ENTREGA FINAL - PROYECTO DE CURSO
Web Traffic Time Series Forecasting

Valentina Echavarria Porras [a], David Santiago Arcila Alvarez[b], Felipe Cadavid Rincon[c].

[a] Bachiller académico, estudiante perteneciente a la Escuela ambiental, Facultad de Ingeniería, Ingeniería Civil, Universidad de Antioquia, cédula:1214743266, Medellín (A), Colombia. valentina.echavarria1@udea.edu.co

[b] Bachiller académico con énfasis en pedagogía, estudiante perteneciente a la Escuela ambiental, Facultad de Ingeniería, Ingeniería Ambiental, Universidad de Antioquia, cédula: 100008537, Medellín (A), Colombia. davids.arcila@udea.edu.co

[b] Bachiller académico, estudiante perteneciente a la Escuela de ingeniería, Facultad de Ingeniería, Ingeniería de Sistemas, Universidad de Antioquia, cédula:1128389605, Medellín (A), Colombia. felipe.cadavidr@udea.edu.co

Aprueba
Raúl Ramos Pollán
Julián David Arias Londoño

UNIVERSIDAD DE ANTIOQUIA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA CIVIL, AMBIENTAL Y SISTEMAS
INTELIGENCIA ARTIFICIAL
MEDELLÍN, ANTIOQUIA
JUNIO DE 2022

1. INTRODUCCIÓN

PROBLEMA PREDICTIVO A RESOLVER
Con base a los requerimientos del reto planteado en Kaggle llamado Web Traffic Time Series Forecasting , el objetivo de este es implementar un modelo que nos permita realizar un pronóstico y predicción de tráfico web basado en series temporales; en este caso, del tráfico web de artículos de Wikipedia y verificar su desempeño contra la medición de precisión SMAPE.

La característica de una serie temporal es proporcionar una representación, la cual, realiza mediciones específicas en un momento dado de forma cronológica. Este mismo está compuesto usualmente de datos por intervalos de tiempo, pero no siempre proporciona vistas acertadas, para predecir y estimar de manera precisa, tal como se ve en algunas predicciones del clima o la bolsa de valores.

Por lo anterior, además de implementar un modelo para predicciones, también se busca llegar a determinar un modelo adecuado y acertado, mitigando problemas de análisis de datos e inferencia en la clasificación de datos, en este caso, para tráfico web.

2. EXPLORACIÓN DESCRIPTIVA DEL DATASET

DATASET A UTILIZAR
El dataset que ha sido seleccionado para este proyecto, el cual reside en Kaggle. (Ver Dataset aquí). El dataset consiste en 145.000 series temporales aproximadamente, que van desde el primero de Julio de 2015 hasta el 31 de Diciembre de 2016.

Adicionalmente incluye datos de prueba para entrenamiento que van desde el primero de enero de 2017 hasta el primero de marzo de 2017 para hacer pruebas de primera fase, y datos de prueba hasta septiembre de 2017 para la segunda fase de pruebas.

El proyecto contiene varias carpetas con diferentes archivos indicados a continuación:

1. Keys (key_1 y key_2): Contiene información concerniente a las páginas y los identificadores de las páginas
    a. ***page***: Contiene el nombre de la página visitada en la serie temporal concatenado en snake case con otros datos que pueden ser importantes para el análisis. Potencialmente se tendrá que masajear y transformar esta columna para tener una mejor representación de su contenido. Datos contenidos en esta columna:
        i. Nombre del artículo
        ii. Versión del artículo de Wikipedia por idioma
        iii. Origen/Agente, es decir, desde donde se abrió el articulo (Mobile, Desktop, etc)
        iv. Fecha del registro
    b. ***id***: identificador de la página visitada en la serie temporal. Establece la correlación de los datos de la columna page con las visitas en los archivos mencionados más adelante
2. Submissions (sample_submission_1 y sample_submission_2) :
    a. ***id***: Identificador, potencialmente la clave primaria que correlaciona las visitas con la página visitada (en los archivos keys)
    b. ***visits***: Contador, representado como un entero, indica la cantidad de visitas para el identificador respectivo.
3. Train (train_1 y train_2):
    a. ***page***: El nombre de la página visitada únicamente, concatenado con el agente en snake case
    b. ***columnas con las fechas en año calendario***: contiene la cantidad de visitas por cada fecha del calendario comprendido en los datos entregados

3. ITERACIÓN DE DESARROLLO

CONFIGURACIÓN INICIAL DEL PROYECTO

Para el desarrollo del proyecto se utilizaron las siguientes herramientas y tecnologías:

- Python: Lenguaje de programación para la interpretación de los datos
- Jupyter - Google Collab: Herramienta en línea que permite redactar, y ejecutar algoritmos usando python en un formato de archivo similar a un archivo de texto enriquecido.. Esta herramienta está integrada con Google Drive.
- Kaggle: Portal de fuente de información para los datos que se van a utilizar en el desarrollo de este proyecto.

INICIALIZACIÓN DEL AMBIENTE

Antes de poder iniciar la creación, procesado y ejecución de algoritmos se deben importar algunas dependencias, configurar la conexión con Kaggle y descargar los archivos necesarios en el ambiente de trabajo.

1. Instalación de dependencias adicionales: Utilizando el comando *pip*, se pueden descargar paquetes que no están por defecto. Para este proyecto se descagaron los siguientes:
    a. pystan: Interfaz de Python para Stan (modelado estadístico y la computación estadística de alto rendimiento, análisis de datos y predicción), para la inferencia bayesiana.(*PyStan*, 2019).
    b. fbprophet: Software de código abierto de Core Data Science de Facebook, permite análisis de series temporales y previsiones a escala (Timalsena, 2020).
    c. kaggle: Paquete para poder conectarse a Kaggle mediante comandos y API.
2. Configuración de credenciales para conexión con Kaggle: Para poder descargar los datos desde Kaggle en Collab se configura un archivo de configuración kaggle.json el cual se utilizará como referencia para conectarse con la API de kaggle

```
!mkdir ~/.kaggle
!touch ~/.kaggle/kaggle.json

api_token = {"username":"felipecadavidrincn","key":"d9b9503283bc4582046878ec6ea10ccf"}

import json

with open('/root/.kaggle/kaggle.json', 'w') as file:
    json.dump(api_token, file)

!chmod 600 ~/.kaggle/kaggle.json
```

3. Descarga de archivos para el modelo: Una vez completada la configuración inicial, se procede a la descarga de los archivos desde el servidor de Kaggle:

```
%%bash
export KAGGLE_CONFIG_DIR=~/.kaggle/
kaggle competitions download web-traffic-time-series-forecasting -f key_1.csv.zip
kaggle competitions download web-traffic-time-series-forecasting -f key_2.csv.zip
kaggle competitions download web-traffic-time-series-forecasting -f sample_submission_1.csv.zip
kaggle competitions download web-traffic-time-series-forecasting -f sample_submission_2.csv.zip
kaggle competitions download web-traffic-time-series-forecasting -f train_1.csv.zip
kaggle competitions download web-traffic-time-series-forecasting -f train_2.csv.zip
```

```
Downloading key_1.csv.zip to /content

Downloading key_2.csv.zip to /content

Downloading sample_submission_1.csv.zip to /content

Downloading sample_submission_2.csv.zip to /content

Downloading train_1.csv.zip to /content

Downloading train_2.csv.zip to /content

100%|██████████| 96.0M/96.0M [00:01<00:00, 72.7MB/s]
100%|██████████| 101M/101M [00:02<00:00, 50.6MB/s]
100%|██████████| 66.0M/66.0M [00:01<00:00, 45.0MB/s]
100%|██████████| 68.2M/68.2M [00:01<00:00, 45.7MB/s]
100%|██████████| 102M/102M [00:02<00:00, 42.1MB/s]
100%|██████████| 150M/150M [00:02<00:00, 60.0MB/s]
```

4. Habiendo terminado esto, podemos empezar a procesar los datos desde los archivos del proyecto seleccionado.

IMPORTACIÓN Y ANÁLISIS DE DATOS

El primer paso es importar los datos, para representar y realizar un entendimiento inicial de la posible información a representar. En síntesis, tenemos que:

- ❖ Es una base de datos de información que tabula visitas por página en el sitio wikipedia.
- ❖ La tabulación se basa en visitas por día calendario.

```
train.head()
```

	Page	2015-07-01	2015-07-02	2015-07-03	2015-07-04	2015-07-05	2015-07-06	2015-07-07	2015-07-08	2015-07-09	...
0	2NE1_zh.wikipedia.org_all-access_spider	18.0	11.0	5.0	13.0	14.0	9.0	9.0	22.0	26.0	...
1	2PM_zh.wikipedia.org_all-access_spider	11.0	14.0	15.0	18.0	11.0	13.0	22.0	11.0	10.0	...
2	3C_zh.wikipedia.org_all-access_spider	1.0	0.0	1.0	1.0	0.0	4.0	0.0	3.0	4.0	...
3	4minute_zh.wikipedia.org_all-access_spider	35.0	13.0	10.0	94.0	4.0	26.0	14.0	9.0	11.0	...
4	52_Hz_I_Love_You_zh.wikipedia.org_all-access_s...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...

5 rows × 551 columns

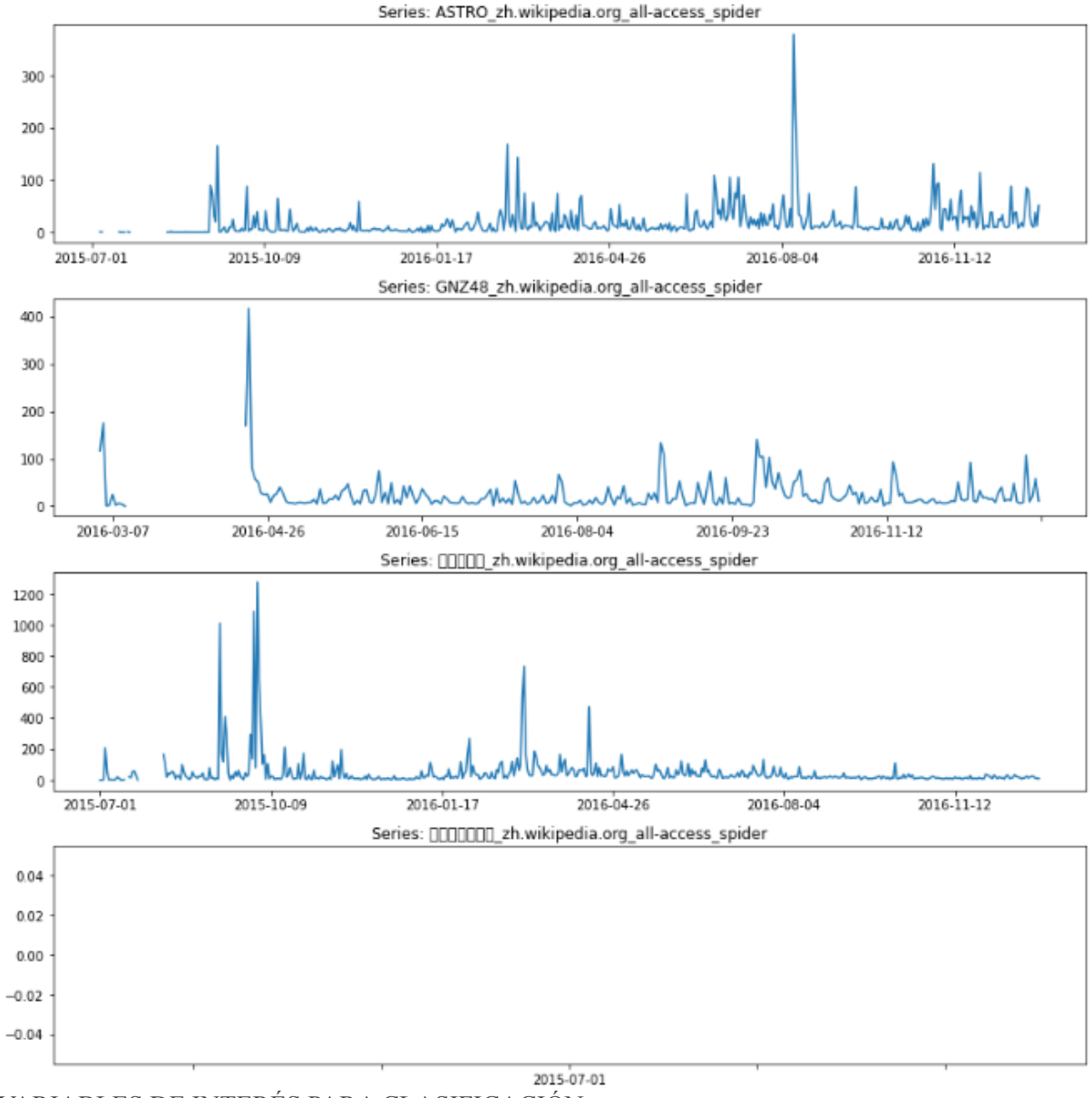
DATOS NULOS

Durante el análisis de los datos se identificó que existen columnas con datos faltantes representados con NaN. En este caso, se asume que los datos NaN corresponden a los días en los que la página no tuvo visitas. Durante el procesamiento de datos, estos serán reemplazados por ceros (0).

	Page	2015-07-01	2015-07-02	2015-07-03	2015-07-04	2015-07-05	2015-07-06	2015-07-07	2015-07-08	2015-07-09
4	52_Hz_I_Love_You_zh.wikipedia.org_all-access_s...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
6	91Days_zh.wikipedia.org_all-access_spider	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
10	ASTRO_zh.wikipedia.org_all-access_spider	NaN	NaN	NaN	NaN	NaN	1.0	1.0	NaN	NaN
13	AlphaGo_zh.wikipedia.org_all-access_spider	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
19	B-PROJECT_zh.wikipedia.org_all-access_spider	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
24	BLACK_PINK_zh.wikipedia.org_all-access_spider	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
27	Beautiful_Mind_zh.wikipedia.org_all-access_spider	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
37	Dear_My_Friends_zh.wikipedia.org_all-access_sp...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
38	Doctors_zh.wikipedia.org_all-access_spider	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
48	Fantastic_Duo_zh.wikipedia.org_all-access_spider	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
145053	Drake_(músico)_es.wikipedia.org_all-access_spider	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
145054	Skam_(serie_de_televisión)_es.wikipedia.org_al...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
145055	Legión_(serie_de_televisión)_es.wikipedia.org_...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
145056	Doble_tentación_es.wikipedia.org_all-access_sp...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
145057	Mi_adorable_maldición_es.wikipedia.org_all-acc...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
145058	Underworld_(serie_de_películas)_es.wikipedia.o...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
145059	Resident_Evil:_Capítulo_Final_es.wikipedia.org...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
145060	Enamorándome_de_Ramón_es.wikipedia.org_all-acc...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
145061	Hasta_el_último_hombre_es.wikipedia.org_all-ac...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
145062	Francisco_el_matemático_(serie_de_televisión_d...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Se definió inicialmente la organización de 64 columnas para el análisis de 62 días, donde las columnas representan visitas por día y las filas las 145063 paginas pertenecientes a wikipedia, que por lo que se tiene entendido representa cada una un tema diferente, es decir, por lo general no son páginas concatenadas en búsqueda de seguir un orden de rastreo, sino que son totalmente independientes entre

sí. Esto permite inferir que cada página presenta diferentes resultados de visualización por día. Esto significa que cada página debe indicar su predicción para cada uno de los 62 días, considerándose cada una de ellas una serie temporal diferente. Para evidenciar esto se ejemplifica con 4 páginas la dispersión obtenida tan solo con los datos NAN.



VARIABLES DE INTERÉS PARA CLASIFICACIÓN

Durante el análisis de los datos, y especialmente de la columna *page*, se identificó que otros parámetros están concatenados en *snake_case* dentro del nombre de cada página. Para esto, se utilizó un regex que permitiera separar estos atributos adicionales para poder identificarlos más fácilmente.

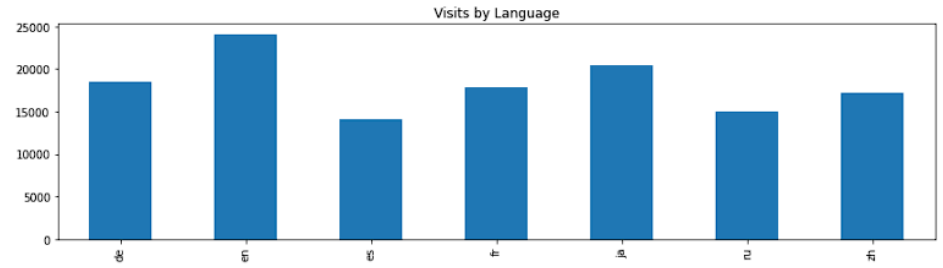
```
page_attributes = train.Page.str.extract(r'(?P<topic>.*)\_(?P<lang>.*).wikipedia.org\_(?P<access>.*)\_(?P<type>.*)')
page_attributes[0:10]
```

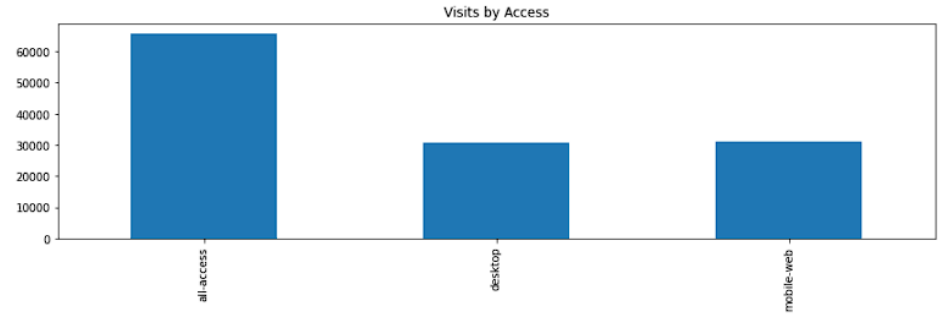
- Fecha: en la que se realiza la visualización de la página (Representado en varias columnas).
- Página: Nombre de identificación
- Topic: Titulo del articulo

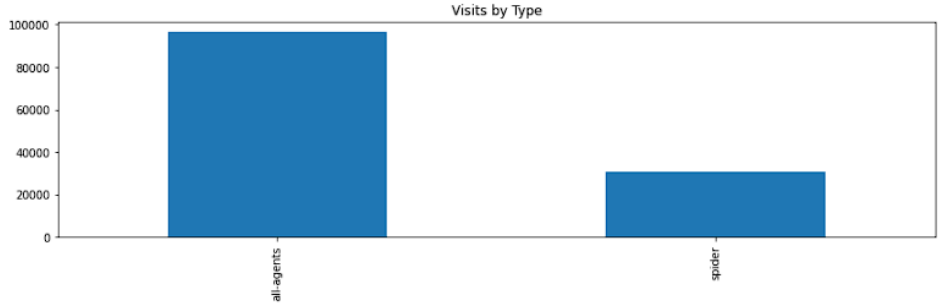
- Idioma: Idioma en el que se presenta el artículo
- Acceso: disposición de la página como puede ser vista por el usuario
- Tipo: origen de redirección de la página, es decir, si el usuario proviene de un motor de búsqueda o ingresó directamente a wikipedia.

	topic	lang	access	type
0	2NE1	zh	all-access	spider
1	2PM	zh	all-access	spider
2	3C	zh	all-access	spider
3	4minute	zh	all-access	spider
4	52_Hz_I_Love_You	zh	all-access	spider
5	5566	zh	all-access	spider
6	91Days	zh	all-access	spider
7	A'N'D	zh	all-access	spider
8	AKB48	zh	all-access	spider
9	ASCII	zh	all-access	spider

Se observaron 7 idiomas diferentes: inglés, alemán, francés, chino, ruso y español, lo que complica análisis de URL, ya que se debe trabajar con cuatro sistemas de escritura diferente. Además de 2 orígenes y 2 tipos de agentes.







GENERACIÓN DE ENTRENAMIENTO Y VALIDACIÓN DE DATOS

Dado al condicionamiento limitado de recursos, se simplifican los datos a una muestra que permita el análisis deseado para este reto, por lo que el training se simplifica a las primeras 5 paginas de la base de datos original, con 495 días de evaluación de visitas, que parten desde el primero de julio de 2015, hasta el primero de noviembre de 2016.

Por otro lado, se conservan el resto de datos para el análisis de validación, conservando las mismas 5 páginas del entrenamiento y variado los días de la toma de datos, los cuales parten del dos de noviembre del 2016, hasta el 31 de diciembre del 2016; con un total de 65 días.

MÉTRICA DE DESEMPEÑO

SMAPE:
Para el modelo, se tendrá en cuenta el modelo recomendado por el reto en Kaggle, en este caso el modelo SMAPE, la cual es una métrica de porcentual de regresión que veremos a continuación:

valores reales (yi) y los valores predichos (yi^).

$$SMAPE = (\frac{1}{N}\Sigma\frac{|y_i - \hat{y}_i|}{(|y_i| - |\hat{y}_i|)/2}) * 100$$

El SMAPE nos permite superar las limitaciones con la medición del error de pronóstico MAPE. En contraste con el error porcentual absoluto medio, SMAPE tiene un límite inferior y un límite superior. Por lo tanto, se lo conoce como simétrico, esta métrica porcentual nos facilita medir el error de manera escalonada, es decir, se busca acotar el error entre valores de 0 a 1, donde 0 significa que el ajuste es perfecto, mientras que 1 sería un mal ajuste. Cabe destacar que muchas veces las métricas porcentuales pueden tener valores mayores a 1.

```
def smape_weirdness_fix(prediction_df, actual_df):
    actual_df = actual_df.fillna(0)
    prediction_df = prediction_df.fillna(0)

    denominator = (np.abs(actual_df) + np.abs(prediction_df)) / 2.0
    diff = np.abs(actual_df - prediction_df) / denominator
    diff[denominator == 0] = 0.0
    return np.nanmean(diff)
```

MODELOS DE PREDICCIÓN:

MEDIAN:
Algoritmo automático basado en la mediana con el enfoque de la mediana propuesto generando varios conjuntos de datos de la distribución.

```python
def nanmedian_zero(a):
    return np.nan_to_num(np.nanmedian(a))

def median_model(df_train, df_actual, p, review=False, figSize=(12, 4)):
    df_train = df_train.fillna(0)
    df_actual = df_actual.fillna(0)
    visits = nanmedian_zero(df_train['y'].values[-p:])
    train_series = df_train['y']
    train_series.index = df_train.ds

    idx = np.arange( p) + np.arange(len(df_train)- p+1)[:,None]
    b = [row[row>=0] for row in df_train.y.values[idx]]
    pre_forecast = pd.Series(np.append(([float('nan')] * (p-1)), list(map(nanmedian_zero,b))))
    pre_forecast.index = df_train.ds

    forecast_series = pd.Series(np.repeat(visits, len(df_actual)))
    forecast_series.index = df_actual.ds

    forecast_series = pre_forecast.append(forecast_series)

    actual_series = df_actual.y
    actual_series.index = df_actual.ds

    if(review):
        plot_prediction_and_actual(train_series, forecast_series, actual_series, figSize=figSize, title='Median model')

    return smape_weirdness_fix(forecast_series, actual_series)
```

ARIMA:

El modelo autoregresivo (AR): Considera que el valor de la serie estacionaria en el tiempo presente t depende de todos los valores pasados que ha tomado la serie, ponderados por un factor de peso πj que mide la influencia de ese valor pasado en el valor presente; y de una perturbación aleatoria presente. Cuando solamente los últimos p valores pasados de la serie afectan significativamente el valor presente, el modelo se denomina autorregresivo de orden p, AR (p).

El modelo media móvil (MA): Considera que el valor de la serie estacionaria oscila o se desplaza alrededor de un valor medio μ. Además supone que el desplazamiento de μ en el tiempo presente t es ocasionado por infinitas perturbaciones ocurridas en el pasado, ponderados por un factor Ψj , que mide la influencia de dicha perturbación en el presente de la serie. Cuando sólo las últimas perturbaciones pasadas afectan significativamente el valor presente de la serie.

Estos dos modelos básicos para series estacionarias se combinan para producir los modelos ARMA (p, q). En general las series de tiempo no son estacionarias pero por medio de transformaciones de varianza y de diferencias pueden ser transformadas en estacionarias. Los modelos ARIMA resultan al integrar a la serie estacionaria ARMA (p,q) estimada, las diferencias y las transformaciones que fueron necesarias para convertir la serie inicial en una serie estacionaria.

Las series de tiempo y en especial los modelos Autorregresivos e Integrados con Promedios Móviles (ARIMA), resultan realmente apropiados para modelar el tráfico moderno con características de correlación fuertes en la modelación del tráfico.

$$Y_t = \beta_1 + \Phi_1 Y_{t-1} + \Phi_2 Y_{t-2} + \ldots + \Phi_p Y_{t-p}$$

$$\text{ARIMA}\ \underbrace{(p,d,q)}_{\substack{\uparrow \\ \text{Non-seasonal part} \\ \text{of the model}}}\ \underbrace{(P,D,Q)_m}_{\substack{\uparrow \\ \text{Seasonal part} \\ \text{of the model}}}$$

```
from statsmodels.tsa.arima_model import ARIMA
import warnings

def arima_model(df_train, df_actual, p, d, q, figSize=(12, 4), review=False):
    df_train = df_train.fillna(0)
    train_series = df_train.y
    train_series.index = df_train.ds

    result = None
    with warnings.catch_warnings():
        warnings.filterwarnings('ignore')
        try:
            arima = ARIMA(train_series ,[p, d, q])
            result = arima.fit(disp=False)
        except Exception as e:
            print('\tARIMA failed', e)

    start_idx = df_train.ds[d]
    end_idx = df_actual.ds.max()
    forecast_series = result.predict(start_idx, end_idx,typ='levels')

    actual_series = df_actual.y
    actual_series.index = pd.to_datetime(df_actual.ds)

    if(review):
        plot_prediction_and_actual(train_series, forecast_series, actual_series, figSize=figSize, title='ARIMA model')

    return smape_weirdness_fix(forecast_series, actual_series)
```

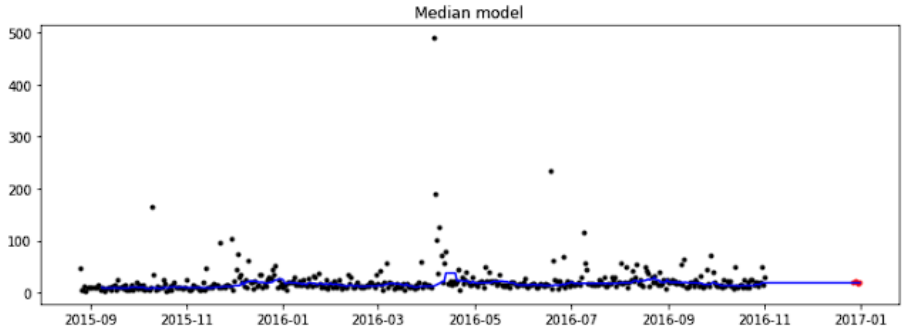
APLICACIÓN DE MÉTRICA DE DESEMPEÑO.

Se establece la aplicación de SMAPE tomando como partida tres páginas a las cuales se les aplica los dos diferentes métodos de predicción, con una muestra de datos de 60 días para cada una de ellas, en las que se obtiene para la primera que el mejor método de predicción, Median con un SMAPE de 0.06, es decir, una probabilidad de acierto del 90% versus el 64% de Arima , para el segundo comparativo se obtuvo que Arima obtiene una mayor probabilidad de acierto en un 66% vs y para el tercero 57% de Median, y para el tercer comparativo ambos métodos presentan un valor casi igual de precisión, con un relativo 47%.

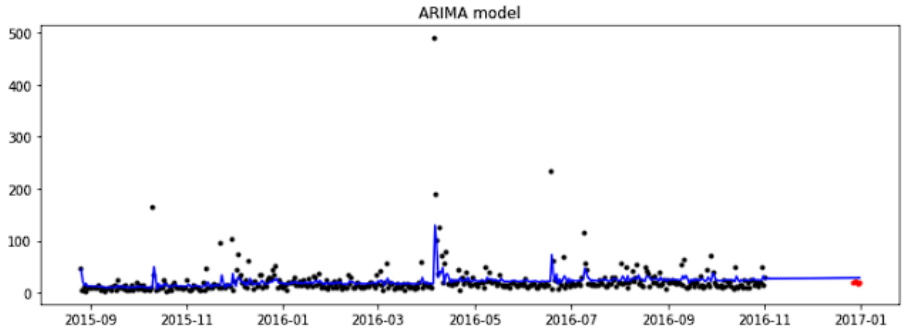
```
def get_better_smape(median_model_score, arima_model_score):
  if median_model_score < arima_model_score:
    result = arima_model_score - median_model_score
    print("The MEDIAN model is better by less than: %.5f" % result)
  elif median_model_score > arima_model_score:
    result = median_model_score - arima_model_score
    print("The ARIMA model is better by less than: %.5f" % result)
  else:
    print("Both results match")
```

```
test1_df_train = series_sample_extractor(X_train, 0, 60)
test1_df_actual = series_sample_extractor(y_train, 0, 60)

score_median = median_model(test1_df_train.copy(), test1_df_actual.copy(), 15, review=True)
score_arima = arima_model(test1_df_train.copy(), test1_df_actual.copy(), 2, 1, 2, review=True)


print("The SMAPE score for MEDIAN is : %.5f" % score_median)
print("The SMAPE score for ARIMA is : %.5f" % score_arima)
get_better_smape(score_median, score_arima)
```

1)

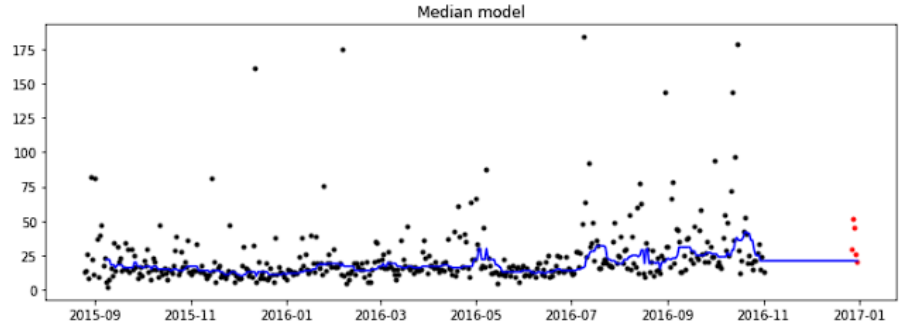



```
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/base/tsa_model.py:132: FutureWarning:
  date_key = Timestamp(key, freq=base_index.freq)
```

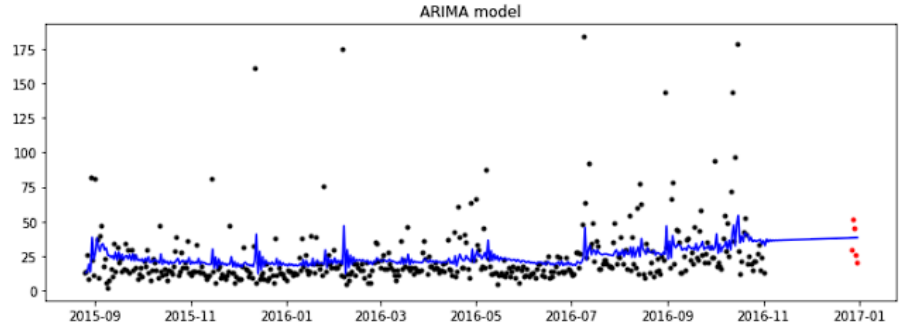



```
The SMAPE score for MEDIAN is : 0.06059
The SMAPE score for ARIMA is : 0.36815
The MEDIAN model is better by less than: 0.30756
```

2)

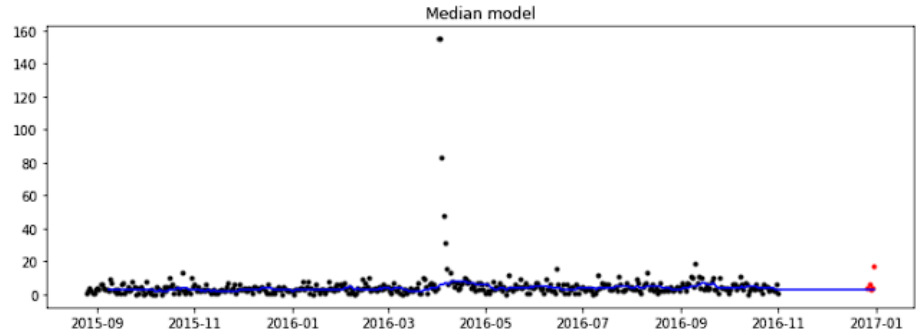



```
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/base/tsa_model.py:132: FutureWarning:
  date_key = Timestamp(key, freq=base_index.freq)
```

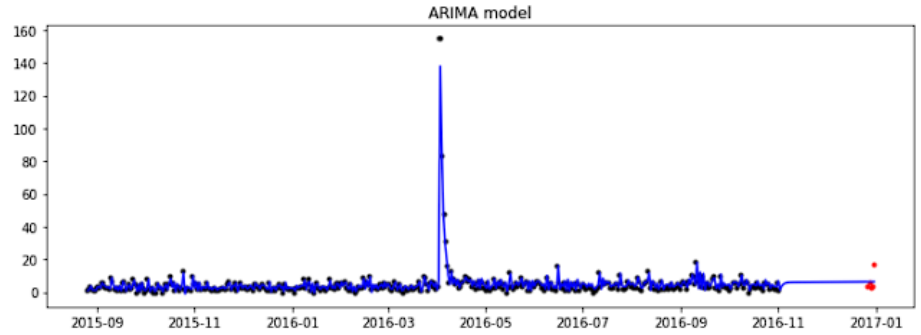



```
The SMAPE score for MEDIAN is : 0.43822
The SMAPE score for ARIMA is : 0.34340
The ARIMA model is better by less than: 0.09481
```

3)




```
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/base/tsa_model.py:132: FutureWarning:
  date_key = Timestamp(key, freq=base_index.freq)
```




```
The SMAPE score for MEDIAN is : 0.52762
The SMAPE score for ARIMA is : 0.52292
The ARIMA model is better by less than: 0.00470
```

4. RETOS Y CONSIDERACIONES DE DESPLIEGUE

VALORES NULOS

Como se mencionó anteriormente, se identificó que algunas entradas tienen valores NaN para cada registro. Inclusive, una cantidad mínima de registros presentaron datos NaN para todas y cada una de las fechas:

	Page	2015-07-01	2015-07-02	2015-07-03	2015-07-04	2015-07-05	2015-07-06	2015-07-07	2015-07-08	2015-07-09
4295	李宏毅_zh.wikipedia.org_all-access_spider	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4296	迪玛希·库达依别列根_zh.wikipedia.org_all-access_spider	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4297	7日羅曼史_zh.wikipedia.org_all-access_spider	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4298	漫漫回家路_(2016年電影)_zh.wikipedia.org_all-access_sp...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4299	為了與你相遇_(電影)_zh.wikipedia.org_all-access_spider	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4300	小林家的龍女僕_zh.wikipedia.org_all-access_spider	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4301	曹晏豪_zh.wikipedia.org_all-access_spider	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4302	徐鈞浩_zh.wikipedia.org_all-access_spider	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4303	生化危機7_惡靈古堡_zh.wikipedia.org_all-access_spider	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4304	A_LIFE～深愛的人～_zh.wikipedia.org_all-access_spider	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Estos registros no fueron tenidos en cuenta.

En el caso de los registros que tenían algunas fechas en NaN, se procesaron como ceros, asumiendo que esas entradas no tuvieron visitas en estas fechas. Si bien no es un reto importante para la manipulación de datos, no es posible determinar con precisión si realmente la ausencia de datos en estas fechas corresponde a que no hubo visitas a las páginas respectivamente.

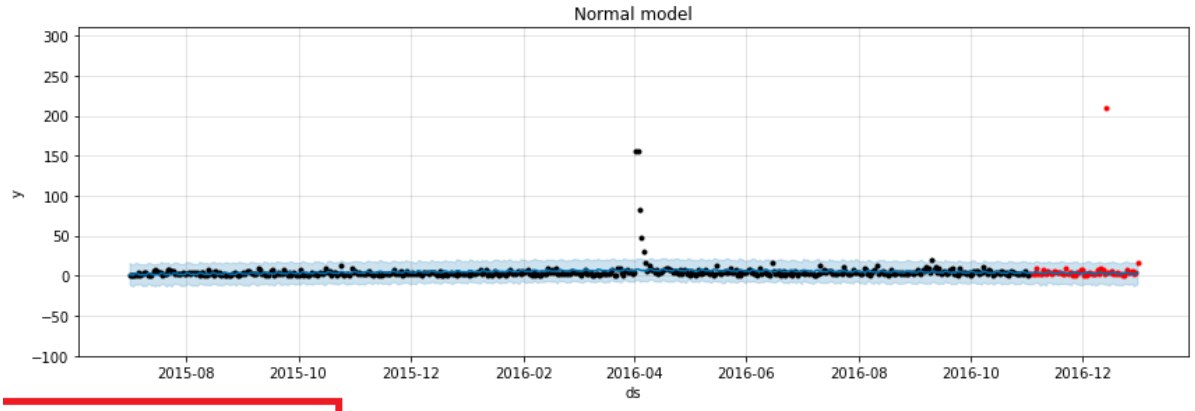
```
df_train = df_train.fillna(0)
df_actual = df_actual.fillna(0)
```

IMPLEMENTACIÓN DE ALGUNOS MODELOS COMO FACEBOOK PROPHET

Durante la búsqueda e implementación de modelos de predicción, hubo uno muy interesante desarrollado por Facebook. Prophet se usa en muchas aplicaciones en Facebook para producir pronósticos confiables para la planificación y el establecimiento de objetivos. Usualmente se ha encontrado que funciona mejor que cualquier otro enfoque en la mayoría de los casos, ya que es susceptible a variables como feriados, temporada alta y baja etc.

Este modelo prometia ser bastante eficiente pero el algoritmo presentaba fallas porque no computaba los valores de predicción y actuales cuando se su precisión a través de SMAPE y por lo tanto siempre retornando un valor nulo

```
[1 rows x 551 columns]
INFO:fbprophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True to override this.
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
```




```
The SMAPE score is : nan
/usr/local/lib/python3.7/dist-packages/pandas/core/generic.py:8870: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

VOLUMEN DE PROCESAMIENTO DE DATOS

Hubo algunas complicaciones iniciales al definir el volumen de los datos. Inicialmente se tenía pensado tomar una mayor cantidad de datos para la predicción, pero la herramienta tiene limitantes de memoria y espacio. En este caso particular, la memoria no fue suficiente para incrementar la cantidad de datos que se podían pasar a cada modelo y fue necesario reducir el volumen de procesamiento

5. CONCLUSIONES

DE LOS RESULTADOS OBTENIDOS

- Al realizar las pruebas con los distintos métodos predictivos, podemos comprobar que no hay uno solo que sea el mejor, debido a la variabilidad que hay en cada muestra tomada. Para esto

sería necesario implementar o analizar la generación de un modelo de predicción preciso para cada una de las series temporales que conforman el dataset, con el objetivo de obtener un SMAPE más próximo a 0, y no la variabilidad tan grande entre resultados, en los que por lo general se obtienen valores próximos al 0.4, es decir, que el modelo tiene una probabilidad de acierto del 60%, el cual puede ser útil al tratarse del pronóstico de visitas de una página web, un tema que no necesita tanta precisión como lo sería un modelo de predicción médico.
Sin embargo si se desea tener un valor de confiabilidad mayor de pronto para el análisis de sitios web más propicios para espacios publicitarios ó control de accesos, ambos métodos no son viables para generalizarse en todo el proceso.

- De las 5 pruebas realizadas, el modelo ARIMA fue el más favorecido, sin embargo, se requieren pruebas más extensivas y con volúmenes de datos más grandes para obtener resultados más confiables.

DE LOS MODELOS Y LOS DATOS ANALIZADOS

- Si bien se identificaron y se agregaron atributos a cada registro, como idioma, agente y tipo, estos no fueron tenidos en cuenta para hacer pruebas más extensivas de los modelos presentados. Esto representa una oportunidad para futuras iteraciones en las que se pueden utilizar representaciones de datos en base a estos atributos adicionales e identificar patrones para la predicción de series temporales.

REFERENCIAS

Anapedia. (n.d.). *Understand advanced metrics - Anaplan Technical Documentation*. Anapedia. Retrieved July 5, 2022, from https://help.anaplan.com/685ff9b2-6370-46ba-af10-679405937113-Understand-advanced-metrics

GeeksforGeeks. (2021, November 28). *How to Calculate SMAPE in Python?* GeeksforGeeks. Retrieved July 5, 2022, from https://www.geeksforgeeks.org/how-to-calculate-smape-in-python/

Indeed Editorial Team. (2021, October 21). *How To Use the SMAPE Formula (4 Methods With Examples)*. Indeed. Retrieved July 5, 2022, from https://www.indeed.com/career-advice/career-development/smape-formula

Villanueva Ocampo, B., López Sarmiento, D., & Rivas Trujillo, E. (2012, Julio- Diciembre). Métodos de modelamiento y predicción de tráfico orientados a plataformas de transmisión de video e IPTV usando series de tiempo. *REVISTA CIENTIFICA*, (16), 11-21.

Web Traffic Time Series Forecasting. (2017, Julio 13). Web Traffic Time Series Forecasting | Kaggle. Retrieved July 5, 2022, from https://www.kaggle.com/competitions/web-traffic-time-series-forecasting/data