Entrega Segundo Sprint Proyecto

Ciencia de Datos Aplicada 2-2021

Integrantes:

Sara Fernanda Sánchez Sánchez sf.sanchez10@uniandes.edu.co
Rafael de Jesús Guzmán Martínez rguzmanm@uniandes.edu.co
Yamile Andrea Cepeda Chaparro ay.cepeda@uniandes.edu.co

TABLA DE CONTENIDO

PRESENTACIÓN PROYECTO:

Presentación de proyecto:

Chiper es un startup colombiano, nació como proyecto del semillero de proyectos Ideamos, en el año 2017 y en el 2018 se materializa e inicia operación en Bogotá, hoy en día se encuentra en México, Colombia, y próximamente en Brasil.

Chiper es un servicio de abastecimiento para los negocios y tiendas de barrio, se encuentran todos los productos de la canasta básica familiar, licores, productos de aseo personal y del hogar y alimentos para mascotas, en Chiper se pueden hacer pedidos mínimos de 90.000 y el envío gratis, todo a través de la app todos los días en cualquier horario. Queremos, a través de este proyecto ayudarles a identificar sus dolores y entregarles una solución que les permita seguir con su crecimiento y expansión.

Definición de roles:

Andrea Cepeda, Líder de proyecto: Está a cargo de la gestión del proyecto. Define las fechas de reuniones, pre-entregables del grupo y verifica las asignaciones de tareas para que la carga sea equitativa. Se encarga de subir la entrega del grupo, si no hay consenso sobre algunas decisiones, tiene la última palabra.

Sara Sánchez, Líder de datos: Se encarga de gestionar los datos que se van a usar en el proyecto y de las asignaciones de tareas sobre datos. Debe dejarlos disponibles para todo el grupo, y puede definir los datos que se usarán para cada iteración. Adicional se encarga de la extracción y preprocesamiento de los datos.

Rafael Guzmán, Líder de negocio y líder de Analítica: Es responsable de velar por resolver el problema o la oportunidad identificada y estar alineado con la estrategia del negocio o las características del grupo para el cual se plantea el proyecto. Debe garantizar que el producto se puede comunicar de forma apropiada al contexto del negocio o de la comunidad. Adicional se encarga de velar por el cumplimiento de estándares de análisis en todos los entregables y el despliegue de las aplicaciones en la máquina virtual asignada para el grupo.

Repositorio GitLab: https://gitlab.com/ing.rafael.guzman89/chip_ml.git

Repositorio GitHub: https://github.com/SanchezSara/CienciaDatos

ACTIVIDADES DEL SPRINT

1. Enfoque analítico:

Propuesta: Crear una herramienta de predicción que permita a Chiper, identificar a los clientes en riesgo de dejar la empresa.

Problema: Chiper ha visto una disminución de sus clientes, los cuales no retiene bien, esto genera una pérdida de sus ingresos y desea conocer por qué está pasando esto y cómo puede fidelizar y retener a sus clientes.

Objetivos del proyecto:

- Crear una herramienta que a partir de la ciencia de datos apoye los procesos de fidelización de clientes.
- Identificar y segmentar los clientes con características similares.
- Predecir dadas ciertas características, los usuarios que abandonarán la empresa.
- Identificar las causas por las que un cliente decide disminuir y abandonar la alianza con Chiper.
- Prevenir el abandono por parte de los clientes.

Dado lo anterior, la tarea predictiva que se trabaja es la de clasificación, con variable objetivo ‘churn’, del tipo booleano, devuelve 1 para Si y 0 para No, cuándo los

clientes han pasado 30 días o más sin realizar ninguna compra. Las varibles predictoras son: 'totalOrdersUSD', 'storeTypeId', 'storeStatusId', 'totalOrders', 'warehouseId', 'cantidadMetodosDePago', 'cityId', 'countryId', 'locationId', 'isActive', 'hasChiperSuppliedPOS'; siendo las de localización, las de mayor relevancia.

2. Entendimiento de los datos/ Preparación de datos:

Overview

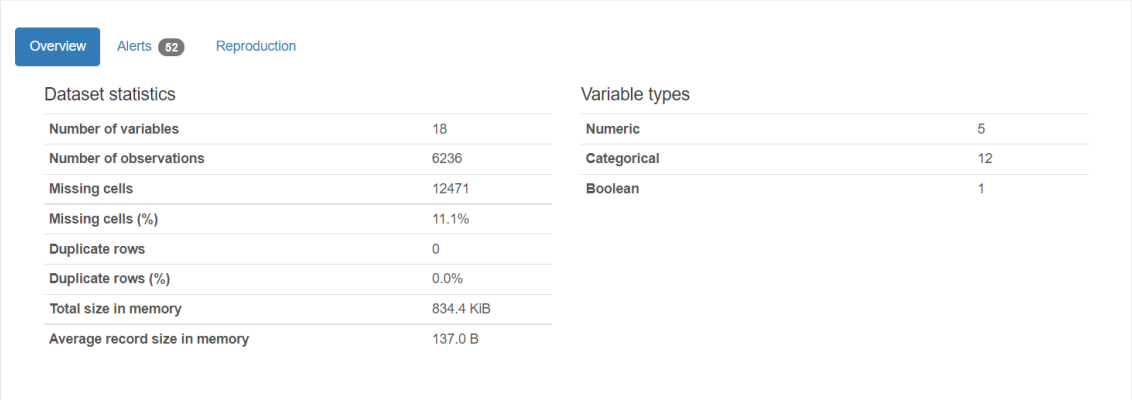

Imagen 1: Perfilamiento de los datos

Para el proyecto desarrollado, lo que se espera identificar, es, por un lado, las características de un cliente, que lo permiten pertenecer a un segmento determinado y por otro el *churn,* entendido esto, como una medida del tiempo que el cliente pasa sin hacer ninguna compra dentro de la aplicación. Así pues, buscamos identificar el segmento de clientes en riesgo de dejar la aplicación, para que, por medio de políticas internas, puedan ser fidelizados.

Para la segunda entrega del proyecto actual, se espera realizar el modelo clasificatorio para la caracterización de los usuarios de la aplicación de Chiper.

Para este caso, contamos con un total de 2 datasets, que contienen información de los meses de Septiembre y Octubre de 2021, esto con el fin de identificar el comportamiento de los clientes en este periodo de tiempo. Para este primer data set, contamos con 12.385 registros o filas con 18 campos diferentes. Esta información corresponde a un total de 6.236 usuarios.

Los datos obtenidos son de las fechas 2021-09-01 al 2021-09-30, por lo cual contamos con 60 días de información, lo que nos permitirá realizar la evaluación temporal del comportamiento de las tiendas.

El total del dataset presenta 11.1% de datos faltantes, los cuales están explicados por las variables socialClass y numberOfEmployees, las cuales presentan un 99% de datos faltantes, para estos dos casos, se deciden eliminar las variables del dataset, pues afecta significativamente al mismo. Por otro lado la variable

storeStatusId presenyta un total de datos vacíos de 33, lo que representa el 0.5% de esa data, para este caso se llenan los campos faltantes con el dato más frecuente de la misma variable.

Por otro lado, se encontró que que la variable 'hasChiperSuppliedPOS' era del tipo booleano string, por lo cuál se decide codificar, de modo que su procesamiento sea más fácil y acorde con el resto del dataset, obteniendo para esta variable 1 y 0.

El dataset aunque tenía más datos de una variable que otra, no presentaba una diferencia significativa en la evaluación de los modelos y por último se aplicó la normalozación MinMaxScaler, para tener una misma proporción en los datos.

3. [10%] Preparación de los datos:

La división del dataset y verificación de distribución en ambos conjuntos (train y test) se realiza durante el train split, dónde la distribución se establece sea 70-30.

Se divide la data en test y train. Además se da formato a los datos que lo requieren

```
[136] X_train, X_test, Y_train, Y_test = train_test_split(data_mod, Y, test_size=0.3, random_state=1)
```

Imagen 2: Preparación de los datos

La validación cruzada por su parte se realiza cuando se usa fit con X_train = Y_pred y luego se usa el modelo con Y_test.

```
def pred(X_test, clf):
    # Predicton on test with giniIndex
    Y_pred = clf.predict(X_test)
    return Y_pred
```

Imagen 3: Validación cruzada

Del mismo modo, es posible verificar, en la evaluación de los modelos construidos qué tanto se explica de la variable en el conjunto test, siendo este, un muy buen rendimiento.

4. Construcción del modelo.

Se construyeron dos modelos para la tarea de clasificación Árboles de Decisión y Random Forest.

Para Random Forest se encontró que los mejores parámetros son n_estimators=1000 y random_state=42

```
rf.fit(X_train, Y_train)

RandomForestClassifier(n_estimators=1000, random_state=42)
```

Imagen 4: Construcción Random Forest

Por su parte, para el modelo árbol de decisión, se realizó la ejecución del mismo para encontrar los hiper parámetros de número de hojas y profundidad, con estos datos se encontraron los mejores parámetros usando grid search.

```
from sklearn.tree import DecisionTreeClassifier
ct = ColumnTransformer([
       ('num', KNNImputer(n_neighbors=5),
       make_column_selector(dtype_include=np.number)),
       ('w',
       KNNImputer(weights='uniform'),
       make_column_selector(dtype_include=object))])

estimators = [('imputer', ct),
              ('normalize',MaxAbsScaler()),
              ('clf', DecisionTreeClassifier(max_depth=46)),
              ]
pipe = Pipeline(estimators)

param_grid = dict(imputer__num__n_neighbors=[5,7,8],
                  imputer__w__weights=['uniform', 'distance'],
                  clf__criterion=['gini', 'entropy'],
                  clf__splitter=['best','random'],
                  clf__max_depth=[5,10,20,26,30,37,40,46,54],
                  normalize=['passthrough', MaxAbsScaler(), MinMaxScaler()])

grid_search = GridSearchCV(pipe, param_grid=param_grid,cv=5,verbose=3,scoring='accuracy')

grid_search.fit(data_encoder,Y.ravel())
```

Imagen 4: Construcción Árboles de decisión

5. Evaluación del modelo.

En las siguientes imágenes se aprecia la evaluación de los dos modelos, dónde se obtienen rendimientos muy similares en los dos. Sin embargo, si hay que se específicos el modelo Random Forest presenta un mejor rendimiento, siendo este de 0.81 y 0.47 en la explicación de las variables.

Reporte para el Mejor Modelo Random Forest

```
metricas(Y_test, Y_pred)
```

```
Matriz de confusión:  [[1077  258]
 [ 285  251]]
Precisión :  70.97808658471406
Reporte :               precision    recall  f1-score   support

           0       0.79      0.81      0.80      1335
           1       0.49      0.47      0.48       536

    accuracy                           0.71      1871
   macro avg       0.64      0.64      0.64      1871
weighted avg       0.71      0.71      0.71      1871
```

Imagen 5: Evaluación modelo Random Forest

```
[165] print(metricas(Y_test,pred(X_test,clf)))

      Predicted values:
      [0 0 0 ... 0 1 0]
      Matriz de confusión:  [[1057  278]
       [ 282  254]]
      Precisión :  70.06948156066275
      Reporte :               precision    recall  f1-score   support

                0       0.79      0.79      0.79      1335
                1       0.48      0.47      0.48       536

          accuracy                           0.70      1871
         macro avg       0.63      0.63      0.63      1871
      weighted avg       0.70      0.70      0.70      1871

      None
```

Imagen 6: Evaluación modelo Árboles de decisión

6. Despliegue de la solución:

A continuación, se muestra el despliegue del modelo, usando la herramienta REST.

En la Máquina Virtualse logró desplegar un servicio API Rest, pero lamentablemente no hay forma de visualizar la parte gráfica.

Imagen 7: Servicio REST

Imagen 8: Servicio REST

MODELO DE APRENDIZAJE AUTOMATICO

PREDICTOR DE DESERCIÓN DE COMERCIOS DE CHIP

RESULTADOS

CONTINUARA CON CHIP

Imagen 9: Servicio REST

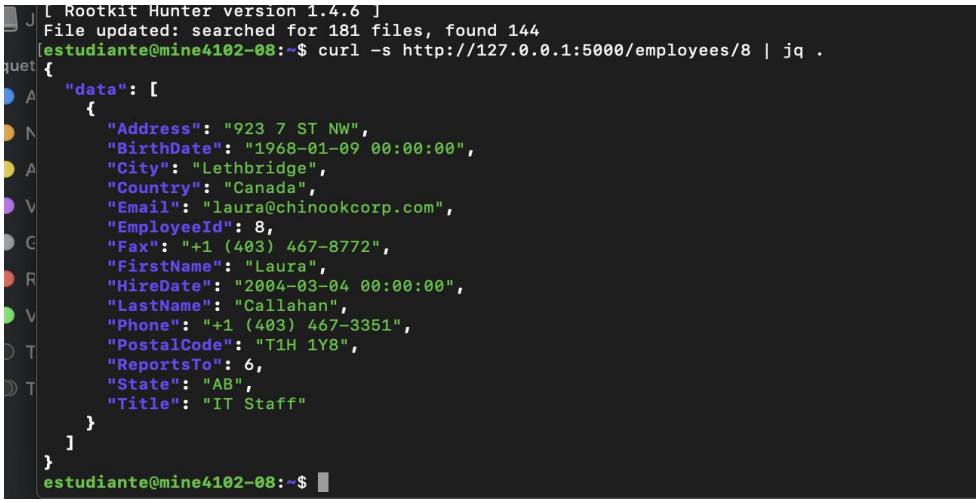

Imagen 10: Servicio REST

7. Conclusiones:

Imagen 11: Reporte de Reuniones

¿Se requieren más datos?

No, dado que los datos recolectados por mes, son todos los que la empresa ha recogido, se logra explicar el comportamiento de los clientes.

Y si, creemos se requieren más datos de cada una de las tiendas con el fin de conocer un poco más del cliente que opera esa tienda, por ejemplo, datos como el género, la edad, el estrato, etc.

¿Qué tipo de datos?

Los datos pueden ser numéricos o cadena de caracteres y nosotros haríamos la transformación correspondiente en los que hubiere lugar.

¿Cuáles fueron las mayores dificultades durante el proceso?

Una de las mayores dificultades que presentamos fue en la consolidación del dataset, la definición de las columnas y la forma de agregación.

¿Cómo soporta la tarea de predicción el problema de negocio?

Le permitirá a la compañía identificar clientes en churn y actuar de manera rápida incentivándolos con diversas campañas de fidelización que demoren su fuga y mientras tanto fidelizarlos desde diferentes estrategias. La empresa de alguna manera ya lo hace sin embargo se requiere tener herramientas ágiles que les permitan tomar muy rápido decisiones.

¿Cómo podría usarse dentro de un producto o un proceso actual de la compañía?

El modelo se puede usar en el área de inteligencia de negocios, allí lo pueden integrar con sus demás herramientas para complementar el trabajo que allí hacen.