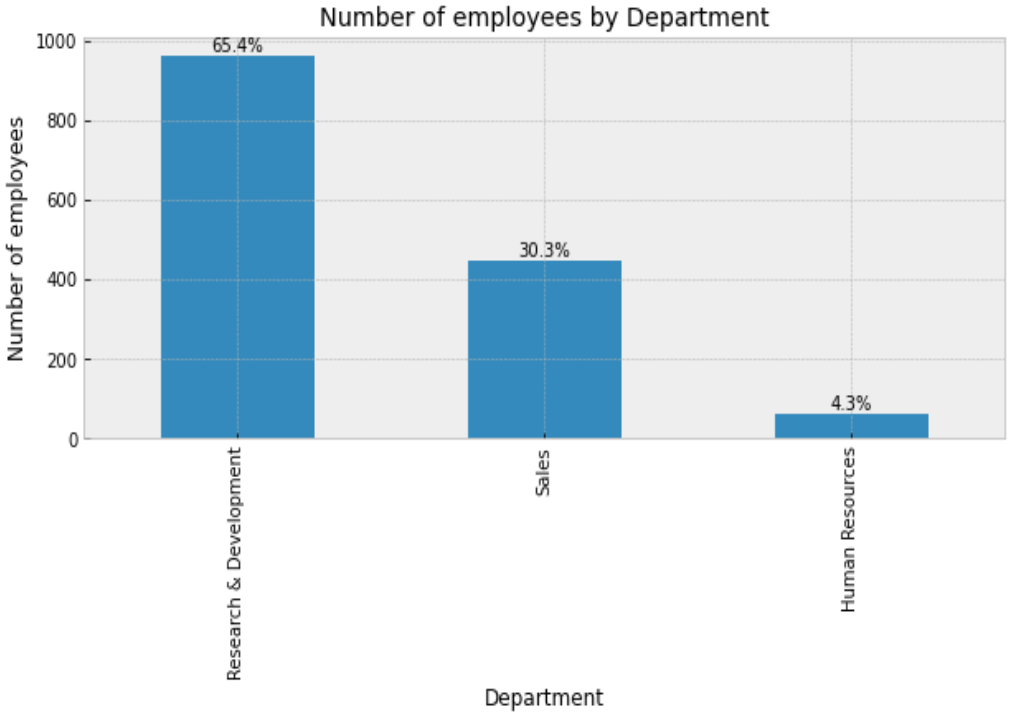
Los empleados evaluados van de 18 a 60 años, en donde la mayor cantidad de empleados tienen 34-35 años, y la minoría de empleados se encuentran en los extremos, es decir pocos empleados muy jóvenes (18-21 años) o mayores (55-60 años):



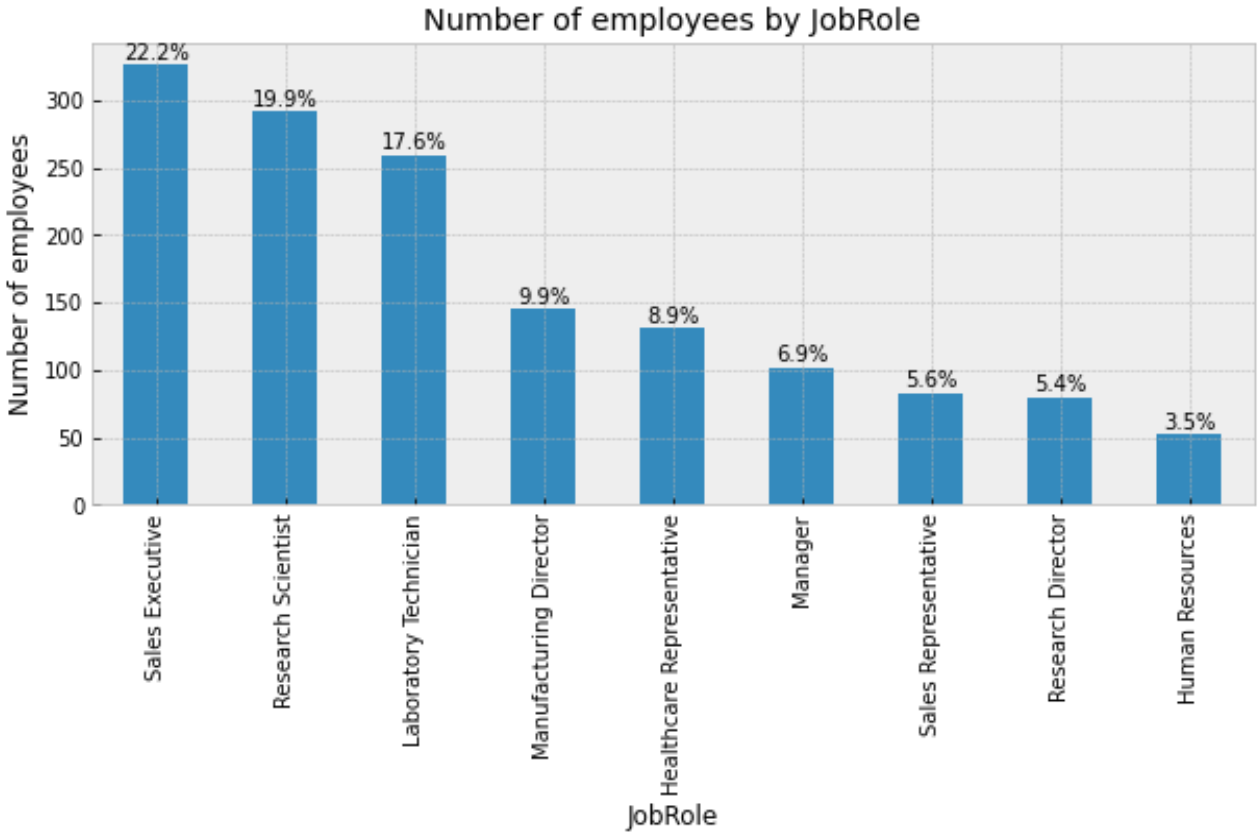
La mayoría de empleados no viajan (10,2%) o lo hacen muy rara vez (81,0%), y solo un pequeño grupo viaja frecuentemente (18,8%):



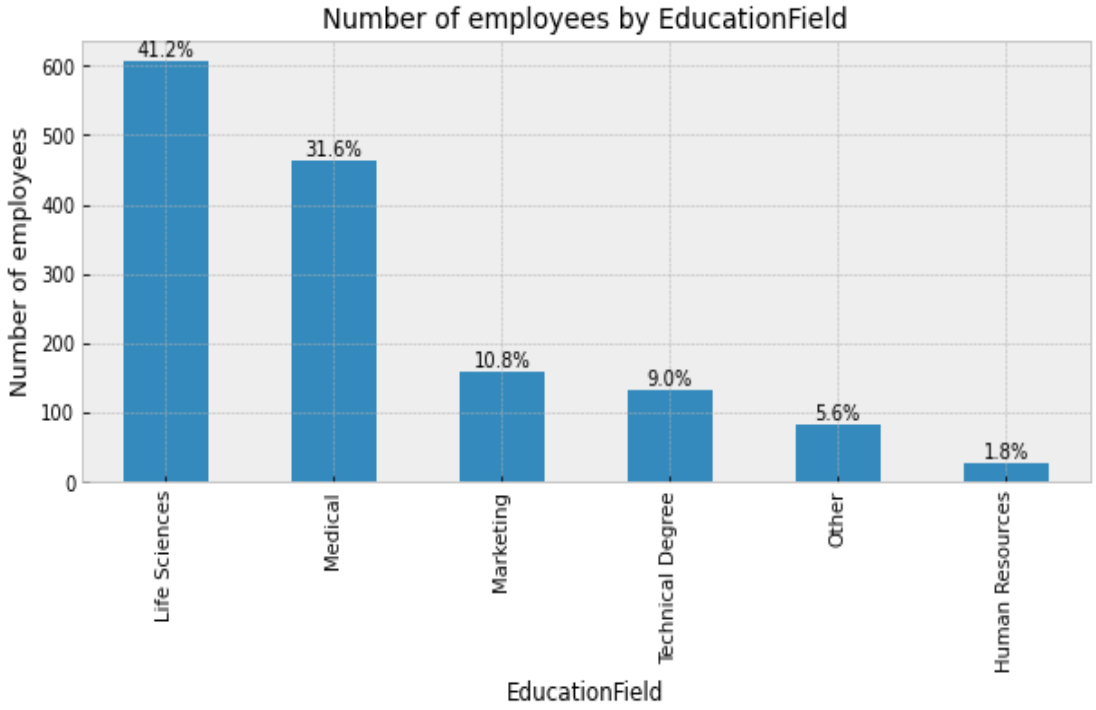
El 65,4% están en el departamento de Investigación y Desarrollo, el 30,3% se dedican a la venta de productos, y el 4,3% restante son personal relacionado a recursos humanos:



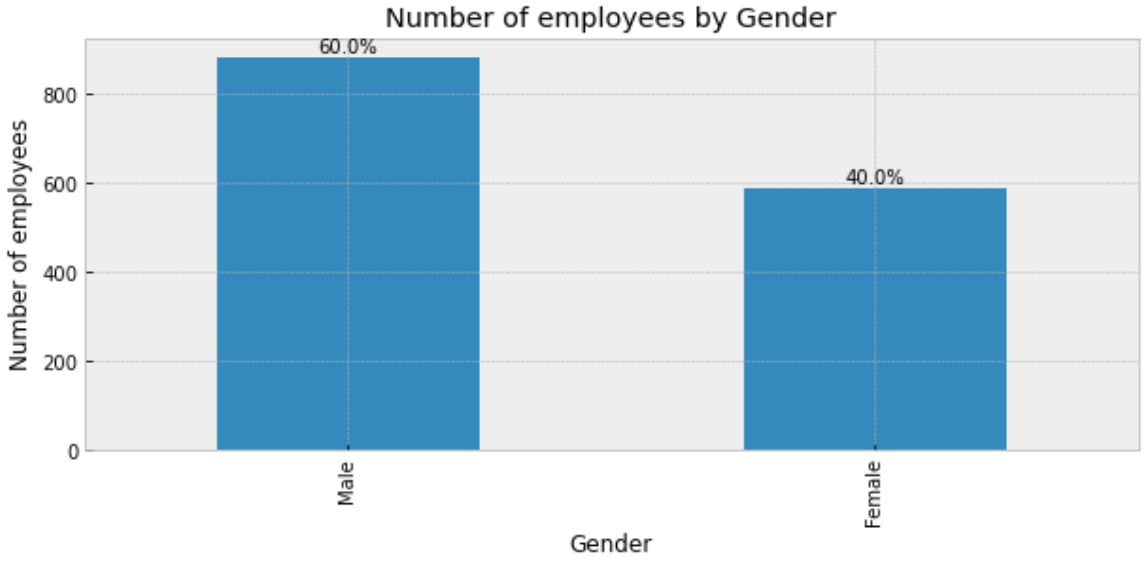
Los roles de los empleados estaban concentrados en 3 grupos: 22,2% son Ejecutivos de Ventas, 19,9% son Científicos de Desarrollo, y 17,6% son Técnicos de Laboratorio. A continuación se muestra el detalle con el porcentaje de todos los roles:



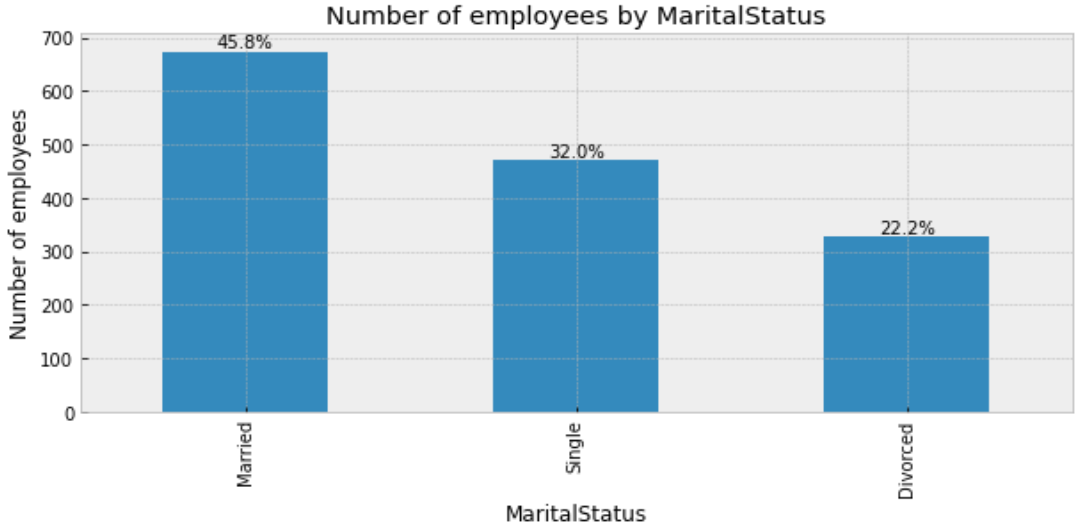
En cuanto al campo de educación, se distinguen 5 grupos y el resto se agrupa en "Otros". La mayoría estudiaron carreras relacionadas con Ciencias de la Vida y Medicina, y estos dos grupos hacen el 72,8%:



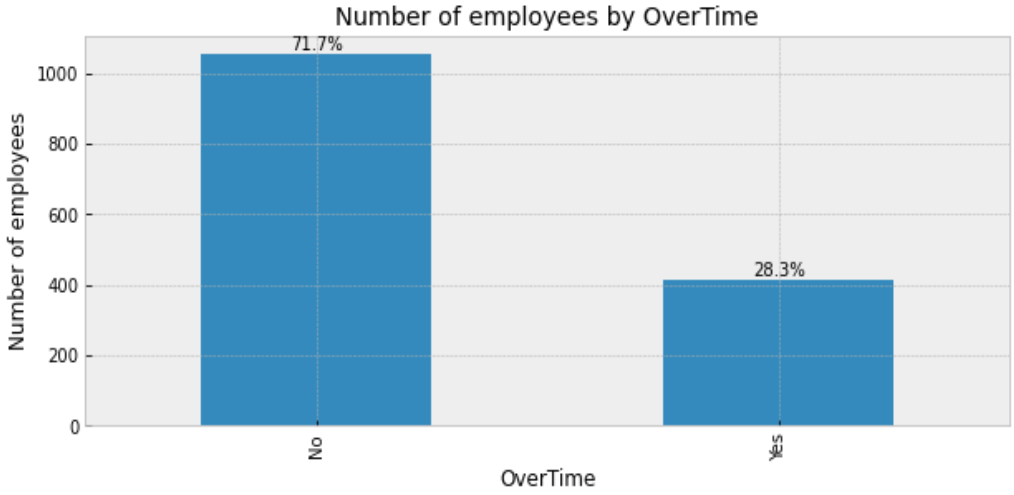
En lo que respecta al género, el 60% son hombre, y el 40& son mujeres:



El estado civil muestra que el 45,8% de los empleados están casados, 32,0% son solteros, y el 22,2% restante son divorciados:

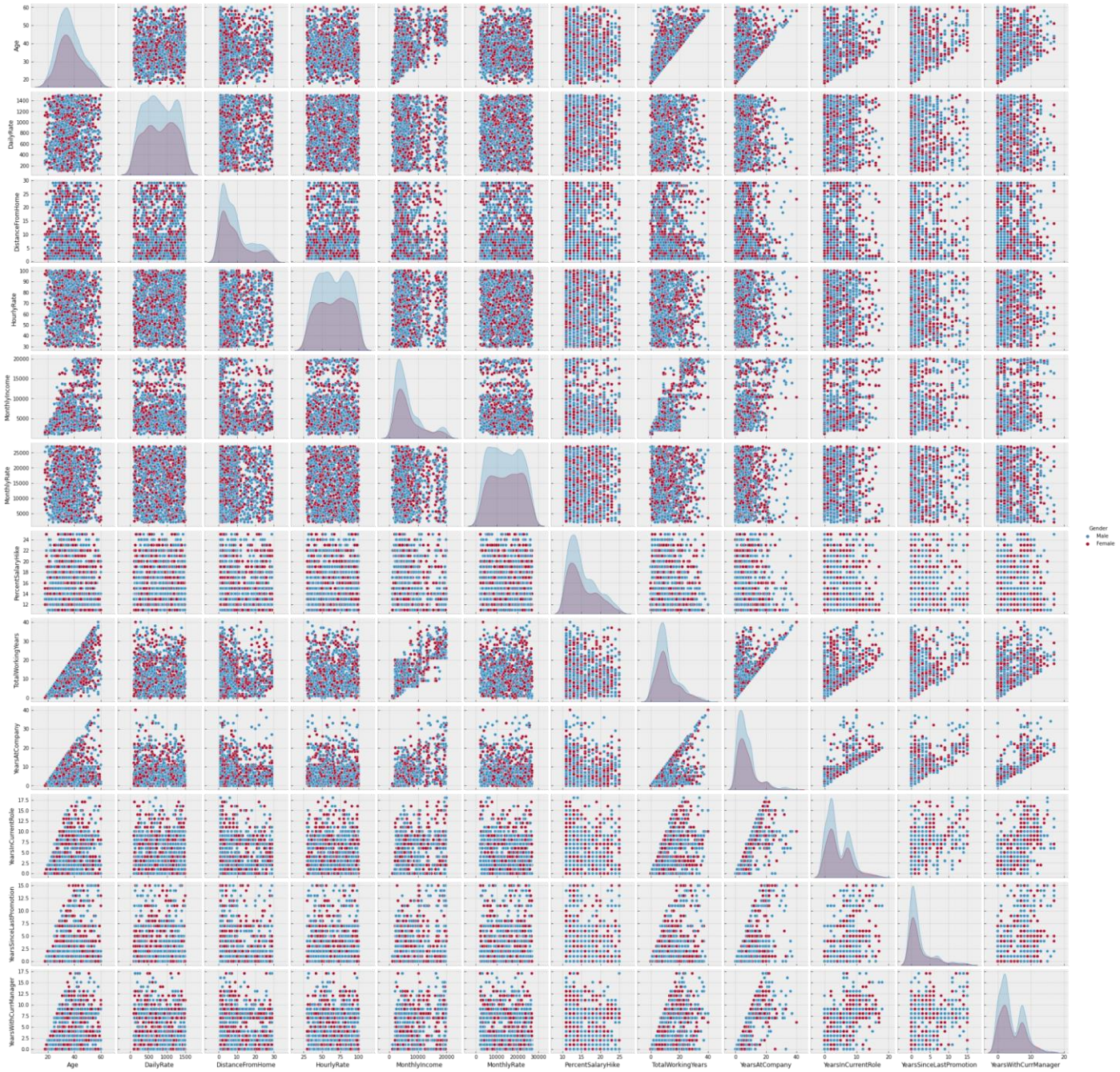


Si evaluamos los casos donde los empleados trabajan horas adicionales a las que les corresponde, el 71,7% indica no trabajar horas extra; pero el 28,3% restante indica haber trabajado tiempo adicional:

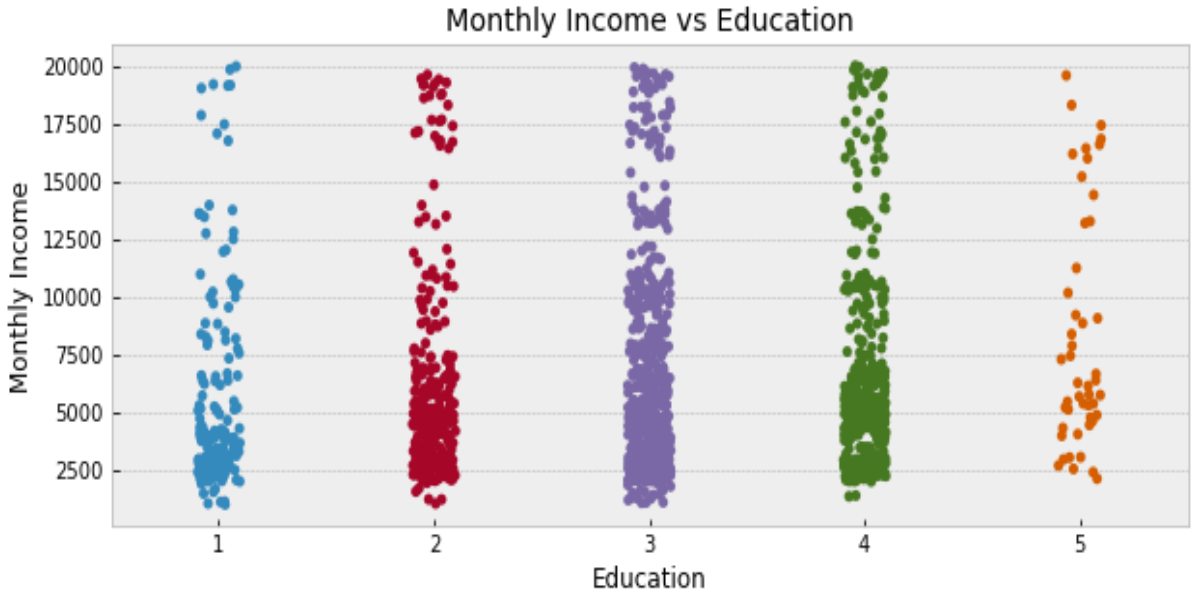


1.2 Análisis Bivariado y Multivariado

A continuación se muestra un pairplot que combina cada variable del dataset:

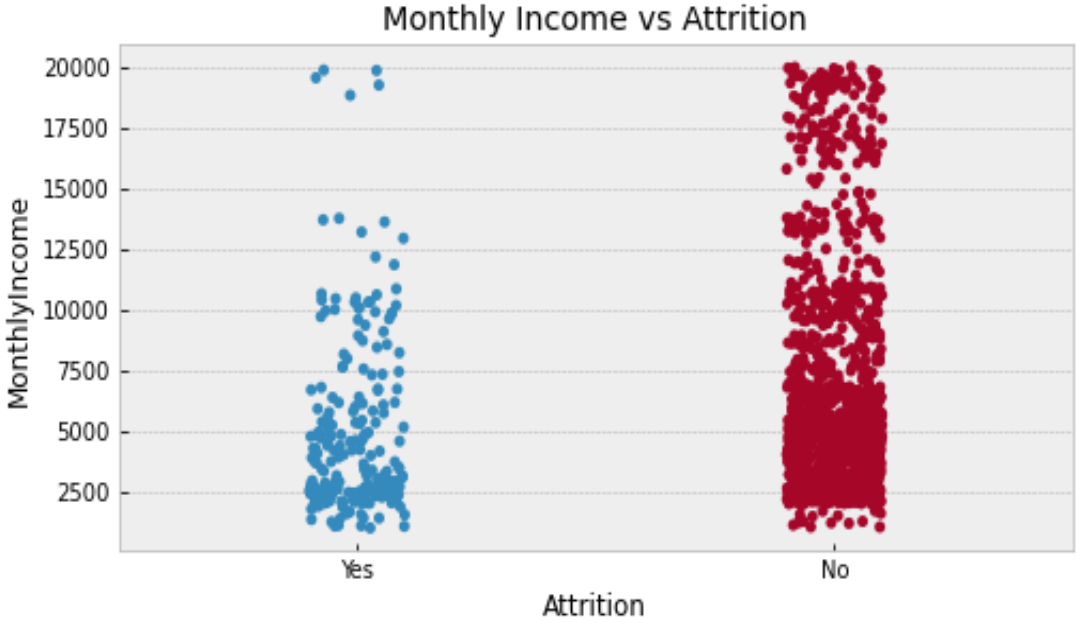


Se combinaron las variables de Monthly Income y Education, y se puede observar que la mayor cantidad de empleados se encuentran concentrados en el grupo 3 (bachelor). En cada grupo de educación se puede ver que el monthly income puede variar de una persona a otra, y que por lo general en todos los grupos, la mayor concentración se encuentra en monthly income más bajos (principalmente en los primeros 2 grupos):

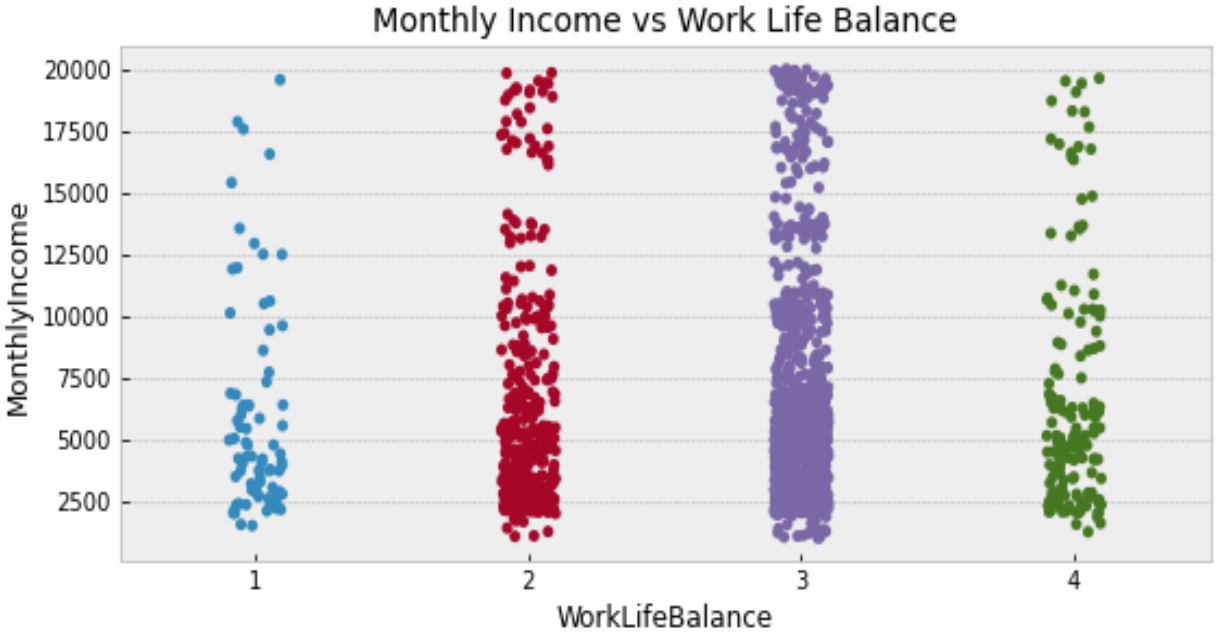


Al evaluar el JobInvolvement vs. el MonthlyIncome, la mayor parte de empleados se concentra en los grupos 2 y 3; y al igual que en el gráfico anterior, en todos los grupos, la mayoría de empleados tienen un menor monthly income:

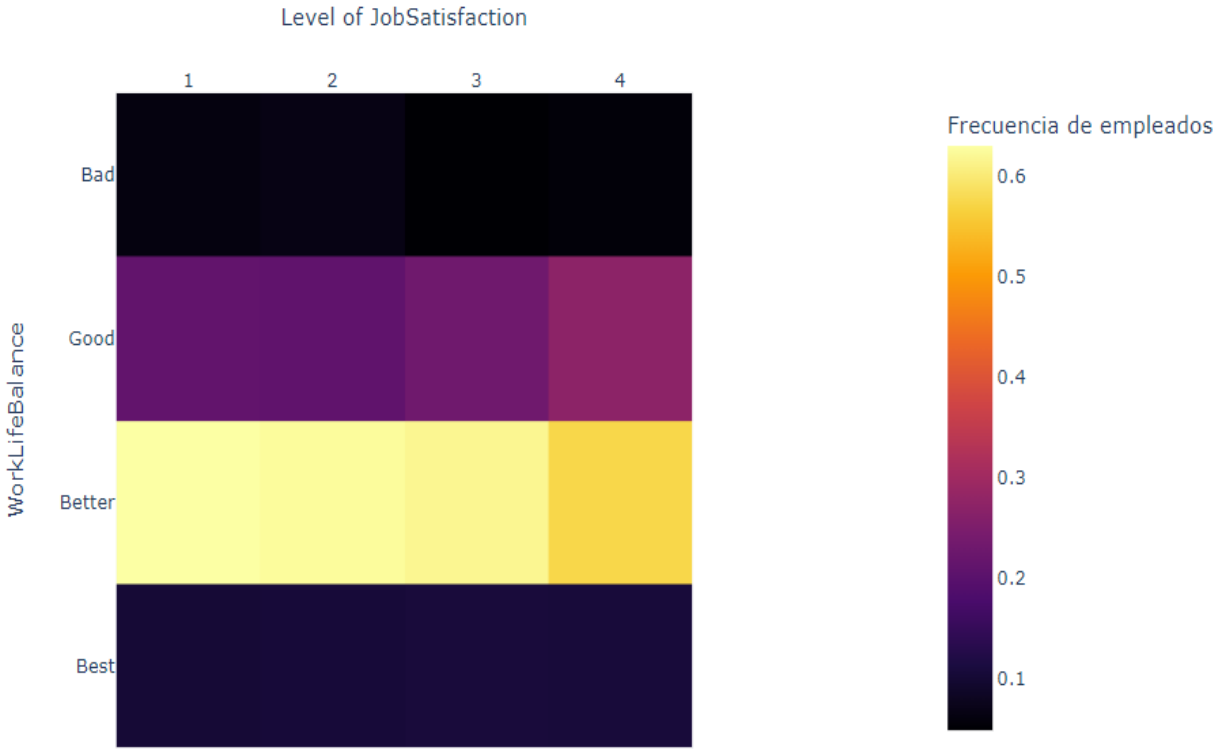
En el attrition los casos donde se ve el mayor desgaste laboral suele ser en aquellos donde los salarios son más bajos pareciendo esta ser una variable a mirar en mayor detalle:



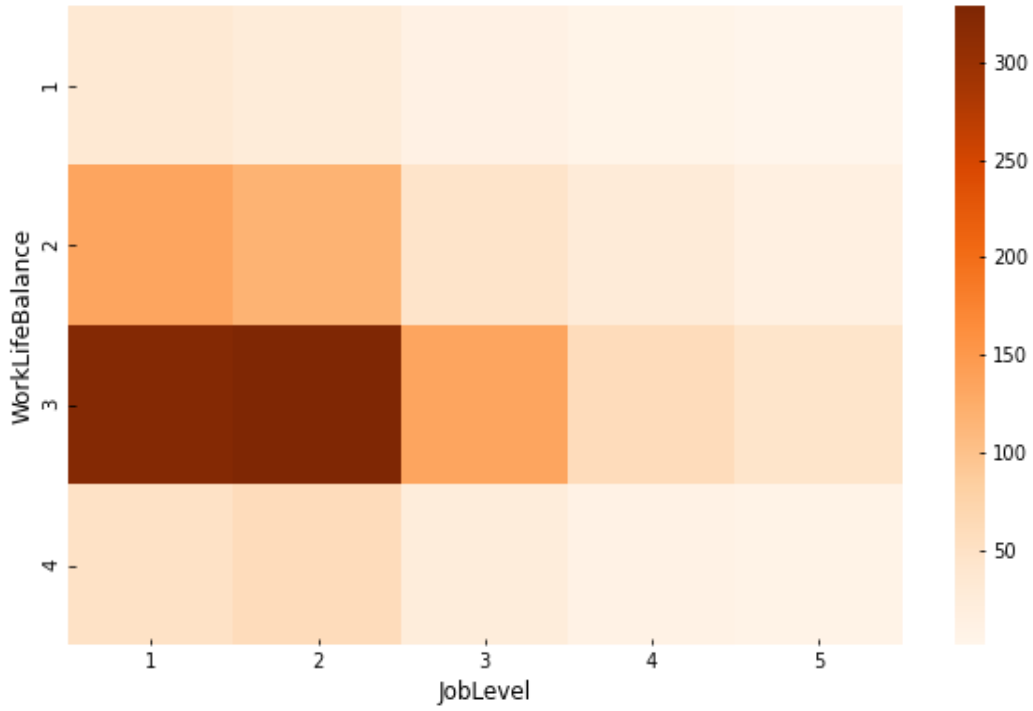
En este caso observamos que no hay una relación proporcional entre ingresos y el balance de vida y trabajo de un empleado. El balance de vida y trabajo a simple vista parece buena:



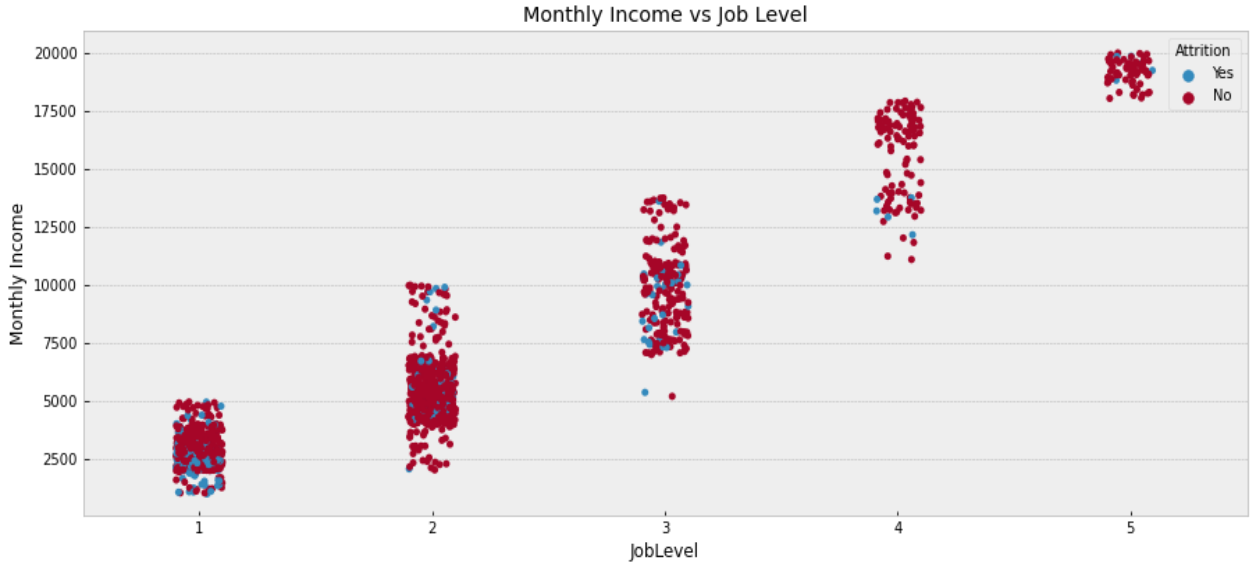
Evaluando el JobSatisfaction con WorkLifeBalance, la mayoría de empleados indican tener un buen balance:



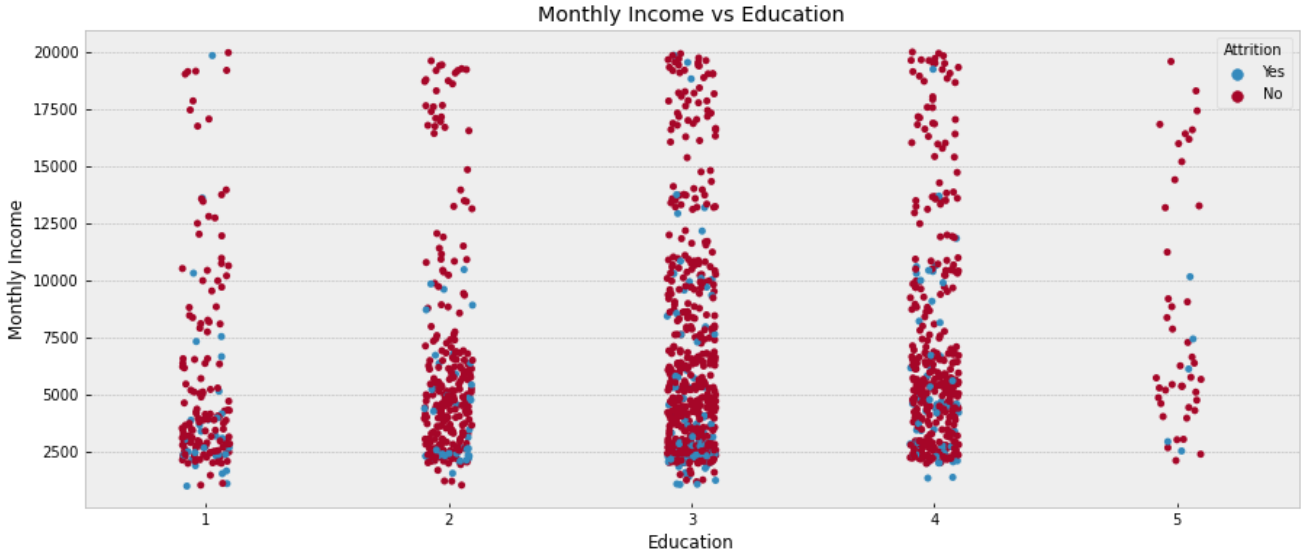
Relacionando el WorkLifeBalance y JobLevel, la mayor concentración de empleados pertencen a niveles bajos en IBM, e indican tener un balance de nivel 3:



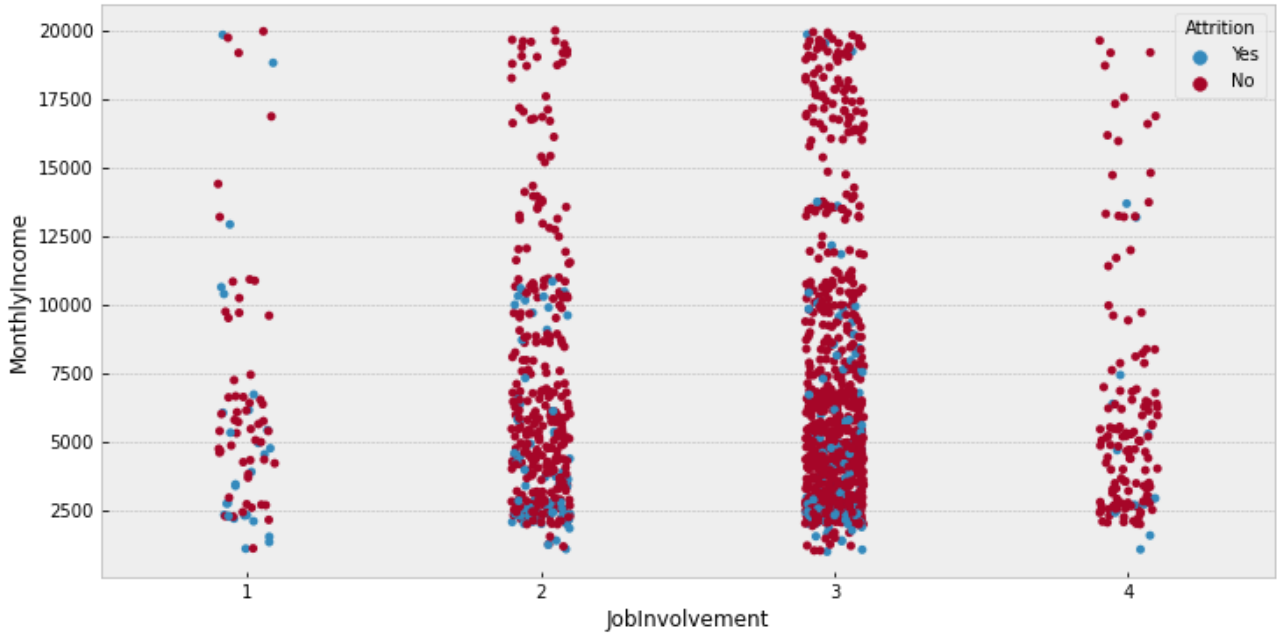
De la mano del gráfico anterior, al abrir el detalle de Job Level por Monthly Income discriminando por la variable attrition, seguimos viendo mayor concentración en los niveles más bajos, pero a medida que sube en dignos niveles, los ingresos mensuales son más elevados, con pocos casos atípicos. Vemos que hay empleados con desgaste laboral en todos los niveles de la organización, pero se concentran más en el nivel más bajo, es decir, bajo Job Level y bajo Monthly Income:



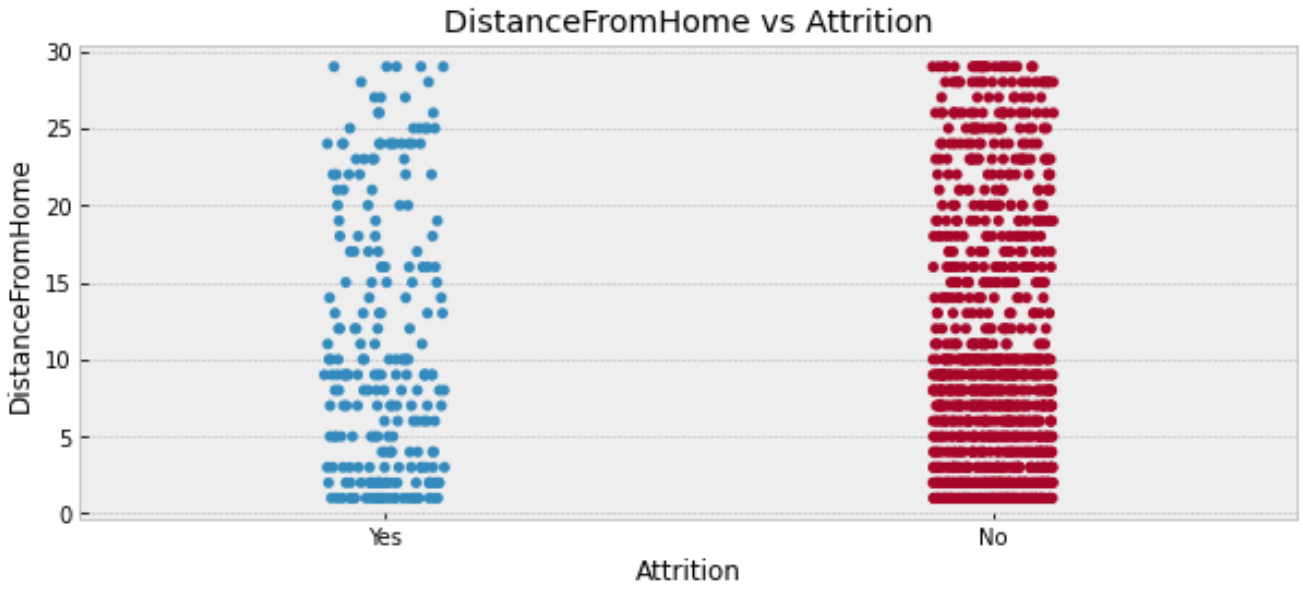
Cuando pasamos a un análisis similar al anterior, pero ahora por niveles de educación; vemos que el desgaste se concentra en empleados de ingresos mensuales bajos, independientemente de su nivel académico:



El attrition está presente en los 4 niveles de implicación laboral, y también aplica en personas con diferentes ingresos mensuales. Sin embargo, la mayor concentración de casos se da en los niveles de menores ingresos mensuales:

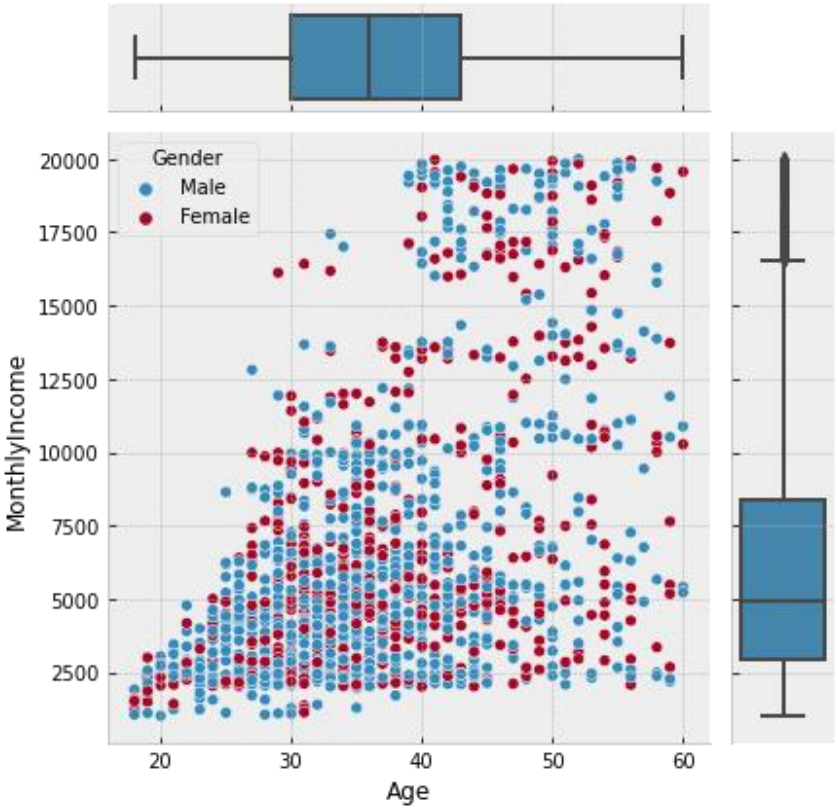


La variable attrition también se cruzó con la variable de distancia entre el trabajo y la casa, y los empleados con desgaste están a diferentes distancias; es decir, puede renunciar al trabajo el que vive lejos, pero también el que vive cerca. De hecho, la mayoría de casos se encuentran concentrados a baja distancia;

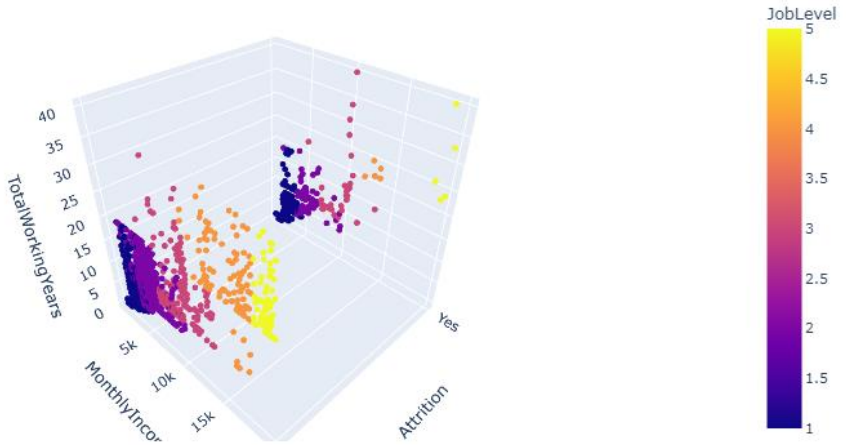


En los 3 gráficos anteriores, se pudo observar que independientemente del nivel laboral, la educación y el involucramiento laboral, por lo general, los empleados con desgaste laboral están en las franjas más bajas de salario mensual.

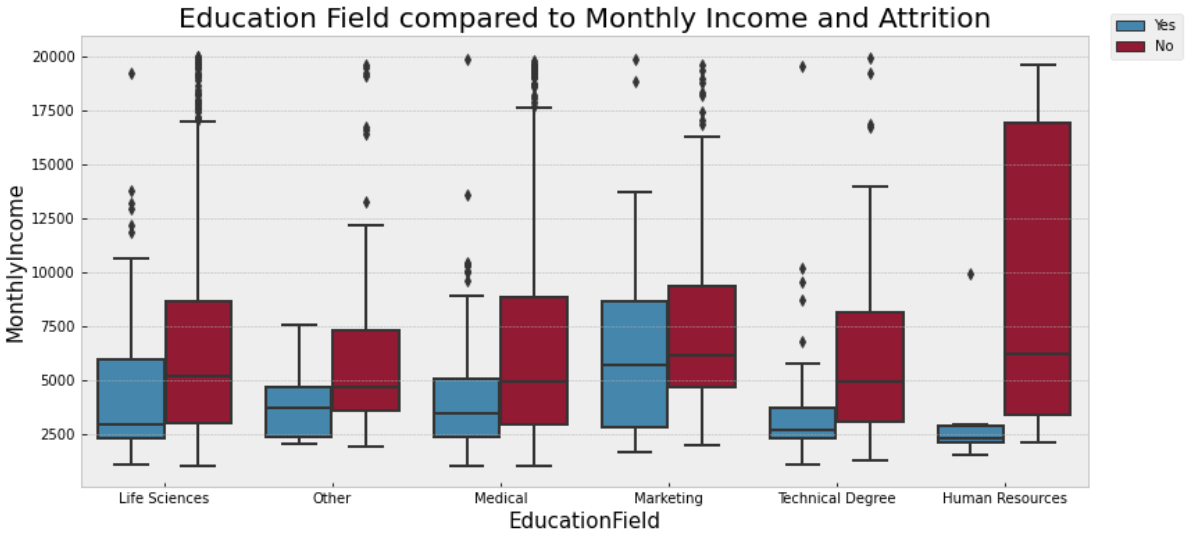
Si evaluamos estas variables, pero ahora por edad, se aprecia que las personas más jóvenes tienen un nivel de ingreso inferior al promedio, lo cual ocurre debido a que están en los niveles más bajos de Job Level (1 o 2). Conforme las personas avanzan en edad, por lo general, también se incrementa su Job Level, y por lo tanto, también sus ingresos. Sin embargo, la relación no es del todo lineal, ya que hay personas que aunque tienen mayor edad, pueden permanecer en los Job Level bajos a lo largo de su vida profesional, lo que los hace tener ingresos no tan elevados:



Vemos que las personas que tienen mayor cantidad de años trabajando, tienen por una parte mayores ingresos mensuales, pero también a mayor experiencia laboral y cargos más elevados. Entonces, estas 3 variables (edad, nivel laboral e ingresos mensuales) nos permiten ver clusters bien definidos en IBM. Al separar por la variable Attrition, vemos que hay una tendencia similar, solo que con menor cantidad de casos de personas que indican tener desgaste.

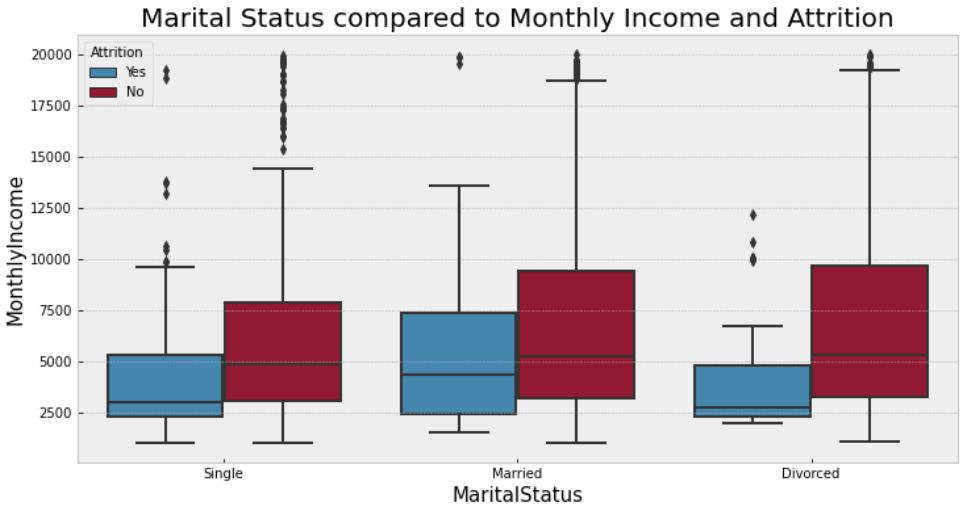


Podemos observar que el tema de los ingresos mensuales si los analizamos desde el punto de vista del campo de educación también es homogéneo en el sentido que en cada campo de educación se comporta igual. Con esto descartamos que hubiese algo atípico, es decir que hubiese attrition en algún campo de estudio con ingresos mensuales mayores a aquellos que no poseen attrition:



Si evaluamos por departamento, vemos que los empleados que indican tener desgaste, tienen ingresos por lo general más bajos a aquellos que indican no presentar desgaste dentro de su mismo grupo. Por ejemplo, en el caso de Recursos Humanos, los empleados con desgaste tienen una concentración de salarios cuyo máximo casi coincide con el inicio del segundo cuartil para el grupo sin desgaste, y si lo vemos a nivel de medianas, los que tienen desgaste cobran alrededor de 3000 USD, cuando el grupo de attrition no cobra alrededor de 5000 USD:

En los 3 grupos de estados maritales de los empleados, aquellos que indican tener desgaste laboral, tienen por lo general menor ingreso mensual.



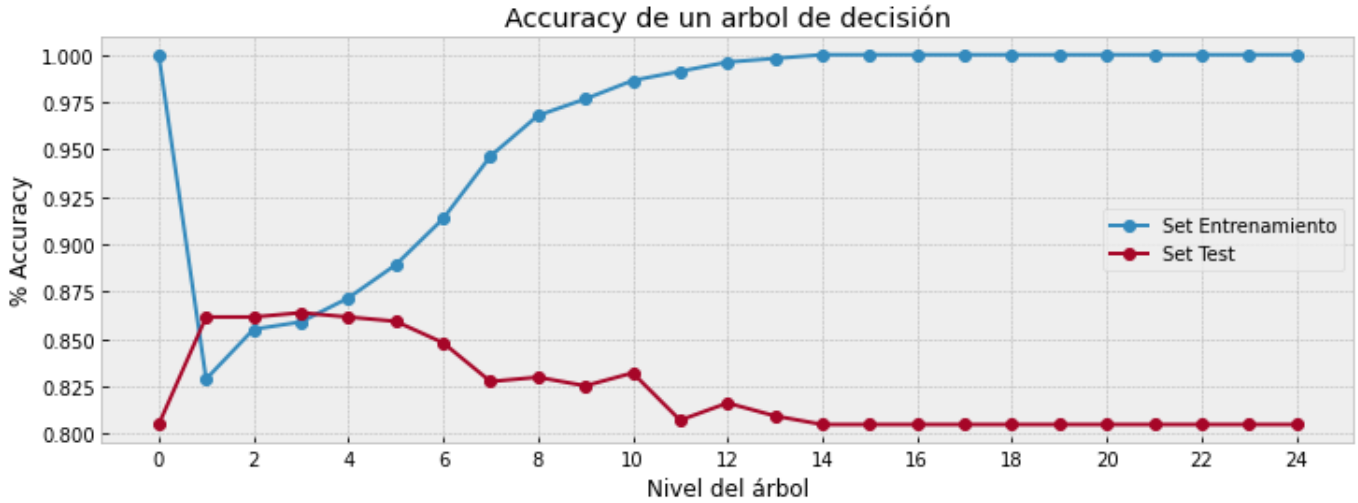
Estos últimos 3 gráficos, donde se muestran las variables con la apertura del attrition, indican que, por lo general los empleados con desgaste se agrupan en los rangos de bajos salarios, independientemente de su educación, departamento, estado civil.

1.3 Modelos con dataset Train/Test 70/30

A continuación se muestran los modelos aplicados, parámetros elegidos y gráfico del ROC en cada caso:

1.3.1 Árboles de Decisión

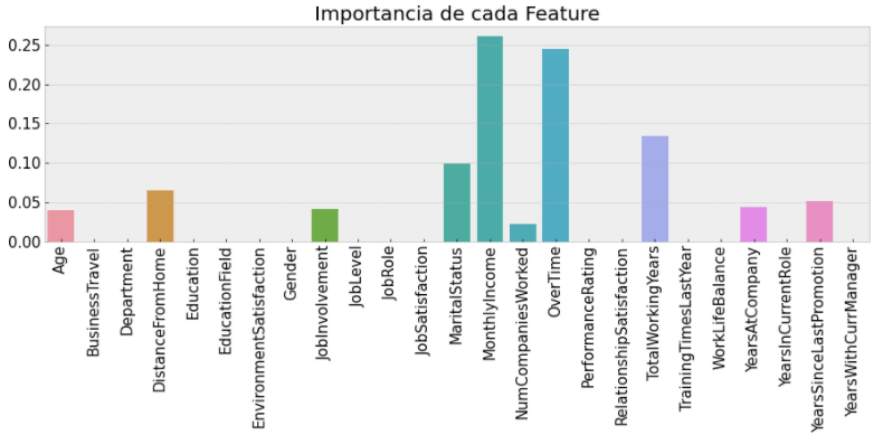
<u>Análisis de profundidad del árbol:</u>
Se realizó un análisis de la métrica Accuracy y la profundidad de los árboles de decisión:

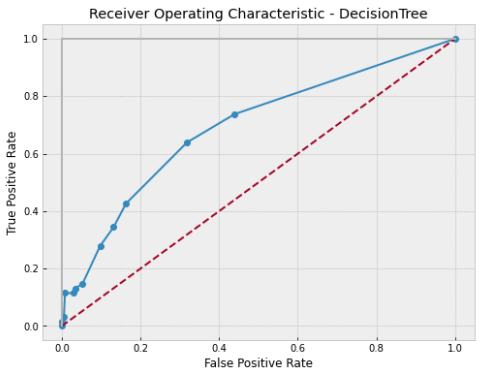


En el grafico se observa como varía el accuracy en función de la profundidad que le damos al arbol. Se observa que una profundidad de 3 o 4 es un valor razonable, antes de empezar a perder accuracy en el set de test y posiblemente empezar a overfitear el set de entrenamiento.

Árbol de decisión 1:
Arbol_de_decisión_1 = DecisionTreeClassifier()
max_depth = 4, random_state = 42

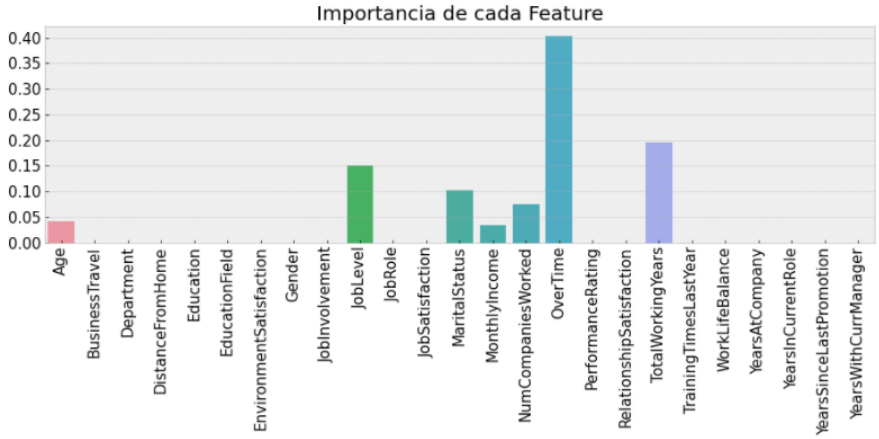
Se muestran a continuación la importancia o peso de cada variable (feature) y la curva roc asociada al modelo:

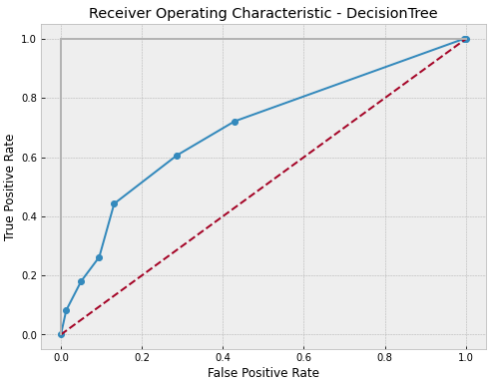




Árbol de Decisión 2:
Arbol_de_decisión_2 = DecisionTreeClassifier()
max_depth = 3, random_state = 42, class_weight='balanced'

Se muestran a continuación la importancia o peso de cada variable (feature) y la curva roc asociada al modelo:





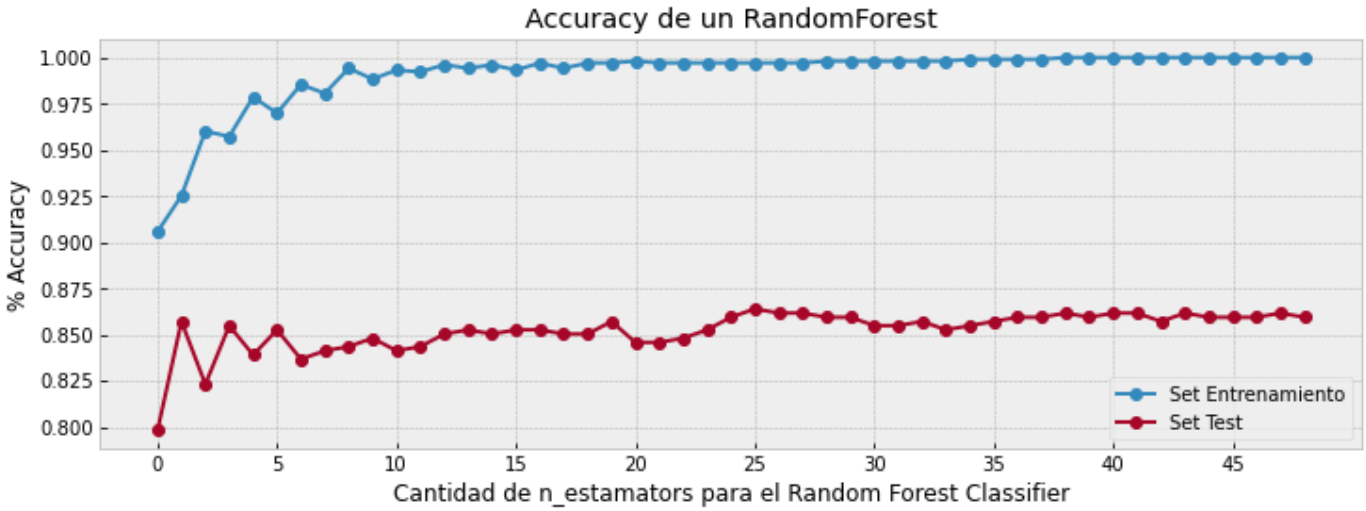
Resultados obtenidos de los árboles de decisión:

	arbol_1	arbol_2
Accuracy	0.861678	0.809524
Precision	0.500000	0.350650
Recall	0.114750	0.442620
ROC_curve	0.691390	0.696270

1.3.2 Random Forest:

Análisis de Cantidad de estimadores:

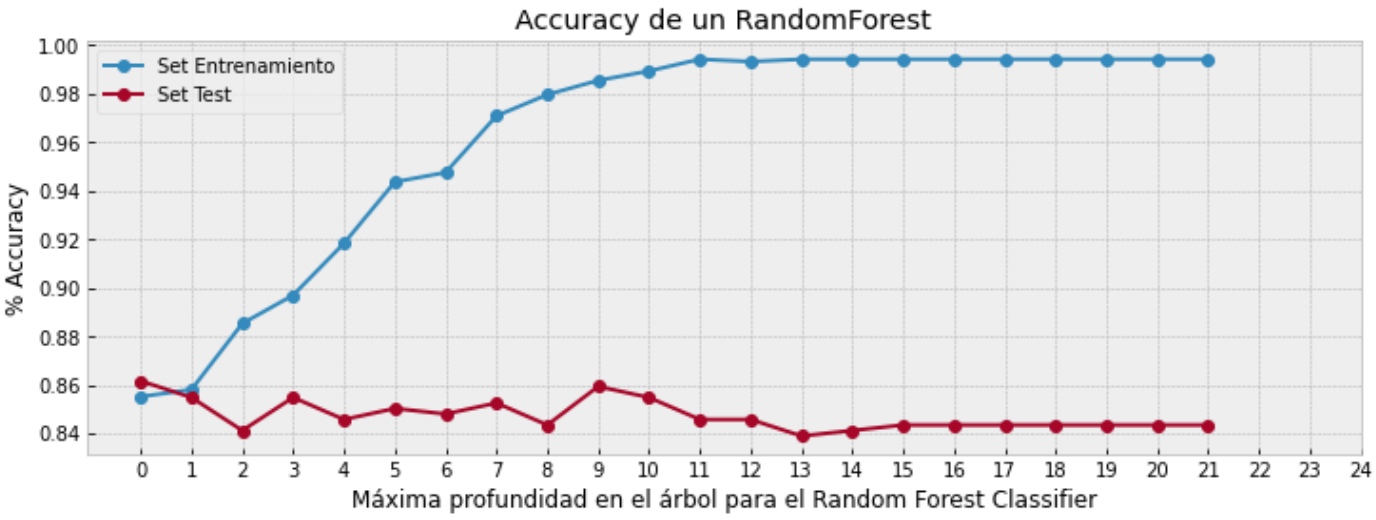
Se realizó un análisis del Accuracy respecto a la cantidad de estimadores de los árboles de decisión que componen los bosques (forest):



Se observa del gráfico que más allá de 9 árboles en el bosque, no mejora el accuracy y se observa overfiting en el set de entrenamiento.

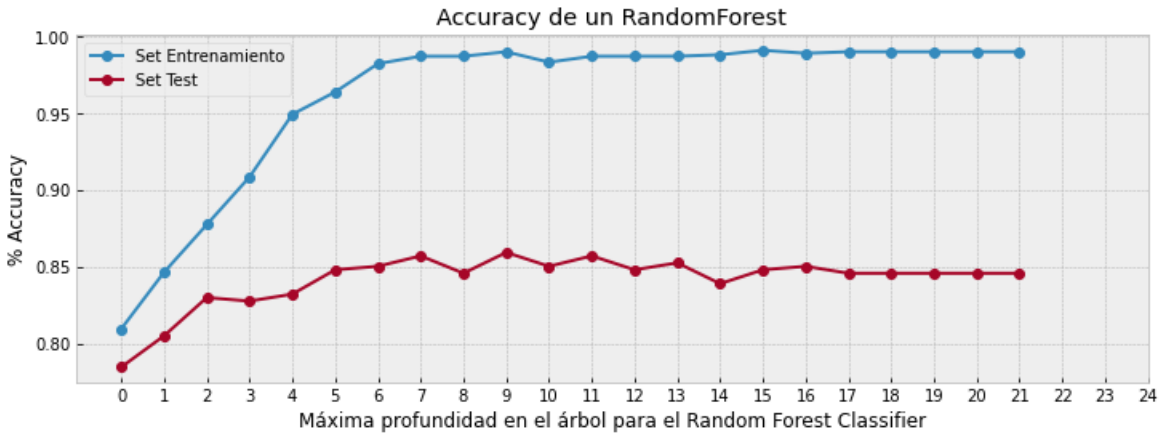
Análisis de profundidad del árbol:

Se seleccionó n_estimators = 9 y se realizó el análisis de la profundidad del arbol.



El gráfico muestra una rápida divergencia del accuracy entre el set de Test y Train con el aumento de la profundidad de los árboles. Teniendo en cuenta que la clase analizada se encuentra desbalanceada, se hizo uso del parámetro class_weight="balanced" y se realizó nuevamente el análisis del accuracy:

n_estimators = 9 + class_weight="balanced"

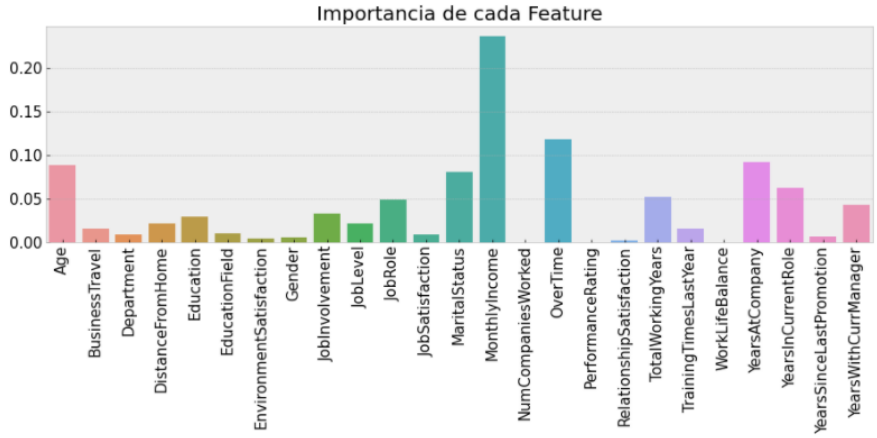


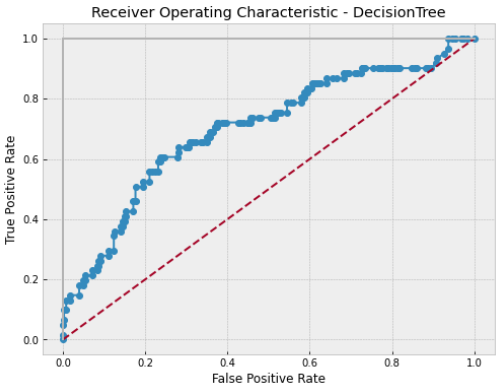
Se consideró un buen resultado en accuracy, los parámetros n_estimators = 9, max_depth =3 y class_weight="balanced".

Se generaron dos modelos random forest, con y sin class_weight="balanced". Se muestran a continuación la importancia o peso de cada variable (feature) y la curva roc asociada al modelo:

Random Forest 1:

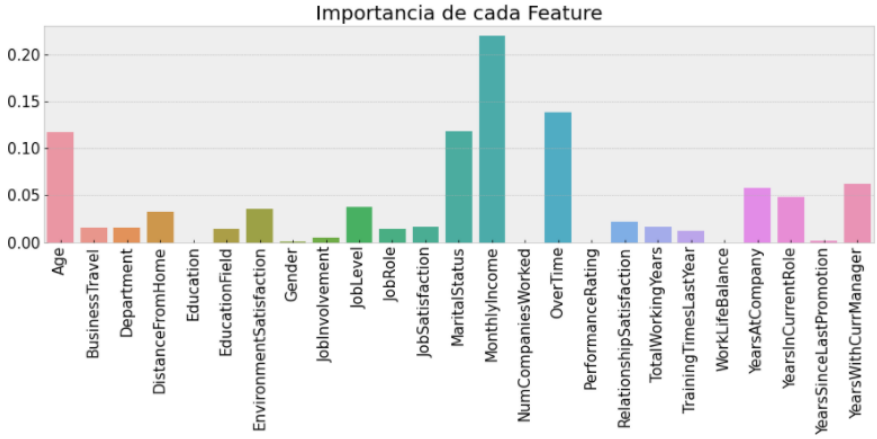
random_forest_model = RandomForestClassifier
n_estimators=9, max_depth =3, random_state=42, max_features="log2"

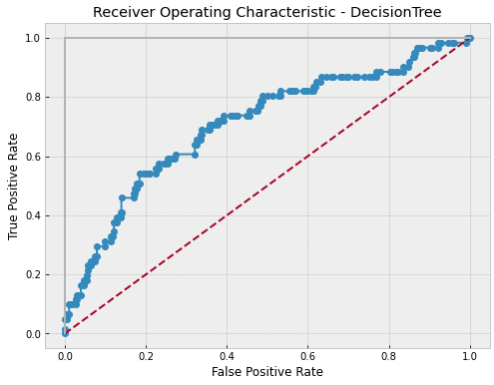




Random Forest 2:

random_forest_model = RandomForestClassifier
n_estimators=9, max_depth =3, random_state=42, max_features="log2"
class_weight="balanced"





Resultados obtenidos de los random forest:

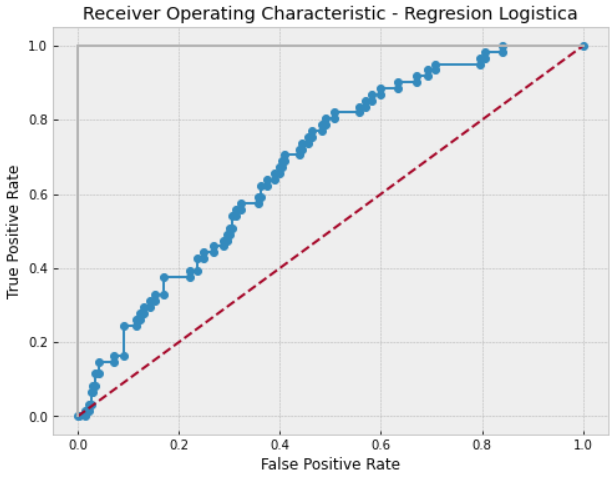
	forest_1	forest_2
Accuracy	0.863946	0.759637
Precision	1.000000	0.297300
Recall	0.016390	0.016390
ROC_curve	0.700630	0.707940

1.3.3 Regresión Logística

Se realizó un modelo de regresión logística con parámetros por default.

regresion_logistica = LogisticRegression()

Se muestran la curva roc obtenida y los resultados de las métricas estudiadas:



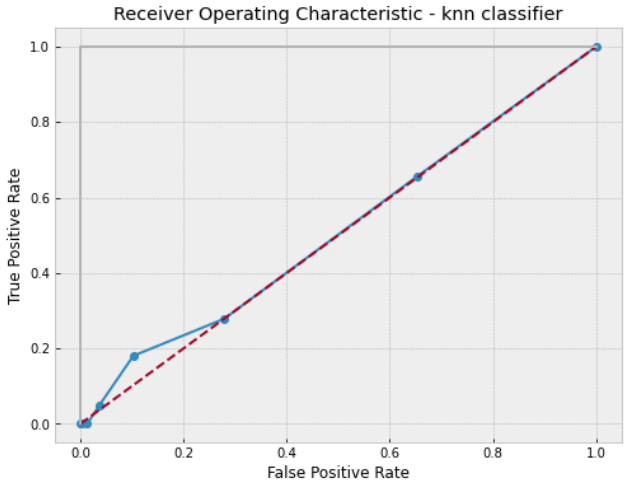
	LogReg
Accuracy	0.843537
Precision	0.277780
Recall	0.081970
ROC_curve	0.686500

1.3.4 K-Nearest-Neighbors (Knn):

Se realizó un modelo de Knn con los siguientes parámetros propuestos:

classifier = KNeighborsClassifier()
n_neighbors = 7, metric = 'minkowski', p = 5

Se muestran la curva roc obtenida y los resultados de las métricas estudiadas:



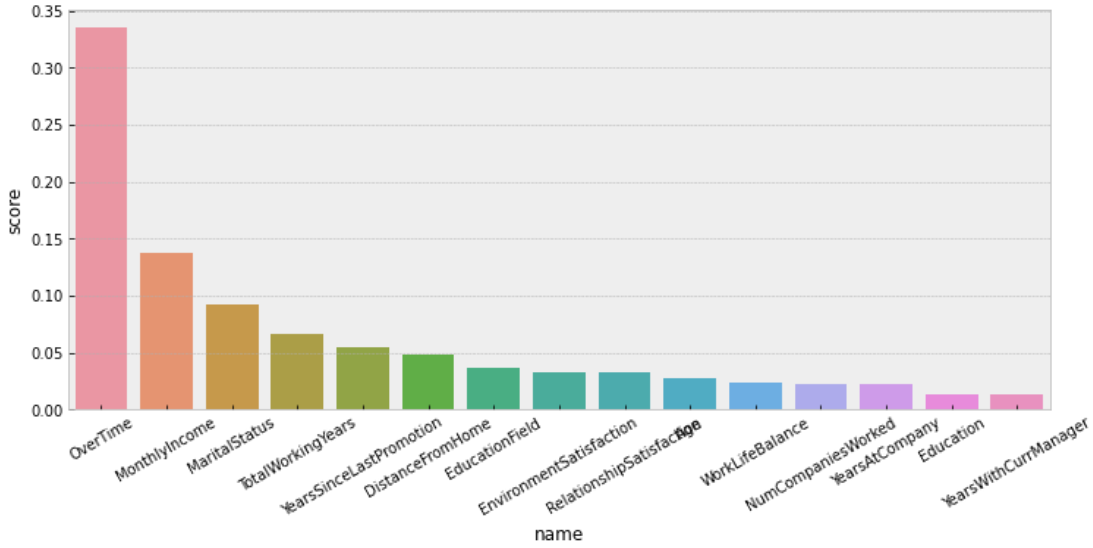
	knn
Accuracy	0.836735
Precision	0.176470
Recall	0.049180
ROC_curve	0.510760

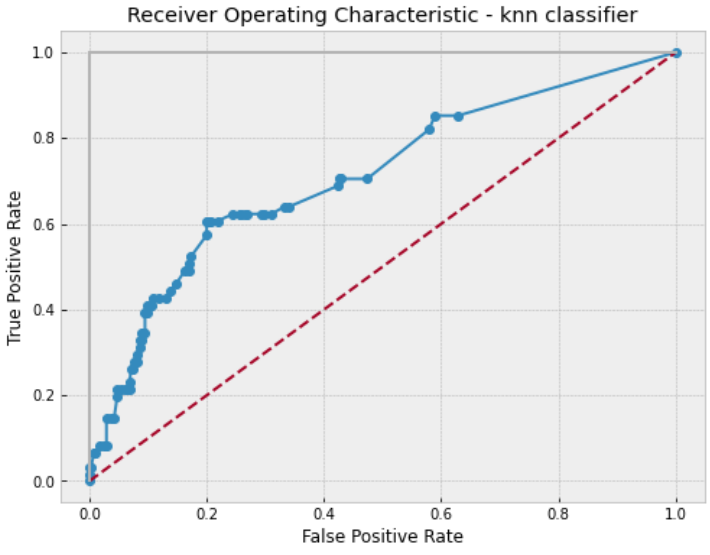
1.3.5 XGBoost:

Se realizó un modelo XGBoost con los siguientes parámetros propuestos:

xgb_model = xgboost.XGBClassifier()
objective='binary:logistic', n_estimators=10, seed=42, max_depth=6, learning_rate=0.01

Se muestran a continuación la importancia o peso de cada variable (feature) y la curva roc asociada al modelo:





	XGboost
Accuracy	0.836735
Precision	0.351350
Recall	0.213110
ROC_curve	0.710960

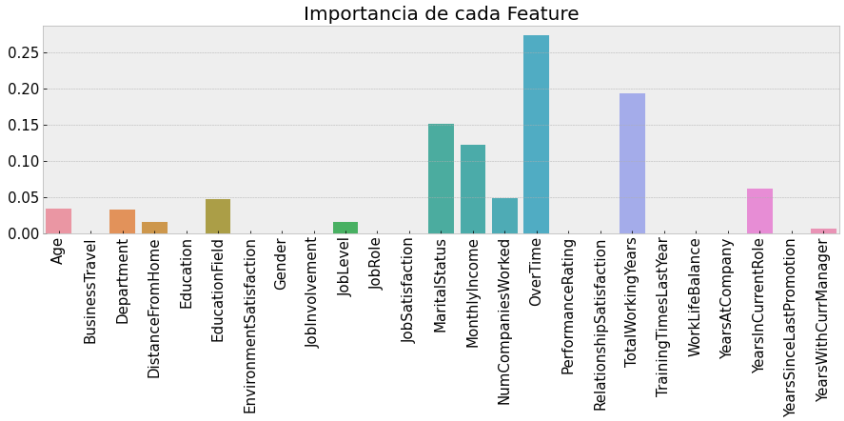
1.4 Modelos con dataset Train/Test y Oversampling

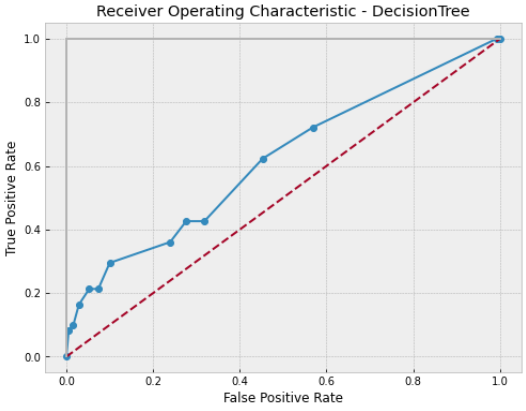
Se volvieron a ejecutar los modelos aplicando la técnica de **oversampling** al dataset para evaluar sus resultados.

1.4.1 Árboles de Decisión

Árbol de decisión 1 con oversampling:

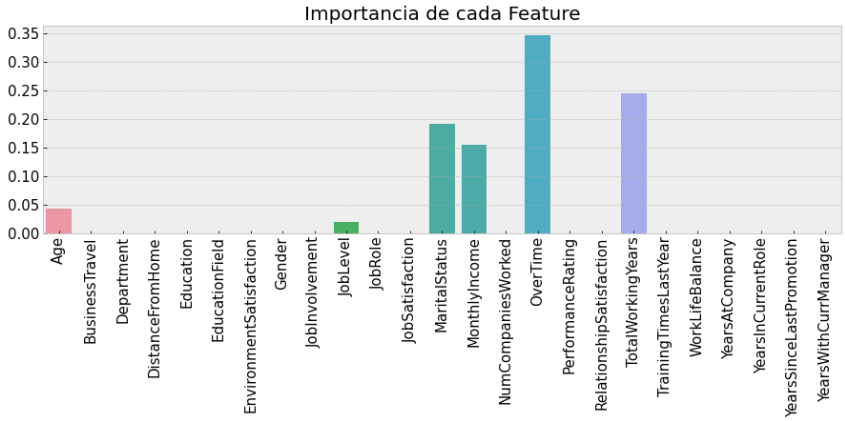
arbol_de_decision = DecisionTreeClassifier()
max_depth=4, random_state = 42





Árbol de decisión 2 con oversampling:

arbol_de_decision = DecisionTreeClassifier()
max_depth=4, random_state = 42, class_weight='balanced'



	arbol_1b	arbol_2b
Accuracy	0.646259	0.811791
Precision	0.176870	0.333330
Recall	0.426230	0.360660
ROC_curve	0.617470	0.641780

1.4.2 Random Forest:

No se ejecutaron estos modelos con la separación del dataset con oversampling. Queda pendiente para trabajos futuros.

1.4.3 Regresión Logística

Se ejecutó el mismo modelo con la separación Train/Test con oversampling.

regresion_logistica = LogisticRegression()

Se presenta la curva roc y resultados obtenidos:

	LogReg_b
Accuracy	0.673469
Precision	0.221480
Recall	0.540980
ROC_curve	0.641780

1.4.4 KNN

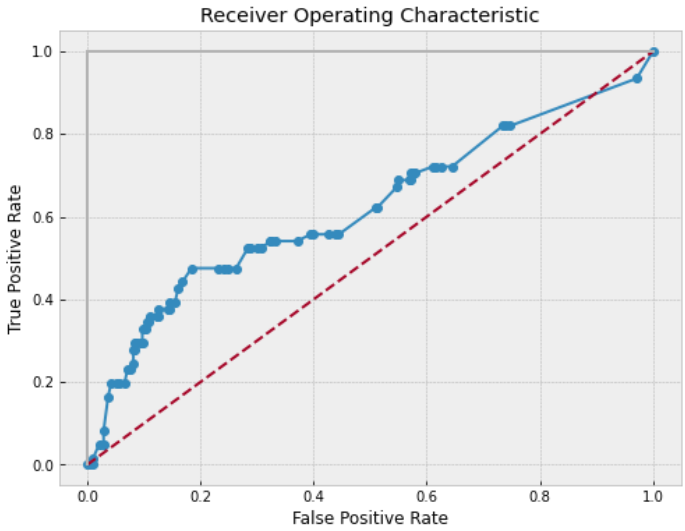
No se ejecutó este modelo con la separación del dataset con oversampling. Queda pendiente para trabajos futuros.

1.4.5 XGBoost:

Se realizó un modelo XGBoost con los mismos parámetros propuestos previmanete, teniendo en cuenta la separación Train/Test con oversampling:

xgb_model_b = xgboost.XGBClassifier()
objective='binary:logistic', n_estimators=10, seed=42, max_depth=6, learning_rate=0.01

Se mustra la curva roc y los resultados obtenidos:



	XGboost_b
Accuracy	0.786848
Precision	0.296300
Recall	0.393440
ROC_curve	0.620530

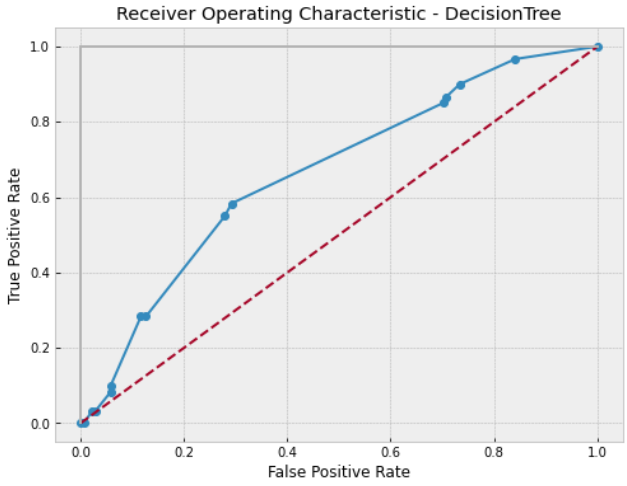
1.5 Modelos con optimización de hiperparámetros

Se seleccionaron 3 modelos para los cuales se corrió una optimización de parámetros utilizando StratifieldFold y SearchGrid.

1.5.1 Árboles de Decisión

Rango de parámetros de búsqueda:

```
'max_depth'     : list(np.arange(2, 11, step=1)),
'criterion'     :['gini', 'entropy'],
'splitter'      :['best', 'random'],
'max_features'  : ['auto', 'sqrt', 'log2'],
'ccp_alpha'     : list(np.arange(0.0, 1., step=0.05))
```



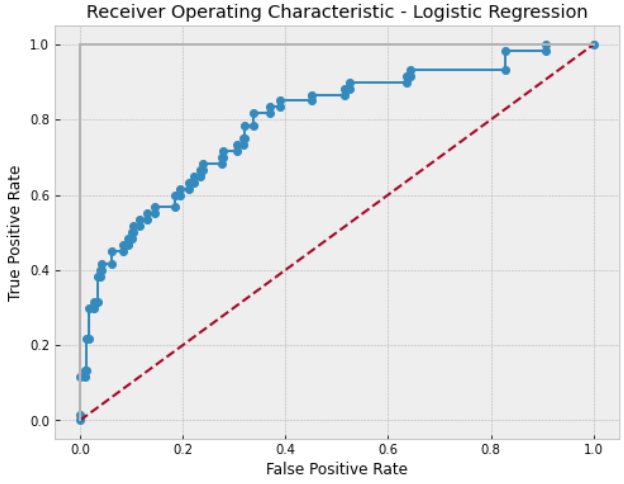
PARAMETROS OPTIMIZADOS:
arbol_de_decision = DecisionTreeClassifier()

```
random_state = 42,
criterion    = 'entropy',
splitter     = 'random',
max_depth    = 5,
max_features ='auto',
ccp_alpha    = 0.0
```

1.5.2 Regresión Logística

Rango de parámetros de búsqueda:

```
'penalty' : ['l1','l2','elasticnet','none'],
'solver':['newton-cg','lbfgs','liblinear','sag','saga'],
'max_iter':[100,200,300]
```



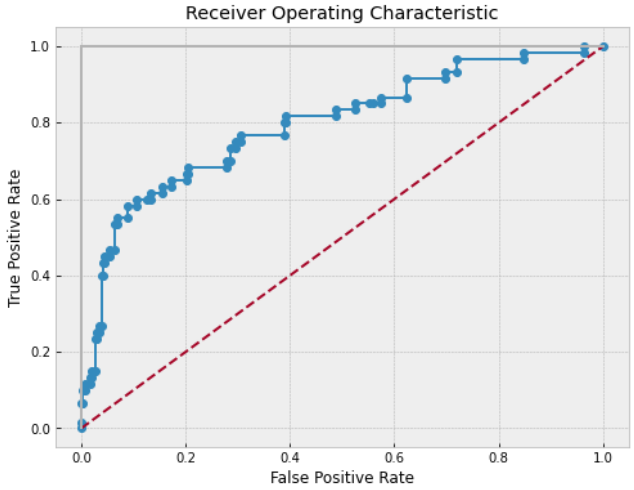
PARAMETROS OPTIMIZADOS:
regresion_logistica = LogisticRegression()

```
max_iter=100,
penalty='none',
solver='newton-cg'
```

1.5.3 XGBoost:

Rango de parámetros de búsqueda:

```
'max_depth'       : list(np.arange(1, 10, step=1)), #np.arange(2,11,1),
'n_estimators'    : list(np.arange(5, 30, step=5)),
'learning_rate'   : list(np.arange(0, 0.3, step=0.01)),
'gamma'           : list(np.arange(0, 10, step=2)
```



PARAMETROS OPTIMIZADOS:
xgb_model_sg = xgboost.XGBClassifier ()

```
objective      = 'binary:logistic',
n_estimators   = 25,
seed           = 42,
max_depth      = 3,
learning_rate  = 0.23,
gamma          = 0
```

1.5.4 Resultados optimizados

	arbol_optimizado	Regresión Logística optimizada	XGboost_optimizado
Accuracy	0.82337	0.866848	0.847826
Precision	0.22222	0.666670	0.600000
Recall	0.03333	0.366670	0.200000
ROC_curve	0.66523	0.796750	0.795830