Trabajo fin de Máster:

Customer Relationship Management y Software (CRM)

Análisis descriptivo y pronóstico de abandono

Título: MODELO DE DESCRIPTIVO Y PRONÓSTICO DE ABANDONO

INTRODUCCIÓN

En los últimos años ha surgido la banca electrónica que se caracteriza por la oferta de servicios online, que secundariamente han generado grandes bases de datos cuyo análisis puede ofrecer estrategias de mercado de gran interés. Entre ellas destacamos el modelo predictivo de abandono que analiza el comportamiento de los clientes, e intenta predecir futuros desertores buscando optimizar costes.

OBJETIVOS

PRIMARIO:

1. Realizar un análisis descriptivo de los diferentes canales de contratación para seleccionar aquellos con mejor comportamiento (contratación por canal, rentabilidad)
2. Diseñar un modelo predictivo de abandono de clientes

DISEÑO DEL ESTUDIO

Se extraen datos de Weborama y salesforce entre Noviembre del 2017 hasta Junio de 2019 para el diseño del modelo predictivo. Para obtener el modelo predictivo de abandono se utilizó una regresión logística y randomforest

Los datos utilizados para el análisis descriptivo de los canales de contratación sólo corresponder a los meses entre Octubre del 2018 y Enero de 2019. Para dicho análisis se seleccionaron las siguientes variables:

1. Site_Offer: Canales de generación de Leads.
2. ID: identificador del cliente
3. Conversion_date_hour: Registro de entradas
4. Conversion_label: Proceso de contración

Para las variables continuas, se calcularon medias; Para las categóricas, se calcularon los porcentajes. Para el análisis de los datos se utilizó Jupyter notebook. Python

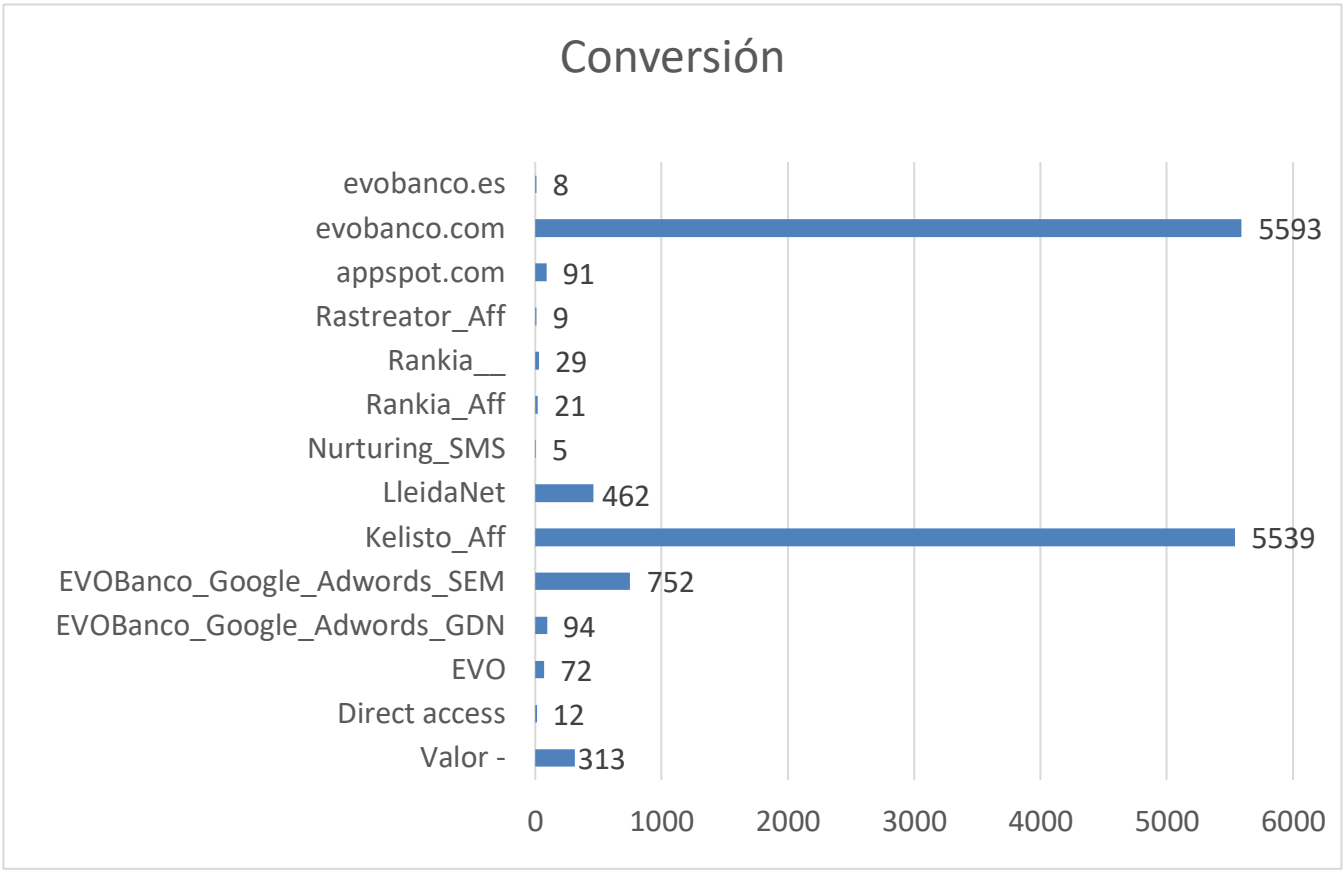
I.ANÁLISIS DE LOS CANALES DE CONTRATACIÓN. RESULTADOS Y DISCUSIÓN

Se parte de una base de datos de 745632 registros.
Canales de contacto con el usuario:

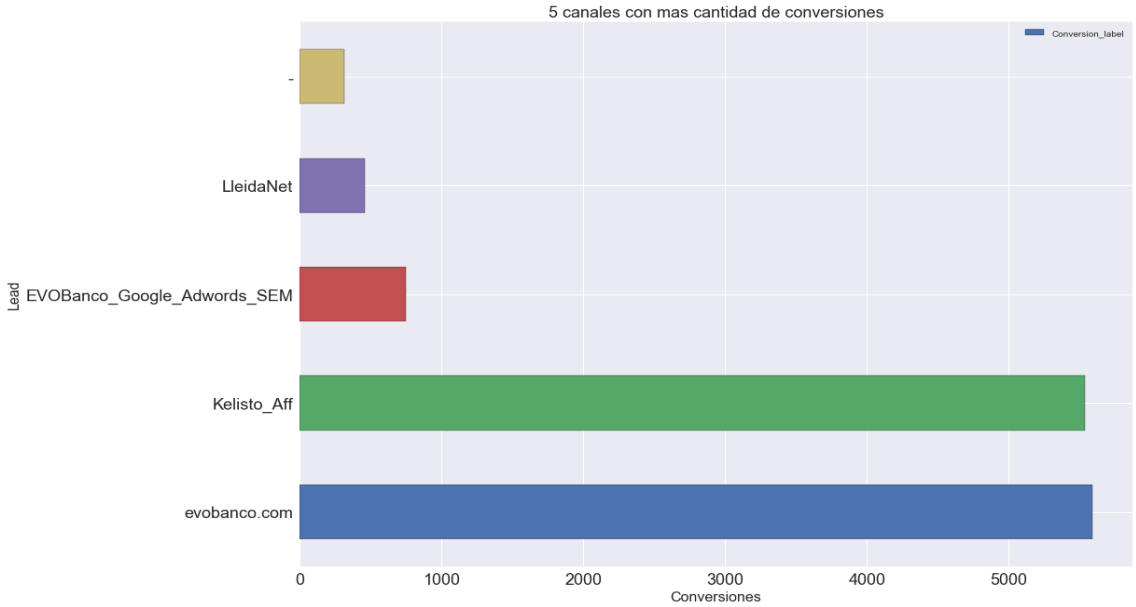
Conversion_label	Site_Offer
0.WELCOME	17
12_Identificacion_iban	30788
13_Identificacion_iban_fotos	354
14_Fin_proceso	15903
15_Descarga_documentacion	20189
16_ContratacionOK	12975
1_Email_movil	366598
2_Confirmacion_OTP	56562
3_Datos_personales	74455
4_Datos_contacto	44725
5_Datos_laborables	33197
6_Metodo_identificacion	34941
7_Identificacion_video_delantera	32720
8_Identificacion_video_trasera	12319
9_Identificacion_video_foto	9888

1_Email_movil es la primera forma del contacto del usuario con la plataforma en número absolutos. Le sigue 3_Datos_personales

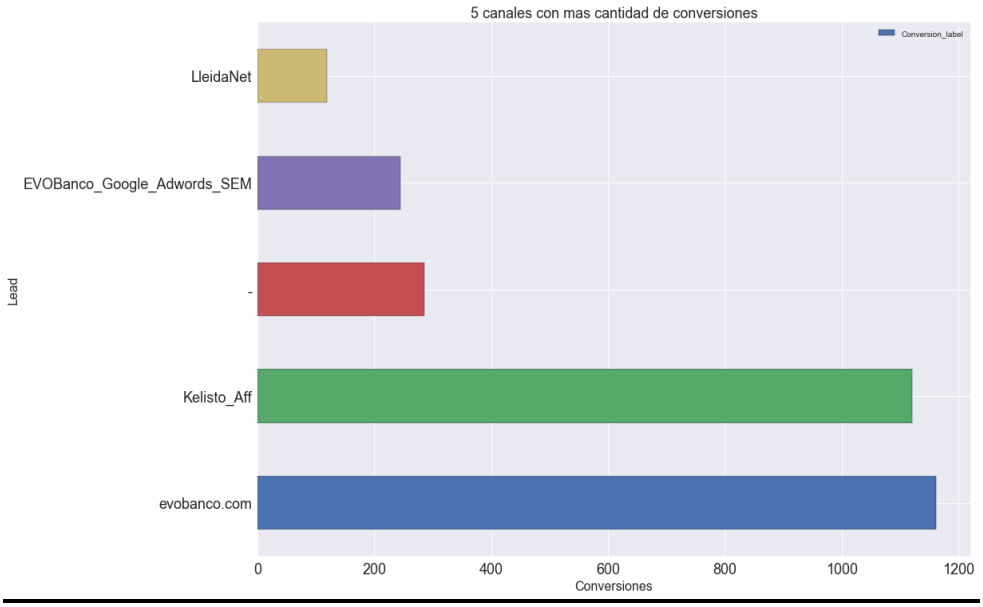
Resultado neto de conversión por canales:



Los cinco canales con posible conversión (bruto):



Los cinco canales con mayor conversión (neto):



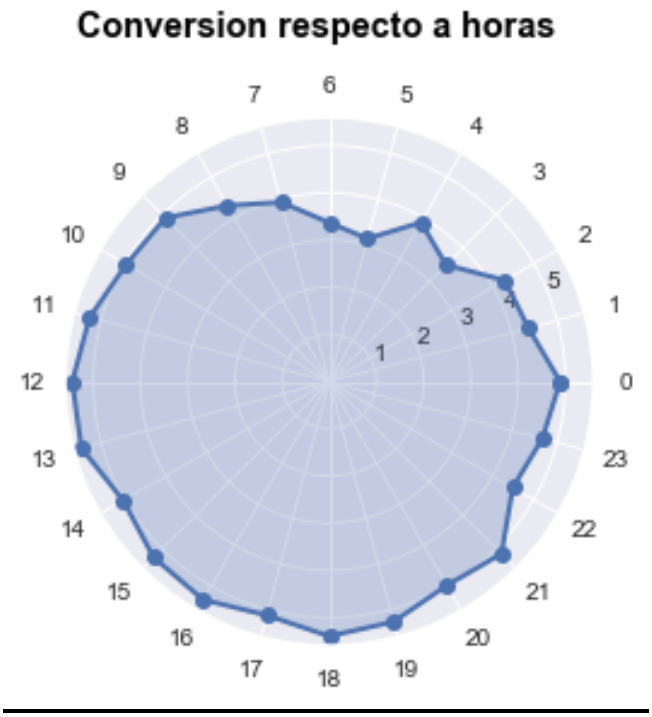
Se observa que la matriz (“evobanco.com”), es el canal con mejor conversión. El segundo canal es canalKelisto_Aff.

Tasa de conversión durante los días del mes

Nota pie de figura: Rosa (octubre), azul (noviembre), negro (diciembre), amarillo (enero).

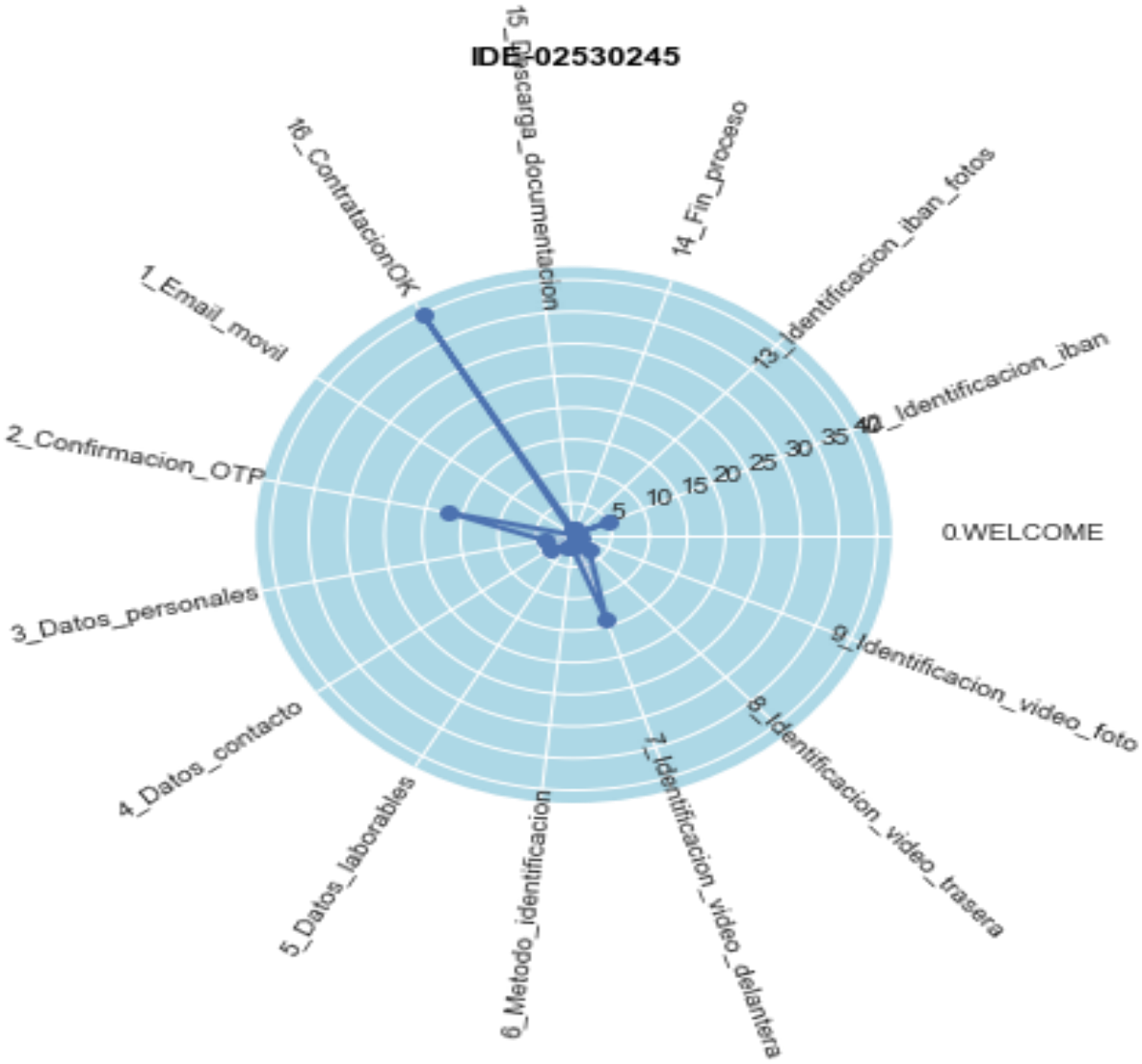
Del análisis se observa que en los meses de Octubre, Noviembre y Diciembre hay una mayor conversión en los primeros días del mes, que posteriormente presenta una disminución y después tres picos de conversión entre los días 10 y 25. En enero destaca un pico de conversión entre los días 5-11 por una campaña publicitaria de préstamos personales (solo para clientes)

Franja horaria de mayor conversión



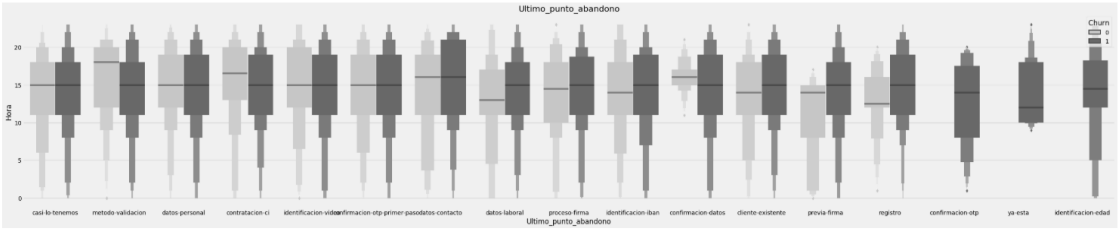
Mayor conversión en horario diurno y vespertino (de 10.00 am a 20.00 pm). Como acción adicional se produce un proceso de nurturing a través de SMS para reimpáctar a los clientes que ya están dentro del funnel. Esa acción se realiza sobre las 10:30 de la mañana, ya que dentro del proceso de firma los horarios de mayor conversión son desde las 10:30 a 11:30

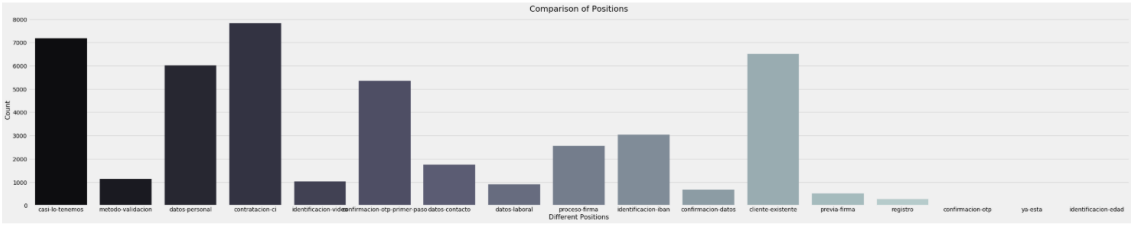
Ejemplo de pasos de clientes:

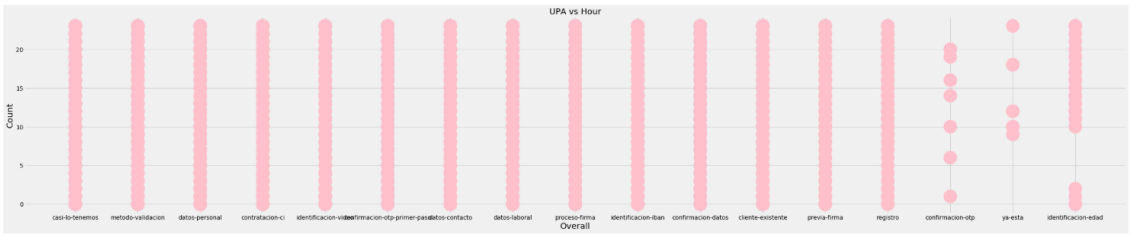


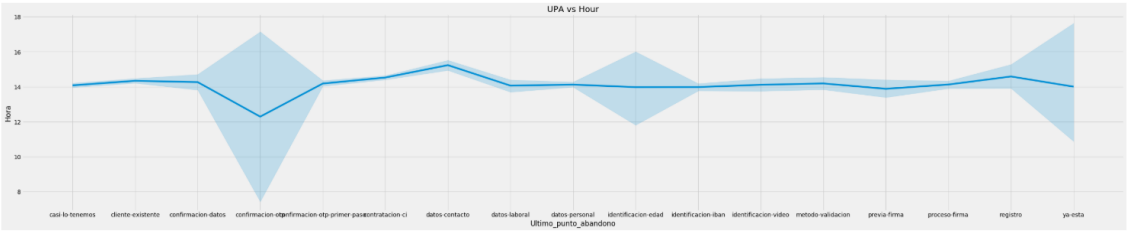
Se puede observar por todos los pasos que tiene que pasar un cliente para poder llegar a finalizar el proceso. Todo esto condicionado por el tiempo, ya sea porque refrescan la página o lo abandonan por tener que hacer otras funciones. Incluso también condicionado de si llama a atención al cliente para que le ayuden en el proceso.

II.GRAPHICAL ANALYSIS









III.MODELO PREDICTIVO DE ABANDONO. RESULTADOS Y DISCUSIÓN

1. Recopilación y limpieza de datos.

```
The number of null values is:  0
ID_Cliente_EVO           0
Producto                 0
Estado                   0
F_creacion               0
Numero_caso              0
Fecha_Hora_Apertura      0
Hora_Apertura            0
Ultimo_punto_abandono    0
Hora                     0
Minuto                   0
Horas_por_15             0
Churn                    0
dtype: int64
```

Se comprueba que no existen valores nulos

2. Seleccionamos que variables queremos incluir en el modelo. .

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 44809 entries, 0 to 44808
Data columns (total 12 columns):
ID_Cliente_EVO           44809 non-null object
Producto                 44809 non-null object
Estado                   44809 non-null object
F_creacion               44809 non-null datetime64[ns]
Numero_caso              44809 non-null int64
Fecha_Hora_Apertura      44809 non-null datetime64[ns]
Hora_Apertura            44809 non-null object
Ultimo_punto_abandono    44809 non-null object
Hora                     44809 non-null int64
Minuto                   44809 non-null int64
Horas_por_15             44809 non-null object
Churn                    44809 non-null int64
dtypes: datetime64[ns](2), int64(4), object(6)
memory usage: 4.1+ MB
```

Se excluyen las variables no relevantes (ID_cliente_EVO, F_Creacion, Numero_Caso, Producto, Fecha_Apertura, Hora_Apertura, Minuto, Hora_por_15)

Las varibles categóricas incluidas se transformaron y agruparon en un formato numérico, ya que nuestro modelo de aprendizaje automático solo puede funcionar con datos númericos.

3. Prueba y entrenamiento del modelo:

   Regresión logística

```
X_train, y_test = train_test_split(dataset, test_size = 0.30)

print("Train: ", len(X_train))
print("Test: ", len(y_test))
```

```
Train:  31366
Test:  13443
```

```
dataset["Churn"] = dataset["Churn"].astype(int)
Y = data["Churn"].values
X = dataset.drop(labels = ["Churn"],axis = 1)
# Create Train & Test Data

X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=101)
```

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
result = model.fit(X_train, y_train)
```

```
from sklearn import metrics
prediction_test = model.predict(X_test)
# Print the prediction accuracy
print (metrics.accuracy_score(y_test, prediction_test))
```

```
0.9744476679312654
```

```
Hora_8                                                 0.684378
Hora_22                                                0.616730
Estado_Pendiente de revisión BO                        0.531901
Hora_9                                                 0.530971
Estado_Activo                                          0.444315
Hora_10                                                0.421864
Hora_6                                                 0.389714
Hora_14                                                0.324728
Hora_23                                                0.297491
Hora_16                                                0.292168
Hora_20                                                0.243722
Hora_12                                                0.237804
Hora_5                                                 0.194572
Ultimo_punto_abandono_previa-firma                     0.190804
Hora_11                                                0.174276
Estado_Pendiente revisión Captación                    0.171471
Hora_18                                                0.164617
Hora_15                                                0.158616
Hora_13                                                0.144495
Hora_0                                                 0.086788
Hora_17                                                0.038688
Hora_19                                                0.035717
Hora_1                                                -0.018605
Ultimo_punto_abandono_confirmacion-datos              -0.051045
Ultimo_punto_abandono_contratacion-ci                 -0.065145
Estado_Potencial                                      -0.066203
Ultimo_punto_abandono_identificacion-iban             -0.088047
Ultimo_punto_abandono_registro                        -0.132270
Ultimo_punto_abandono_proceso-firma                   -0.147632
Ultimo_punto_abandono_confirmacion-otp-primer-paso    -0.155805
Ultimo_punto_abandono_datos-personal                  -0.156053
Ultimo_punto_abandono_casi-lo-tenemos                 -0.159489
Hora_21                                               -0.161771
Ultimo_punto_abandono_datos-contacto                  -0.165282
Estado_Pendiente de Electronica ID                    -0.167745
Hora_7                                                -0.195928
Ultimo_punto_abandono_identificacion-video            -0.255464
Hora_3                                                -0.288094
Hora_2                                                -0.361135
Ultimo_punto_abandono_metodo-validacion               -0.421134
Ultimo_punto_abandono_datos-laboral                   -0.428498
Hora_4                                                -0.526057
dtype: float64
```

Se obtiene un modelo con alta precisión (Accuracy 0,97)

En la regresión logística las variables con valor positivo se relacionan con el evento (“abandono”). Observamos que las que mejor predicen el abandono son haber iniciado el proceso web de contratación a las 8, 9, 22 horas y encontrarse en “estado pendiente

de revisión". Por el contrario, los procesos finales de la contratación web como la "validación e introducción de los datos laborables", predicen que es poco probable que estos clientes abandonen.

Se realiza un análisis random forest, comprobando un similar grado de precisión

```
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators=200, random_state=0)
classifier.fit(X_train, y_train)
predictions = classifier.predict(X_test)
```

```
from sklearn.metrics import classification_report, accuracy_score
print(classification_report(y_test,predictions ))
print(accuracy_score(y_test, predictions ))
```

```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       229
           1       0.97      1.00      0.99      8733

    accuracy                           0.97      8962
   macro avg       0.49      0.50      0.49      8962
weighted avg       0.95      0.97      0.96      8962

0.9742245034590493
```

CONCLUSIONES

1. El análisis de los datos generados a partir del registro de clientes en portales de banca online es útil para conocer los canales con mayor rentabilidad, además de los días y franjas horarias con mayor conversión. En este caso se observa que los canales con mejor conversión son EVO, Kelisto , SEM, - y Lleida.
2. Los canales con peor resultado son Rastreator_Aff, Nurturing_SMS y Direct Access.
3. En los meses analizados el mayor pico de conversión es a principios de mes, excepto Enero que presenta un comportamiento atípico debido a estrategias estacionales de marketing.
4. La franja horaria de mayor conversión es la comprendida entre las 11 a 1 de la mañana, debido al departamento de abandono que se encarga de volver a contactar con el cliente ya sea por email o SMS. Y como último recurso, se contacta con el cliente por el SmartCenter.
5. En el modelo predictivo las variables que mejor predicen el abandono son iniciar el proceso de contratación web 8:00, 9:00, 22:00 horas y el "estado pendiente de revi

sión” con alta precisión. Deberían diseñarse estrategias de marketing específicas p ara estas situaciones.