**Procesamiento Inicial de los Datos**

Se hace análisis del estado de los datos a partir del diccionario variables. Se encuentran las siguiente inconsistencias:

**train.csv**:

- No existe la columna *Dist_Sum_Nal* del diccionario de datos; esta característica no será incluida en el entrenamiento.
- Tiene una columna que no está en el diccionario de datos, *Dist_Max_INTER*; esta característica se añade al conjunto de entrenamiento, pues *test.csv* también la tiene.

**test.csv**:

- Tiene las siguientes columnas que no están en el diccionario de datos:
  - *Dist_max_COL*
  - *Dist_max_INTER*
  - *FECHA_FRAUDE*
  - *Dist_Max_INTER*
  - *Dist_mean_NAL*
  - *Dist_sum_INTER*
  - *Dist_mean_INTER*

- Se toma las siguientes acciones con las columnas anteriores:
  - *Dist_max_COL* será renombrada a *Dist_max_NAL*
  - *Dist_Max_INTER* se deja porque *train.csv* también la tiene.
  - El resto se elimina, porque no estan en el *train.csv*
  -
- Se toman otras acciones:
  - La columna *Dist_Sum_Nal* se elimina porque *train.csv* no lo tiene.
  - La columna *FECHA* se reubica a la posición que tiene en *train.csv*.

Con los cambios hechos en este archivo se crea uno nuevo llamado *test2.csv*

En las dos tablas, las columnas que hace referencias a fecha serán cambiadas por sus valores en timestamp, esto para manejar una medida de distancia más precisa entre los diferentes valores.

Para cada tabla las siguientes columnas tienen valores faltantes:

**train.csv**

- *FECHA_VIN* - continua
- *SEXO* - categórica
- *SEGMENTO* - categorica
- *EDAD* - continua
- *INGRESOS* - continua
- *EGRESOS* - continua
- *Dist_Sum_INTER* - continua
- *Dist_Mean_INTER* - continua
- *Dist_Max_INTER* - continua
- *Dist_Mean_NAL* – continua

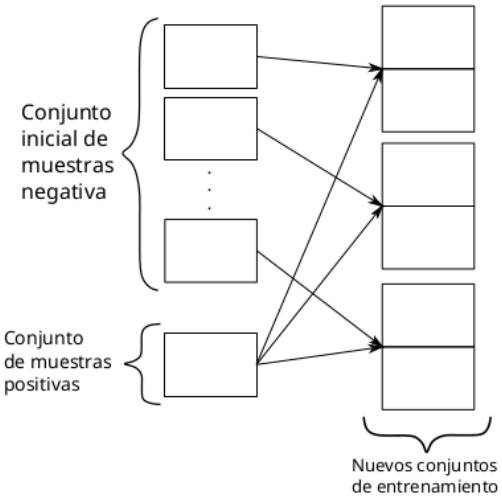
- **test2.csv**
  - *Dist_Sum_INTER* - continua
  - *Dist_Mean_INTER* - continua
  - *Dist_Max_INTER* - continua
  - Dist_Mean_NAL – continua

Para las columnas marcadas como continua los valores faltantes fueron inputados a partir de la media de la correspondiente columna. Para las columnas marcadas como categórica los valores faltantes fueron inputados a partir de la moda de la correspondiente columna.

Las variables categóricas de cada tabla (Canal1, COD_PAIS, CANAL, SEXO, SEGMENTO), fueron transformadas para representarse como One Hot Encoding. Las variables iniciales se eliminaron de la tabla.

**Validación**
En el data set *train* se tienen 2965 muestra con una proporción de desbalance en las muestras que son fraudes y las no fraudes de apróximadamente 0.327.

El modelo de entrenamiento que se eligió fue una red neuronal. Para evitar que este terminara sesgando sus decisiones debido al desbalance se optó por realizar un entrenamiento por conjuntos balanceados. Consiste en separar las muestras positivas y negativas del conjunto de entrenamiento. Luego, el conjunto de muestras negativas se subdivide en conjuntos de aproximadamente el tamaño del conjunto de muestras positivas. Con cada conjunto de las muestras negativas y el conjunto de las muestras positivas se crea un nuevo conjunto de entrenamiento, como se indica en el flujo de la siguiente figura:




Con cada conjunto nuevo de entrenamiento se crea un modelo. La predicción para una muestra será la media de las predicciones de todos los modelos.

Para la configuración de la red se eligió un grupo de valores para las capas internas, el valor de regularización L2, el ńumero de iteraciones y la función de optimización de los pesos. Se hizo una combinación de todos con todos en un proceso de validación cruzada de 5 folds.

**Predicción de test**

Con la configuración de parámetros que se obtuvo el mejor resultado en la validación se creo un modelo con todas las muestras de train y se realizó la predicción para test.

**Análisis de Resultados**

De las validaciones que se hizo el valor AUC más alto obtenido en el conjunto de prueba fue 0.86. Aunque no es despreciable, tiene que mejorar mucho más. Se tienen que buscar otras estrategias para mejorar la capacidad predictiva del modelo. Por ejemplo mirar estrategias que apunten a modificación a nivel de algoritmos y no a técnicas de muestreo. Además, la cantidad de datos de entrenamiento es muy pequeña, seguro con una cantidad mucho mayor la capacidad predictiva mejoraría, teniendo en cuenta que se trabajó con redes neuronales.

Se podría implementar un proceso que a partir de técnicas de clustering se pueda clasificar muestras que con seguridad no son fraude o son fraude y sólo realizar el entrenamiento de modelos con las muestras que haya incertidumbre, tal vez de esta manera se elimine mucho "ruido" de los datos.

Se debe mejorar además la calidad de los datos existentes para no perder información. Se podría agregar información que de cuenta de la frecuencia con que realiza transacciones. A la fecha se le podría agregar la hora de realización de la transacción. Todo esto apuntando a captar el comportamiento de los clientes.

**Ejecución de los Procesos**

- **Procesamiento de datos**: en la carpeta *data_processing* se encuentra el archivo Python *data_processing.py*. La ejecución de este transformará los archivos *train.csv* y *test2.csv* en *new_train.csv* y *new_test.csv*, correspondientemente. Estos archivos se utilizan como entrada para el proceso de validación.

- **Validación**: en la carpeta *training_validation* se encuentra los siguientes archivos que se pueden configurar:
  - *params.ini:* parámetros de ejecución del script de validación:
    - ruta para el archivo de parámetros de entrenamiento.
    - Ruta a archivo con el esquema de resultados.
    - Ruta al archivo *new_train.csv.*
    - Ruta al archivo donde se guardan los resultados.
  - *training_params.json*: parámetros para contruir la red neuronal.

  La ejecución del archivo *experiment.py* iniciará todo el proceso de validación y producirá un archivo en excel con los resultados de todas las validaciones cruzadas. Tiene dos hojas, una con los resultados de todos los fold de todas las validaciones y otra con la media por validación.

- **Predicción de test**: en la carpeta *test_prediction* se encuentra el archivo *test_prediction.py*. Su ejecución crea un modelo con la mejor configuración, que tiene quemada, del proceso de validación y realiza una predicción para el data set *test2.csv.*