**Recursos Humanos**

1. **Introducción / Problema del Negocio**

Una empresa solicita ayuda debido a que muchos empleados han ido abandonando la compañía, por lo tanto, se quiere saber porque sus empleados se van.

Dado el conjunto de datos disponibles, se debe determinar un plan para utilizar el modelado de datos y así proporcionar una visión empresarial impactante.

Teniendo esto en cuenta, si se pudiera predecir las probabilidades de que un empleado renuncie, la empresa podría seleccionar a esos empleados para que reciban un trato especial.

1. **Data**

El departamento de recursos humanos de la empresa ha compilado un conjunto de datos que creen que serán útiles para tratar con el problema. Incluye detalles de los niveles de satisfacción de los empleados, evaluaciones, tiempo de trabajo, departamento y salario.

La empresa comparte sus datos enviando un archivo llamado hr\_data.csv. Este conjunto de datos contiene las siguientes propiedades:

* Filas: 15.000 muestras (filas) que representan a los empleados anteriores o actuales.

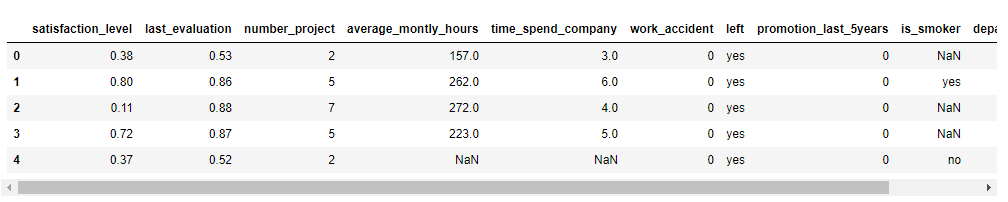
Columnas:

* **left**: si el empleado abandonó o no la empresa.
* **satisfaction\_level**: representa el nivel de satisfacción de cada empleado.
* **last\_evaluation:** indica la última evaluación realizada al empleado.
* **average\_montly\_hours:**  promedio de horas trabajadas al mes.
* **time\_spend\_company:** tiempo que tiene en la compañía.
* **number\_project:** número de proyectos realizados por cada empleado.
* **work\_accident:**  si el empleado ha tenido algún accidente en el trabajo o no.
* **promotion\_last\_5years:**  muestra si el empleado ha sido ascendido de cargo en los últimos cinco (5) años o no.
* **is\_smoker:** indica si el empleado es fumador o no.
* **department:**  departamento de la empresa en el cual trabaja el empleado.
* **salary:**  el salario que devenga cada empleado.

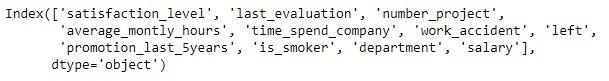
1. **Metodología**

**Fase de preparación de los datos:**

El primer paso que se utilizo fue cargar los datos usando la librería Pandas, mostrando las primeras filas para así verificar que están estructurados los datos en una tabla con sus respectivas columnas (las cuales están conformadas por cada variable necesaria para realizar el estudio) y sus filas las cuales representa a cada uno de los empleados.

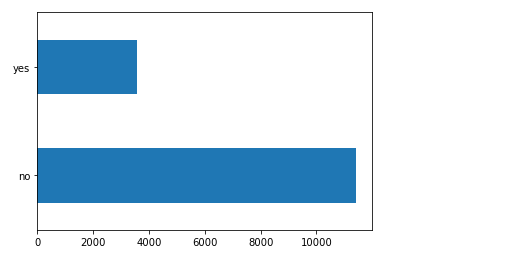


**1.1 Se muestran las primeras filas de la tabla para comprobar que fueron cargados con los datos con éxito.**



**1.2 En esta imagen se muestran todas las columnas(variables) que posee la tabla.**

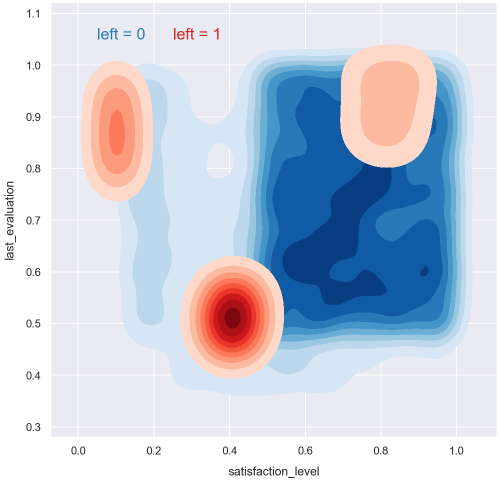
**Selección de Variables**: como el problema a resolver consiste en poder predecir que empleados se van a ir o no de la compañía, necesitamos revisar la columna(variable) ‘left’, sabiendo que está compuesta por solo dos (2) clases: si el empleado se fue o no se fue.



**1.3 Gráfica de barra de la variable ‘left’**

La variable a predecir ‘left’ presentaba un desbalance en sus clases ya que 11428 registros pertenecen a la clase ‘no’ y 3571 registros a la clase ‘yes’. Como indica la gráfica de la imagen 1.3, esto nos dice que se tenía que tratar con un problema de clasificación desbalanceado, lo cual implica que se tendría que tomar medidas especiales para contar cada clase cuando se calcule su precisión.

Luego para buscar relaciones existentes entre las distintas variables que fueran estadísticamente significativas para nuestro propósito, encontré que los resultados arrojados en la última evaluación realizada a los empleados, está altamente relacionada con su nivel de satisfacción, los cuales, a su vez, una vez asociadas mostraban relación con nuestra variable target (‘left’). Podemos observar dichas relaciones en el siguiente gráfico de densidad:



**1.4 Gráfico de Densidad, mostrando como influyen esas 2 variables en la variable target.**

A través del grafico de densidad se pueden hacer algunas observaciones:

1. Existe la tendencia de que la mayoría de empleados que abandonaron la empresa arrojaron un nivel de satisfacción menor a 0.5 y en su última evaluación su puntuación fue menor a 0.6.
2. Otra importante porción de los empleados que abandonaron la empresa contaba con un nivel de satisfacción inferior al 0.2, a pesar de que en su última evaluación tuvieron un promedio elevado de 0.8.

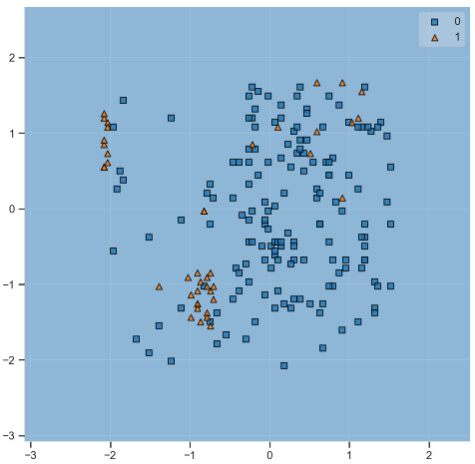
Aprovechando esta relación existente entre esas variables con la variable target, se realizó el modelado de datos.

**Fase de Modelado de datos**

Para el modelado de datos se recurrió a la ejecución de varios modelos de machine learning de aprendizaje supervisados especializados para tratar con problemas de clasificación binaria, ya que el propósito final es poder predecir si el empleado abandona la empresa o permanece en ella. A continuación, se informará sobre el desempeño realizado por cada modelo que se usó hasta llegar al modelo predictivo más óptimo para llevar a producción:

* Primer modelo usando el algoritmo de Support Vector Machine (Máquinas de Vectores de Soporte):

El resultado es 75.9 % de precisión, el cual no es tan malo tomando en cuenta que es el primer modelo. Es importante resaltar que las clases en nuestra variable target están altamente desbalanceadas, ya que la mayoría de registros en los datos corresponden a empleados que aun trabajan en la compañía.



**1.5 Gráfico de región, se puede ver como la mayoría de registros pertenecen a la clase 0.**

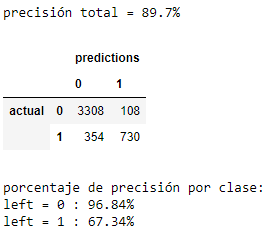
Teniendo en cuenta esto, era necesario enfocarnos en la precisión de cada clase además de la precisión general.



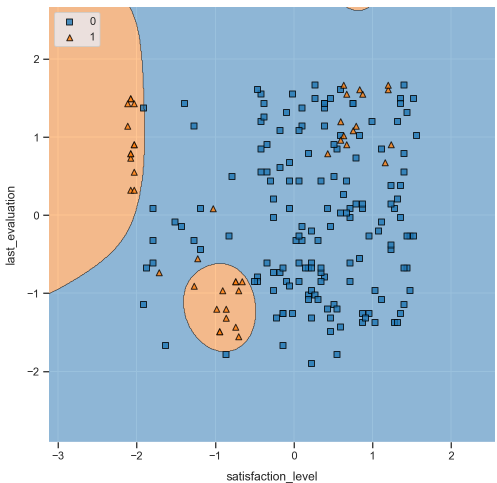
El modelo estaba clasificando cada muestra como 0, esto significa que estaba prediciendo que no hay empleados en el conjunto de prueba que dejaran la compañía.

Como era de esperarse un modelo lineal no era útil para estos datos.

* Segundo modelo usando el algoritmo de Support Vector Machine (Máquinas de Vectores de Soporte):



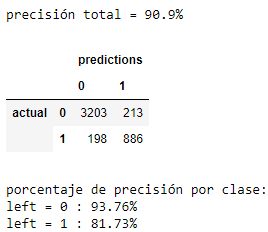
**1.6 Precisión del Segundo Modelo.**



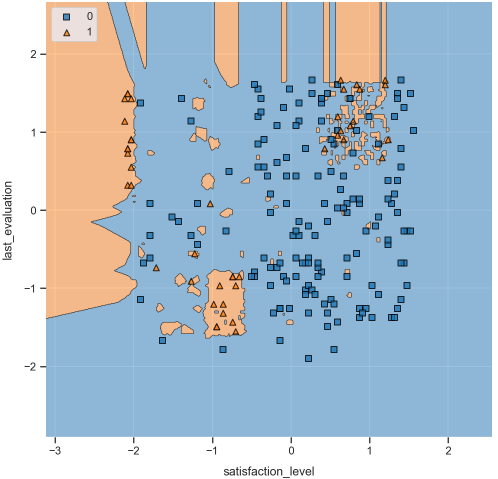
**1.7 Gráfico de región del segundo modelo.**

Para este segundo modelo utilicé como parámetros para el algoritmo el kernel rbf en lugar del kernel lineal utilizado en el primer modelo, como se puede apreciar en la imagen 1.6 y 1.7, este resultado está mejor. Podemos ver una precisión general cercana al 90%, donde la precisión de la clase 1 es 67% comparado con el 0% del linear SVM.

* Tercer modelo usando el algortimo de K-Nearest Neighbors (K- Vecinos Cercanos):



**1.9 Precisión del tercer modelo.**

****

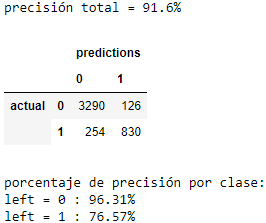
**1.10 Gráfico de región tercer modelo, muestra claramente que se han sobre ajustado los datos.**

En este modelo asigné como parámetro de número de vecinos igual a 3.

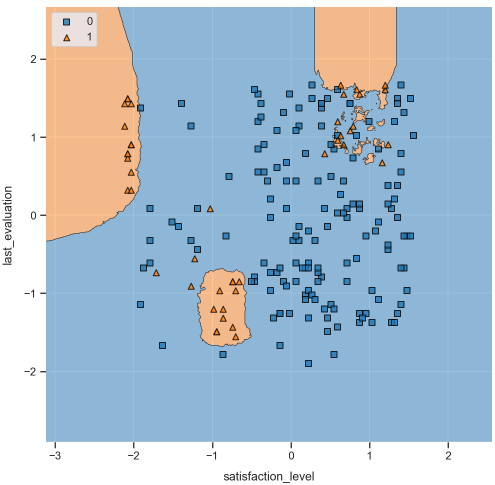
Como muestra la imagen 1.9, vemos que incrementó la precisión general sobre el 90 %, también hubo una mejora significativa para la clase 1 en particular con una precisión cercana al 82 %. Sin embargo, el gráfico de regiones en la imagen 1.10, indica que hubo un sobre ajuste de los datos. Esto se puede mejorar incrementando el número de vecinos cercanos en el clasificador.

* Cuarto modelo usando el algortimo de K-Nearest Neighbors (K- Vecinos Cercanos):

Para reducir el sobre ajuste presentado en el primer modelo, se entrenó un modelo KNN con n\_neighbors = 25.



**1.11 Precisión del cuarto modelo.**

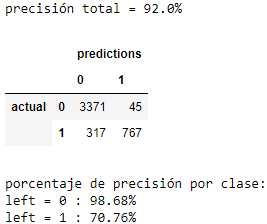
****

**1.12 Gráfico de región del cuarto modelo.**

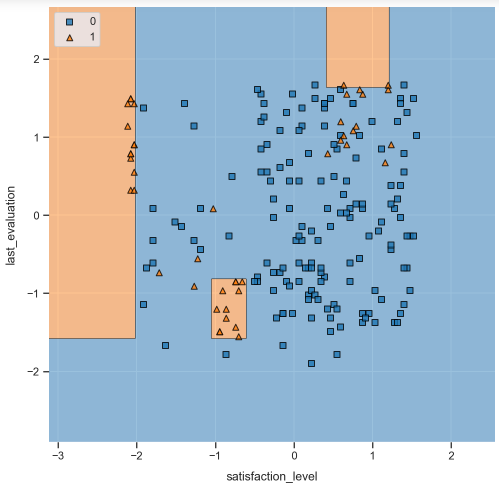
Como podemos ver en la imagen 1.12, los límites de decisión están significativamente menos entre cortados en comparación con la gráfica del modelo n\_neighbors = 3.

Observando las métricas, la precisión de la clase 1 es menor.

* Quinto modelo usando el algoritmo del clasificador Random Forest (Bosque Aleatorio)



**1.13 Precisión del quinto modelo.**

****

**1.14 Gráfico de región del quinto modelo.**

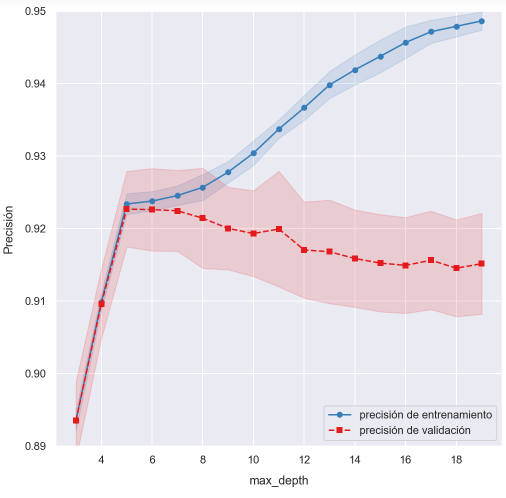
En este modelo se usó el clasificador Random Forest el cual es un algoritmo ensamblador que contiene múltiples árboles de decisión, seleccioné como parámetros 50 árboles y max\_depth = 5.

Como se puede ver por la precisión que arrojó el modelo, no hubo mucho avance con relación al cuarto modelo, se necesita usar un método más completo (como el k-fold cross validation).

Para los siguientes modelos utilicé el método de k-fold cross validation y así dividí los datos en varias partes para buscar un modelo con un mejor desempeño y precisión sobre todo en la clase 1.

* Sexto modelo usando el algoritmo de Decision Tree (Árbol de Decisión).

Se probaron 10 modelos con los datos, usando el método de validación cruzada con el parámetro max\_depth de 2 a 20 para así poder encontrar cual es el que arroja un mejor desempeño en el entrenamiento de los datos.



**1.15 Curva de Validación del Árbol de Decisión.**

La configuración del parámetro de max\_depth para los árboles de decisión controla el equilibrio entre el desajuste (underfitting) y el sobreajuste (overfitting) de los datos. Esto se refleja en la curva de validación, donde podemos ver bajas precisiones para valores de max\_depth pequeños (underfitting), ya que no permitimos que el árbol de decisión cree suficientes ramas para capturar los patrones en los datos.

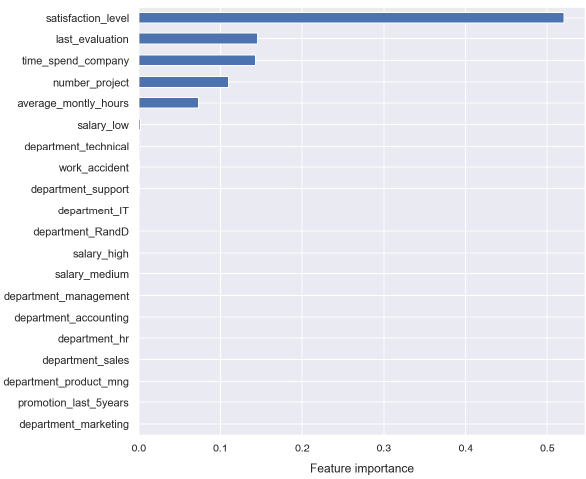
Para los valores de max\_depth grandes a la derecha del gráfico, podemos ver que sucede lo contrario, ya que los árboles de decisión aquí se superponen a los datos de entrenamiento. Esto se evidencia por el hecho de que nuestra precisión de validación (cuadrados rojos) disminuye a medida que aumenta max\_depth.

Se observa cómo la precisión del entrenamiento (círculos azules) continúa aumentando a medida que aumenta max\_depth. Esto sucede porque los árboles de decisión pueden capturar patrones cada vez más detallados en los datos de entrenamiento. Al observar las precisiones de validación, podemos ver que estos patrones no se generalizan bien para los datos invisibles (test data).

Según este gráfico, un buen valor para max\_depth parece ser 6.

* Reducción de Dimensionalidad con PCA (Análisis de Componentes Principales).

Después de entrenar una variedad de modelos para predecir la rotación de empleados con los datos, todavía se tenía que utilizar la mayoría de las variables disponibles. Por lo tanto, el siguiente paso que realicé fue determinar la función "feature importance" utilizando un clasificador de árbol de decisión y seleccioné todas las variables disponibles para formar un árbol de decisiones con el conjunto de datos completo, el resultado se puede apreciar en la siguiente gráfica:



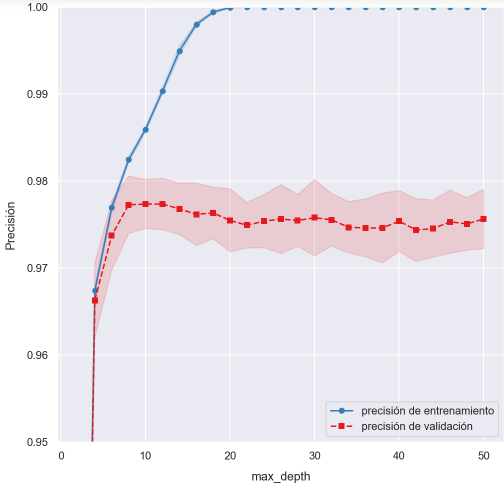
**1.16 Gráfica de barras que indica que variables influyen más en el modelo.**

Como se muestra en el gráfico de barras, hay varias variables que son de gran importancia cuando se trata de hacer predicciones, y el resto parece tener una importancia cercana a cero.

Dejamos a un lado las cinco variables más importantes del gráfico anterior para que se puedan usar para modelar y seleccioné el resto para usarlo con el algoritmo PCA.

* Primer modelo listo para producción usando el algoritmo de Decision Tree (Árbol de Decisión).

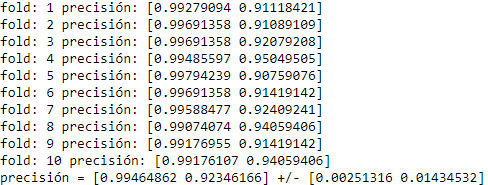
Después que pude determinar las cinco variables que influyen más al algoritmo para realizar predicciones, lo primero que hice fue ejecutar una curva de validación, usando un clasificador de árbol de decisión, usando como parámetro de max\_depth un rango de 2 en 2 hasta 52 para así encontrar cual es el valor óptimo de dicho hiperparámetro.



**1.17 Curva de Validación, teniendo en cuenta la precisión en el eje y, max\_depth en el eje x.**

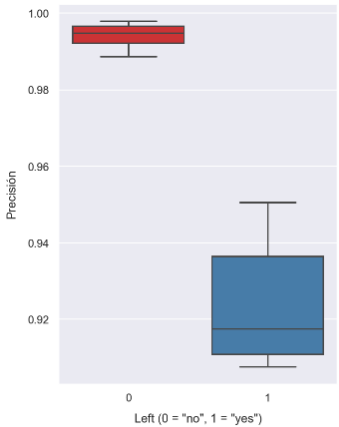
Al observar esta curva de validación, se puede ver que la precisión del conjunto de entrenamiento (círculos azules) se acerca rápidamente al 100%, alcanzando esta marca alrededor de max\_depth = 20. El conjunto de validación (cuadrados rojos) alcanza una precisión máxima alrededor de max\_depth = 8. Esto sucede porque los modelos en este rango se sobre ajustan en los datos de entrenamiento.

Basado en los resultados, seleccioné max\_depth = 8 como valor óptimo para usar en el modelo de producción.



**1.18 Se puede ver el resultado luego de ejecutar el clasificador usando los datos en diez partes.**

Una vez ejecutado el clasificador con el método de k- fold validación cruzado usando como hiperparámetro max\_depth = 8, puede verse que este modelo se está desempeñando mucho mejor que los modelos anteriores para la clase 1, con una precisión promedio del 92,2% +/- 1,5%. Esto se puede atribuir a las variables adicionales que usé, en comparación con los modelos anteriores que se basaban en solo dos variables.



**1.19 Gráfico de caja que muestra la precisión de cada clase en la variable target.**

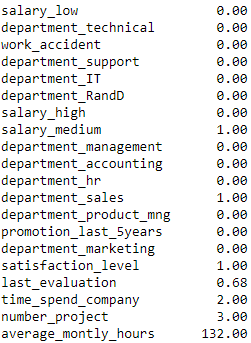
Luego, ejecuté el modelo con el conjunto completo de los datos para descubrir el desempeño real del clasificador y el resultado fue el siguiente:



**1.20 Precisión del modelo: 99,35 % para la clase 0 y 94,25 % para la clase 1.**

Estas precisiones de test (prueba) deben estar dentro o muy cerca del rango de las precisiones de k-fold cross validation que calculamos previamente. Para la clase 0, podemos ver el 99,2%, que se encuentra dentro del rango de k fold del 98,8% al 99,7%, y para la clase 1, se puede ver el 94,4%, que se encuentra también dentro por del rango de k fold del 90,7% al 95 %.

Finalmente realicé una prueba con el modelo escogiendo un registro (el número 8500) dentro de los datos, al cual le coloqué el nombre del empleado imaginario “Juan” y lo filtré en la columna original de variables, arrojando los siguientes datos:



**1.21 Los datos del registro 8500 pertenecientes a “Juan”.**

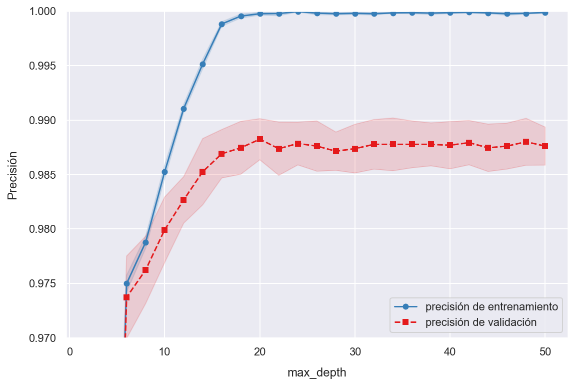
Después de transformar los datos con el algoritmo PCA, al realizar la predicción en el modelo para saber si “Juan” permanecerá en la empresa o la abandonará, el modelo le asignó más de un 99 % de probabilidad de que “Juan” seguirá en la empresa, tal como lo indica en la imagen 1.22.



**1.22 El modelo asignó un 99,61 % a la clase 0.**

* Segundo modelo listo para la producción usando el algoritmo ensamblador de Random Forest (Bosques Aleatorios).

Para este segundo modelo de producción, escogí el clasificador random forest debido a que está conformado de múltiples árboles de decisión, lo cual podría generar mejores resultados que el modelo anterior y en efecto así fue como veremos más adelante, repetí el mismo proceso usando el método de k-fold validación cruzada y en los hiperparámetros del clasificador, asigné a max\_depth un rango de 2 en 2 hasta llegar a 52, de manera que al trazar la curva de validación se pueda reflejar cual es el valor óptimo para el algoritmo. El resultado obtenido después de trazar la curva de validación fue:

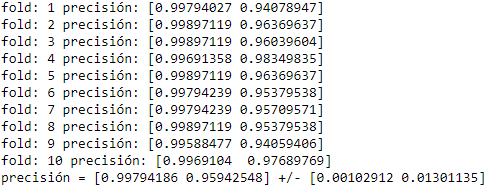


**1.23 Curva de validación del clasificador Random Forest.**

A diferencia de la curva de validación anterior, el conjunto de validación aquí parece converger a la máxima precisión a medida que aumenta max\_depth. Este no es el comportamiento que vimos anteriormente para el árbol de decisión, donde la precisión del conjunto de validación alcanza un máximo de alrededor de max\_depth = 8, y luego se redujo ligeramente para valores más altos de max\_depth.

En este caso, para nuestro modelo Random Forest, solo es importante establecer max\_depth lo suficientemente alto para evitar un desajuste, y no deberíamos preocuparnos por el sobreajuste en valores grandes de max\_depth. Dado esto, y el hecho de que la precisión de la validación parece converger alrededor de 20, seleccionaremos max\_depth = 25 como el valor óptimo para este hiperparámetro.

Una vez logré encontrar el valor óptimo de max\_depth, ejecuté el clasificador seleccionando también n\_estimators = 50, es decir el bosque aleatorio, estará conformado por 50 árboles de decisiones. Al realizar el cálculo de la precisión de cada clase bajo el método k-fold validación cruzada, obtuve las siguientes puntuaciones:



**1.24 La imagen muestra las puntuaciones de las precisiones dadas en ambas clases.**

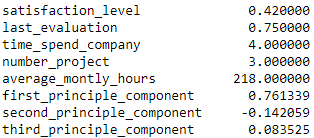
Comparando este modelo con los resultados que generó el árbol de decisión entrenado anteriormente, se puede observar una mejora significativa de la precisión de cada clase, con la clase 0 aumentó de 99.5% a 99.8% y con la clase 1 aumentó de 92.2% a 95.94%.

Luego de ver que mejoraron los resultados con los datos de entrenamiento, procedí a evaluar el desempeño del modelo con los datos de prueba, obteniendo una precisión del 99.59% para la clase 0 y 97.22% para la clase 1, tal como lo muestra la imagen 1.25.



**1.25 Precisiones con el conjunto de datos de prueba.**

Para finalizar, realicé una prueba con este modelo de la misma manera en que probé el modelo anterior, seleccionando un registro (691) y filtrándolo con el nombre de “María” para predecir qué clase el modelo le asignaba y con cuánta probabilidad, este empleado abandonaría la empresa o no.



**1.26 Datos de “María”.**

El resultado de la predicción acera de María es:



**1.27 La imagen indica que existe un 84 % de probabilidad que María abandone la empresa.**

Este resultado indica que el modelo está prediciendo con un 84% de probabilidad que María dejará la compañía.

Una manera de mejorar las posibilidades de que la compañía pueda retener a María como empleada, es que ellos deben reducir la cantidad de tiempo que ella necesita pasar en el trabajo. Usando el modelo se puede probar el efecto que podría tener sobre su probabilidad de irse.

Se configura average\_montly\_hours = 120 y time\_spend\_company = 2 y luego se re-evalua la probabilidad de predicción del modelo:



**1.28 Ahora el modelo indica que existe un 86 % de probabilidades que María continúe en la empresa.**

Este resultado sugiere que, si se reduce el número de horas al mes en el trabajo a 120 y la cantidad de tiempo en la compañía a nivel 2, existe un 86% de probabilidad que María no abandone la compañía.

1. **Discusión**

Retomando al problema original, el cual era que la empresa quiere reducir el número de empleados que abandonan, se ha logrado crear un herramienta la cual permite con una alta precisión el poder predecir la probabilidad de que un trabajador renuncie, teniendo así la empresa la posibilidad de tener un trato especial con dicho empleado, estudiando las causas, bien sea por el salario, las horas de trabajo al mes o demás variables que estén incidiendo en el descontento del empleado, como lo vimos en el ejemplo de “María” que su promedio de horas de trabajo al mes, era muy elevado.

1. **Conclusión**

Luego de haber realizado varios modelos predictivos con los datos suministrados por el departamento de recursos humanos de la empresa, se logró crear una herramienta (‘rh-analisis-pca-forest.pkl’) que les permitirá a las personas encargadas teniendo prudencia y ética profesional llevar un seguimiento de todos los empleados, observando las posibles causas que les esté produciendo un descontento con la empresa y así tomar una decisión oportuna con ese empleado para evitar que abandone la compañía.