Modelo de Machine Learning para Predicción UrbanData Taxis

Dataset de Entrenamiento

A partir de nuestro dataset preprocesado de Taxi Datatrips, que contiene información como fecha y localización de subida y bajada de pasajeros, cantidad de pasajeros, distancia recorrida, cobro total y otras tarifas y banderas, se desea construir un modelo de serie tiempo para la predicción de las características más relevantes para nuestro cliente

Para esto, se toman las siguientes consideraciones:

- Columnas como 'passenger_count', 'tip_amount', 'location' y otras tarifas y banderas no son relevantes y podemos prescindir de ellas
- para mantener la serie de tiempo del modelo, podemos agrupar la data importante de forma diaria, a partir de la fecha de subida de pasajeros
- Las columnas esenciales de data para entrenamiento y predicción serán 'trip_distance' y 'total_amount'

Por lo tanto se decide construir a partir de esta data original otra tabla llamada "**auxiliar_diario.csv**" que servirá para entrenar el modelo y realizar las predicciones. Esta tabla tendrá las siguientes columnas extraidas de la data original (detalles en 'EDA_ML.ipynb' en Carpeta 'Notebook' del Repositorio)

año	Int	Año del Registro
mes	Int	Mes del Registro
dia	Int	Día del Registro
taxi	Str	Tipo de Taxi (green o yellow)
total_viajes	Float	Total de viajes realizados durante ese día por ese tipo de taxi
distancia_total	Float	Total distancia recorrida ese dia por tipo de taxi en millas
distancia_prom	Float	Distancia promedio por viaje para ese día y taxi en millas
tarifa_total	Float	Tarifa total recolectada por dia y taxi en US Dollars
tarifa_prom	Float	Tarifa promedio por viaje para ese dia y taxi en US Dollars

Preview del dataset "auxiliar_diario.csv"

	año	mes	dia	taxi	total_viajes	distancia_total	distancia_prom	tarifa_total	tarifa_prom
0	2019	1	1	green	15433	59278.51	3.841023	269440.34	17.458714
1	2019	1	2	green	19900	79372.41	3.988563	364589.71	18.321091
2	2019	1	3	green	21931	85921.28	3.917800	401273.70	18.297100
3	2019	1	4	green	23123	88613.15	3.832251	416698.39	18.020948
4	2019	1	5	green	20810	71967.18	3.458298	337079.39	16.197952
...	...	...	...	...	...	...	...	...	...
725	2019	12	27	yellow	169613	539740.85	3.182190	3310285.14	19.516695
726	2019	12	28	yellow	172026	552390.33	3.211086	3321643.82	19.308964
727	2019	12	29	yellow	165125	557714.53	3.377529	3233992.67	19.585118
728	2019	12	30	yellow	174562	535018.10	3.064917	3319196.06	19.014425
729	2019	12	31	yellow	169191	496247.29	2.933060	3068026.90	18.133511

730 rows × 9 columns

El Producto de ML será una función desplegada en StreamLit que permitirá al usuario hacer predicciones con series de tiempo automáticas, de alcance personalizado, tanto para taxi verde como amarillo, sobre cualquiera y todas de las columnas de información relevante (total_viajes, distancia_total, distancia_prom, tarifa_total y tarifa_prom)

Entrenamiento y Validación

El Modelo escogido para este proyecto será Prophet de Meta, por su buen rendimiento en series de tiempo y relativamente sencilla implementación para datos simples y numéricos como los que disponemos

Utilizando librerías propias de Prophet, se realiza una **validación cruzada** para medir la precisión de nuestro modelo. Esto quiere decir, se ‘esconden’ ciertos valores conocidos para contrastarlos con las predicciones hechas por el modelo y cuantificar su performance.

Algunas métricas de rendimiento para proyecciones de distinto horizonte de la predicción de **’distancia_prom’** de ‘**yellow’** taxis

horizon	mse	rmse	mae	mape	mdape	smape	coverage
3 days	0.023878	0.154526	0.116443	0.037367	0.030182	0.037342	0.696970
4 days	0.026063	0.161441	0.121272	0.038935	0.031547	0.038848	0.696970
5 days	0.024124	0.155319	0.112076	0.036142	0.024905	0.035913	0.727273
6 days	0.013716	0.117113	0.092834	0.030851	0.026966	0.030435	0.848485
7 days	0.015812	0.125747	0.103058	0.034334	0.027526	0.033811	0.787879
8 days	0.020580	0.143458	0.116166	0.038533	0.027526	0.037870	0.757576
9 days	0.024184	0.155512	0.117677	0.039314	0.023071	0.038470	0.696970
10 days	0.021554	0.146814	0.107722	0.035704	0.023071	0.035002	0.727273

Para los valores de mse, rmse, mae, mape, mdape y smape un valor cercano a 0 indica un buen rendimiento

Para mejorar aún más dichas métricas, se realiza una búsqueda por grilla (**Grid Search**) de **hyperparámetros** a ajustar que mejoran la predicción de nuestro Modelo, testeando la mejor combinación entre todos ellos.

```
param_grid = {
    'changepoint_prior_scale': [0.01, 0.05, 0.1],
    'seasonality_prior_scale': [5.0, 10.0, 20.0],
    'seasonality_mode': ['additive', 'multiplicative'],
    'fourier_order': [5, 10, 20],
    'add_weekly_seasonality': [True,False]
}
```

Observamos que luego de la búsqueda,testeo y modificación de hyperparámetros aplicados al modelo, este retorna un puntaje aún mejor en las métricas de performance

horizon	mse	rmse	mae	mape	mdape	smape	coverage
3 days	0.023079	0.151918	0.111842	0.035797	0.026370	0.035875	0.727273
4 days	0.025386	0.159331	0.115280	0.036949	0.026370	0.036948	0.666667
5 days	0.023487	0.153254	0.108323	0.034873	0.020717	0.034739	0.696970
6 days	0.013426	0.115868	0.091356	0.030274	0.028973	0.029963	0.757576
7 days	0.015653	0.125112	0.103918	0.034456	0.029095	0.034066	0.757576
8 days	0.020556	0.143374	0.118637	0.039123	0.029095	0.038603	0.727273
9 days	0.024042	0.155056	0.119249	0.039627	0.021982	0.038907	0.696970
10 days	0.021498	0.146623	0.108330	0.035735	0.021337	0.035136	0.757576

Con estas mejoras, concluimos que el Modelo desarrollado, aunque imperfecto, otorga un buen pronóstico de las columnas de interés, y está listo para desplegarse en StreamLit para su uso por el cliente, con posibilidades de mejora futura en la implementación de hyperparámetros específicamente adecuados a cada predicción deseada y mejorar aún más el rendimiento del Modelo.