MlFlow & Feast

Laboratorio 3

Configuración del entorno

Verificación de la instalación

```
● (.venv) [diggspapu@diggsy feature_repo]$ feast materialize 2025-10-01T00:00:00 2025-10-19T23:59:59
/home/diggspapu/lab-3-mlops/.venv/lib/python3.10/site-packages/feast/infra/online_stores/sqlite.py:91: UserWarning: Field name "vector_enabled" in "SqliteOnlineStoreCo
nfig" shadows an attribute in parent "VectorStoreConfig"
  class SqliteOnlineStoreConfig(FeastConfigBaseModel, VectorStoreConfig):
/home/diggspapu/lab-3-mlops/.venv/lib/python3.10/site-packages/feast/infra/online_stores/sqlite.py:91: UserWarning: Field name "vector_len" in "SqliteOnlineStoreConfig
" shadows an attribute in parent "VectorStoreConfig"
  class SqliteOnlineStoreConfig(FeastConfigBaseModel, VectorStoreConfig):
Materializing 3 feature views from 2025-09-30 17:00:00-07:00 to 2025-10-19 16:59:59-07:00 into the sqlite online store.

driver_hourly_stats:
100%|█████████████████████████████████████| 5/5 [00:00<00:00, 626.11it/s]
driver_hourly_stats_fresh:
100%|█████████████████████████████████████| 5/5 [00:00<00:00, 810.68it/s]
house_features:
100%|█████████████████████████████████| 20640/20640 [00:03<00:00, 5597.94it/s]
● (.venv) [diggspapu@diggsy feature_repo]$
```

```python
features = store.get_online_features(
    features=[
        "house_features:MedInc",
        "house_features:HouseAge",
        "house_features:AveRooms",
        "house_features:AveBedrms",
        "house_features:Population",
        "house_features:AveOccup",
        "house_features:Latitude",
        "house_features:Longitude",
        "house_features:MedPP",
        "house_features:RatioRoomsBed",
        "house_features:HouseP",
    ],
    entity_rows=entity_df.to_dict(orient="records")
).to_df()

features.head()
```
✓ 7.6s

	house_id	AveRooms	HouseAge	MedPP	HouseP	AveOccup	AveBedrms	MedInc	Longitude	Latitude	RatioRoomsBed	Population
0	0	6.984127	41.0	3.257687	822.888889	2.555556	1.023810	8	-122.23	37.88	0.146591	322.0
1	1	6.238137	21.0	3.934608	5065.730228	2.109842	0.971880	8	-122.22	37.86	0.155797	2401.0
2	2	8.288136	52.0	2.589838	1389.920904	2.802260	1.073446	7	-122.24	37.85	0.129516	496.0
3	3	5.817352	52.0	2.214765	1421.753425	2.547945	1.073059	5	-122.25	37.85	0.184458	558.0
4	4	6.281853	52.0	1.763125	1232.528958	2.181467	1.081081	3	-122.25	37.85	0.172096	565.0

```python
for model in ['linear_regression', 'random_forest_regression','gradient_boosting_regressor']:
    print(f"____{model}____")
    # Cargar modelo desde MLflow
    model_uri = f"models:/{model}/None" # Ajusta el nombre si es diferente
    model = mlflow.sklearn.load_model(model_uri)

    # Seleccionar columnas de entrada
    X = features[[
        "MedInc", "HouseAge", "AveRooms", "AveBedrms",
        "Population", "AveOccup", "Latitude", "Longitude",
        "MedPP", "RatioRoomsBed", "HouseP"
    ]]

    # Hacer predicciones
    predictions = model.predict(X)

    # Mostrar resultados
    for house_id, pred in zip(features["house_id"], predictions):
        print(f"Predicción para house_id {house_id}: {pred}")
```
[36] ✓ 1.2s

```
____linear_regression____
Predicción para house_id 0: 4.261807704929353
Predicción para house_id 1: 4.736701228770947
Predicción para house_id 2: 3.785234637398119
Predicción para house_id 3: 3.2075470718407146
Predicción para house_id 4: 2.655767865840474
Predicción para house_id 5: 2.8644450900146055
Predicción para house_id 6: 2.611333644404887
```