**Course Approve Detection Classifier**

En este archivo se explicará el proceso y la lógica que se empleo para resolver el ejercicio desafío que se me propuso.

Se comenzó con el ejercicio propuesto realizando un pequeño EDA con las distintas variables presentes en el CSV. Se vio la distribución de todas las variables, los posibles valores que estas podían tomar y el tipo de datos que contenían las distintas columnas, sin perder de vista el problema a resolver.

**Consideración:** Algo importante a mencionar antes de comenzar a detallar el proceso de transformación de los datos, es que, en la consigna pude ver que se pedía realizar un índice compuesto por el **id alumno** / **id materia** / **partición**. Si bien dada la naturaleza de los datos esto parecía ser lo más apropiado, decidí no incluir la partición en el índice. Eso lo hice así ya que no me parecía correcto considerar una **variable temporal** en el índice, ya que cada partición generaba una repetición de registros de cada alumno y curso.

Dicho de otra manera, me resultaba mas complejo evaluar la performance de un único alumno, cuando la información respecto a la performance de este mismo se encontrada dividida en una variable temporal.

Por lo tanto, lo que decidí hacer es trabajar los datos de manera que cada fila del Data Set fuera un único alumno en un determinado curso, así de esa manera se tendría toda la información del alumno en una única fila y sería más fácil crear un modelo para detectar si ese alumno iba a aprobar o no determinado curso.

Echa esta aclaración comenzare a describir como fue el proceso de transformación de los datos.

**Data Wrangling:**

Lo primero que se hizo fue crear una copia nueva del Data Set original, en donde el índice estaba compuesto por **id alumno / id materia**.

Por lo que una única fila se correspondía con un alumno en determinada materia, un id de alumno podía aparecer varias veces (un alumno inscripto en distintas materias) así lo mismo con las materias (varios alumnos realizando una materia).

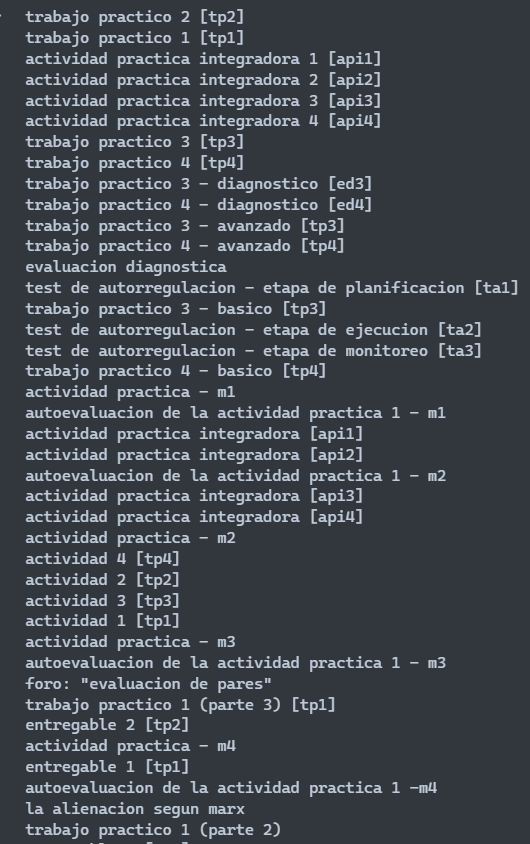
Agrupe en base al periodo de la materia y en base a la nota final de cada alumno. La nota final se corresponde a la **variable target** del modelo de clasificación.

Una vez generado el índice lo que hice fue tratar de trabajar la información disponible de manera que se pueda obtener la mayor información con los datos que se tenía.

**Trabajos Prácticos:**

Primero agregue la información en base a los trabajos prácticos lo que busque acá era ver que trabajos prácticos había entregado cada alumno y que nota había obtenido, además de ver que trabajo practico se correspondía a cada materia, porque no todas las materias tenían la misma cantidad y tipo de trabajos prácticos.

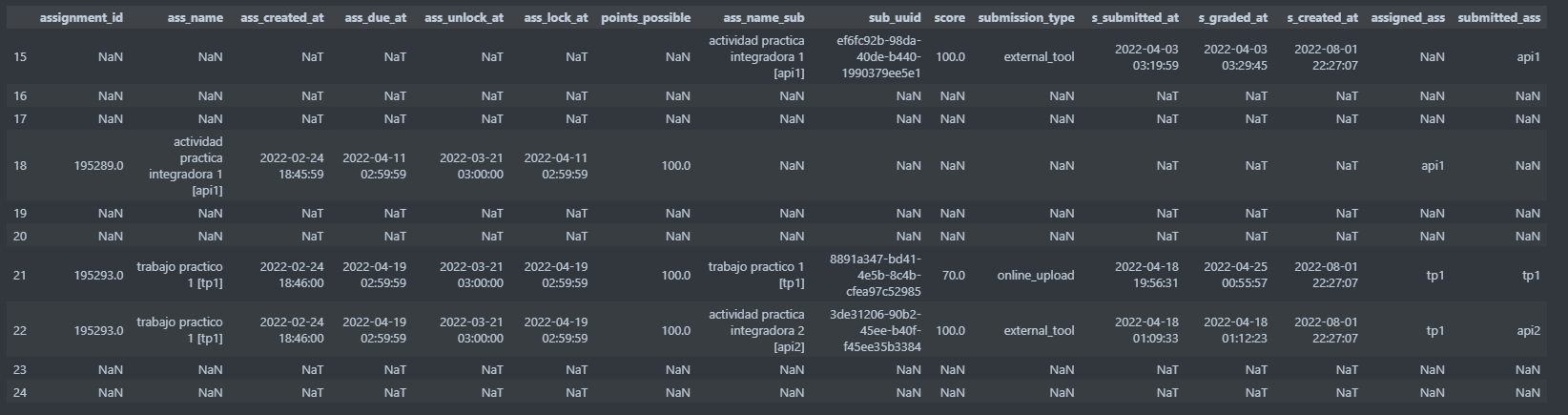
Las variables que se utilizaron para obtener esta información fueron “**ass\_name**” y “**ass\_name\_sub**”, viendo los distintos nombres que tomaban los distintos trabajos vi que podía unificarse los trabajos practicos en una única consideración.



Se crearon dos variables nuevas (“**assigned\_ass**” que se obtuvo de la columna ass\_name y “**submitted\_ass**” se obtuvo de la columna ass\_name\_sub), en las cuales mediante **regex** se extraía de los distintos valores de tipo de trabajo practico, este mismo se encontraba encerrado entre corchetes “[….]”.

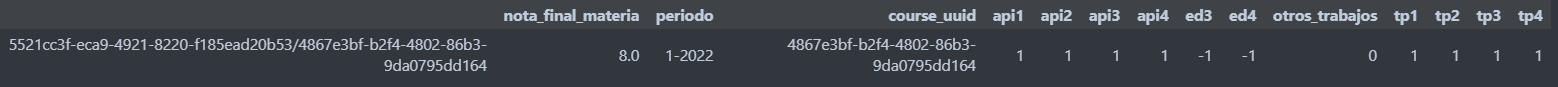
Se agruparon los trabajos prácticos en varias categorías: TP1, TP2, TP3, TP4, API1, API2, API3, API4, ED3, ED4 (no figuraban valores de ED1 y ED2).

Luego se hizo la consideración de que, **si los trabajos prácticos solo tenían nombre, pero no el tipo** (“entregable grupal 1” por ejemplo) se los consideraría dentro de una categoría llamada “**otros\_trabajos**”.

Al crear las dos variables nuevas y mientras analizaba los datos obtenidos me di cuenta que en la **partición** de la asignación de un trabajo practico no necesariamente correspondida con la entrega del mismo, y que en una misma partición podía haber con un mismo nombre de trabajo asignado otro tipo de trabajo entregado. Además de que la entrega de un trabajo podía tener otro nombre diferente al asignado. como en el ejemplo de abajo.

**Consideración:** Debido a que la variable **ass\_name** y **assigned\_ass** repetían valores varias veces y no se correspondían con las tareas que los distintos alumnos entregaban se utilizó la variable **submitted\_ass** para obtener los trabajos prácticos entregados por los alumnos, y se verifico con los trabajos asignados que, los trabajos entregados se correspondan a los que se asignaban en cada materia.

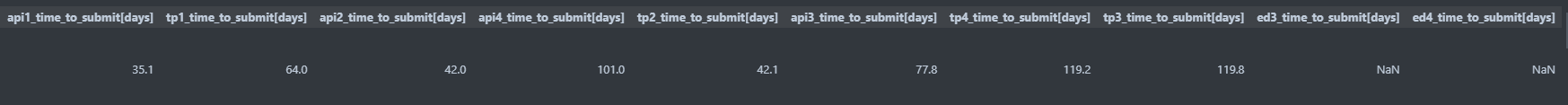
Los trabajos prácticos que no eran asignados en una materia se los flaggeo con un valor de -1. Por lo tanto, los posibles valores que podía tener cada entrega de trabajo practico era:

* 1 a n (n siendo un numero entero) ­– Si el trabajo practico se asigno en la materia y el alumno lo entrego. Si un trabajo practico era entregado varias veces n se correspondía con la cantidad de veces.
* 0 – Si el trabajo practico era asignado, pero no entregado.
* -1 – Si el trabajo practico no se correspondía con la materia considerada.

Aplicando la misma lógica que para lo anterior obtuve también el **puntaje** de cada alumno en la entrega de cada trabajo.

Luego de obtener la información mencionada, trabaje con las variables temporales para obtener el **tiempo que demora cada alumno en entregar cada trabajo.**

Para esto primero obtuve la fecha en la que se asignaba cada trabajo practico (columna ‘**ass\_created\_at**’) en cada curso. Luego obtuve la fecha en el cual cada alumno entrego (columna ‘**s\_submitted\_at**’) los trabajos prácticos de cada materia.

 Con estas dos fechas fui capaz de obtener el tiempo que transcurrió desde la asignación, hasta la entrega de cada trabajo practico.

**Formato de entrega:**

El siguiente paso del Data Wrangling fue obtener la cantidad de entregas de cada formato que un alumno había realizado, siguiendo la misma lógica (agrupando por el índice y obteniendo las cantidades de formatos entregados de cada alumno).

Los posibles tipos de entrega son: 'external\_tool', 'online\_upload', 'online\_quiz', ‘discussion\_topic', 'basic\_lti\_launch', 'online\_text\_entry', 'media\_recording'.

**Exámenes:**

La última información que se agrego a la data final fue la relevante a los exámenes.

**Consideración:** En la columna ‘ass\_name\_sub’ (se corresponde a los trabajos prácticos entregados) contenía un valor que hacia referencia a una ‘evaluación diagnostica’, me pareció que esta evaluación era importante a la hora de determinar si un alumno aprobaba o no una materia. Debido a esto, agregue a la misma como una feature para el modelo.

Luego de agregar las notas, lo que hice fue flagear con -1 aquellos exámenes que el alumno no hubiese rendido, ya que si los flageaba con 0 podría estar imputando un valor erróneamente (como si el alumno hubiera obtenido un 0 en dicha evaluación).

**Limpieza registros sin exámenes.**

Por último, una vez que ya generé las variables de interés necesarias, procedí a hacer una limpieza de datos. Pude observa que había alumnos que no tenían ningún examen rendido, esto podía deberse a algún tipo de error en la carga de datos o algún error externo que no es de mi conocimiento. Ya que podían afectar negativamente la performance del modelo, decidí eliminar estos registros.