



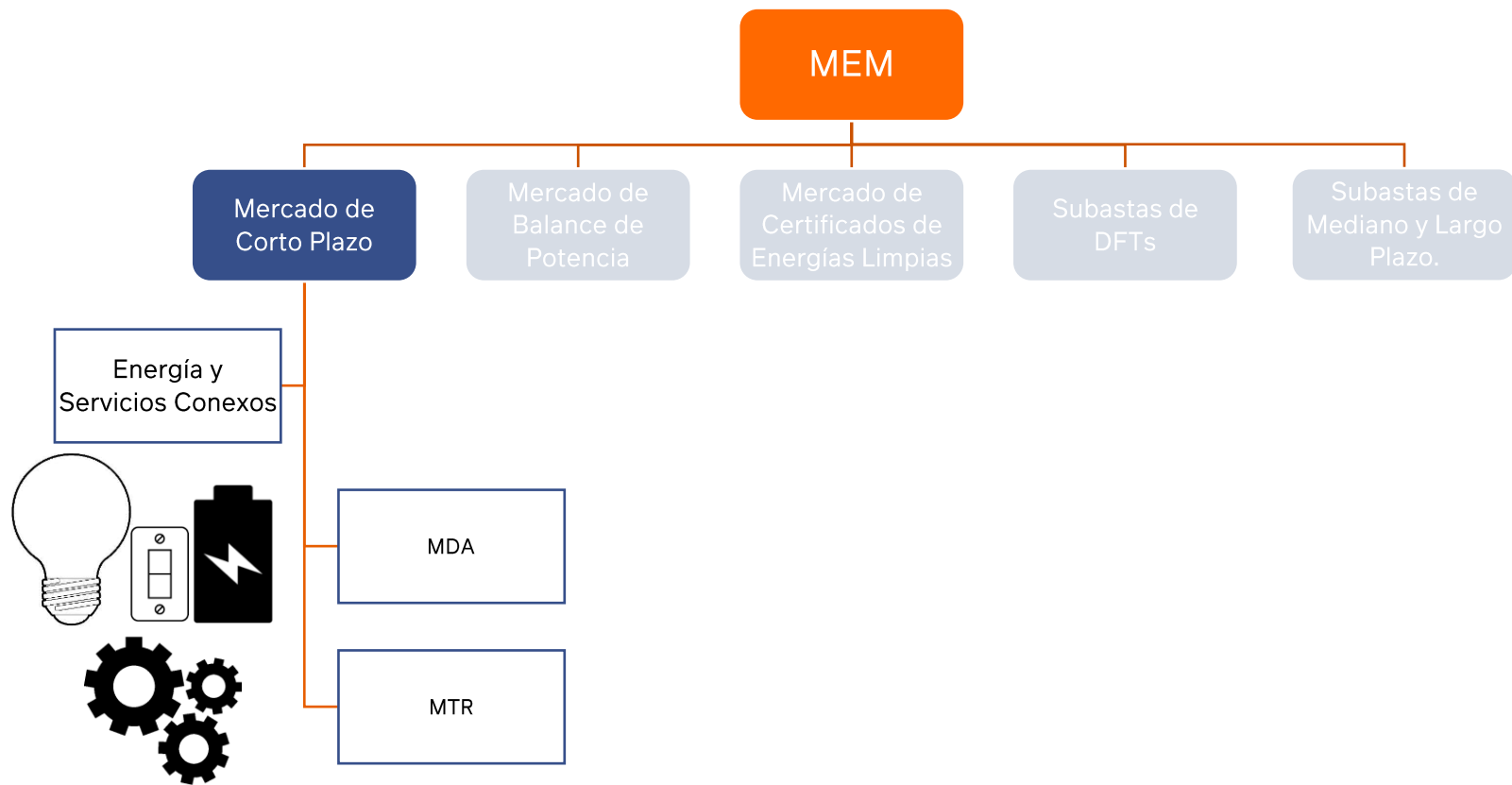
Data Translator

Proyecto Final

**José Jiménez
Gonzalez**

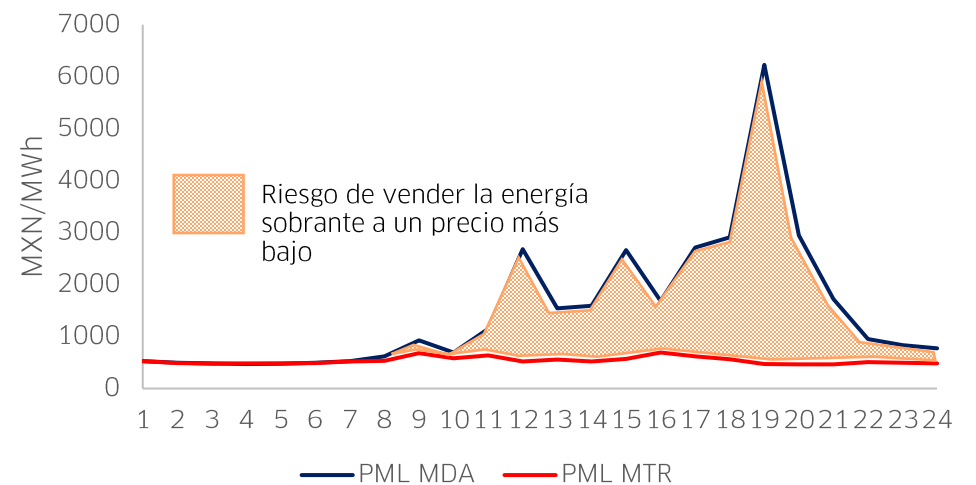
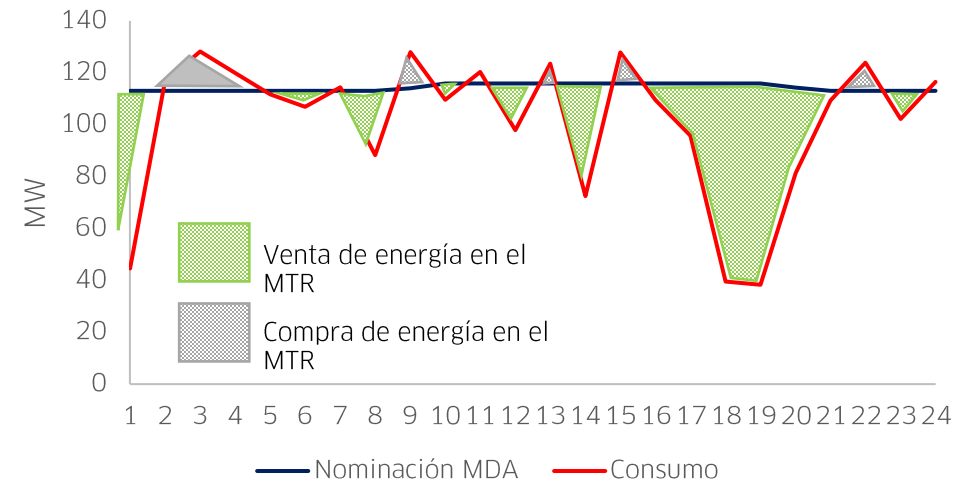
Iván López García

Operación del Mercado Eléctrico



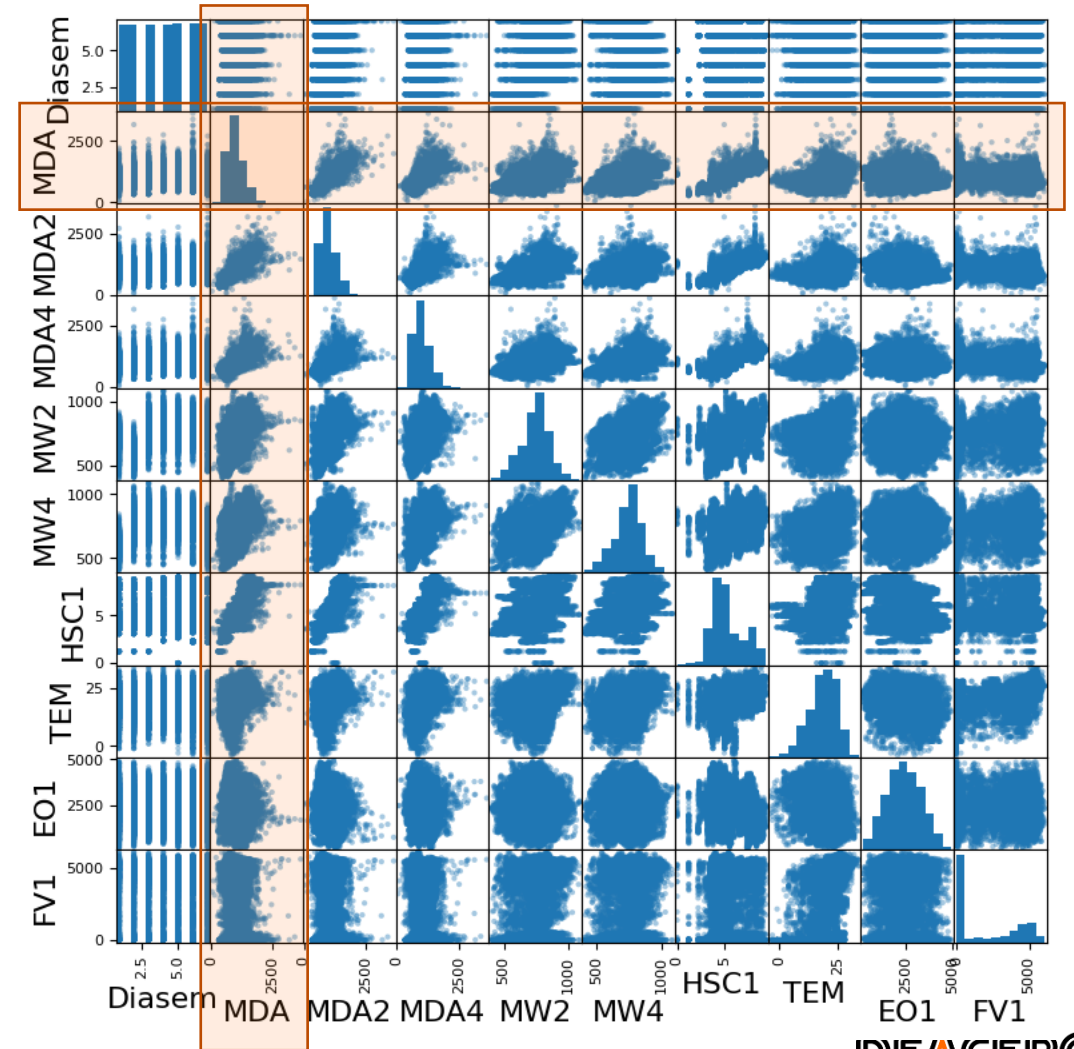
Importancia en la precisión de la nominación

- Independientemente del consumo, la **energía nominada en el MDA** se compra a **precio del MDA**.
- Los **desbalances se liquidan en el MTR**. Si en una hora se consumió más de lo nominado, esta energía adicional se compra automáticamente a precio del MTR. En caso contrario, esta energía se vende a precio del MTR.
- Tener desviaciones entre nominación y consumo implicaría **asumir el riesgo de pérdidas económicas** causadas por la diferencia de precios entre mercados.
- Existen horas del día donde existe una mayor volatilidad (por lo general, de las 19 a las 24 hrs), por lo que existe un mayor riesgo a pérdidas si se tienen desviaciones entre el consumo y la nominación.

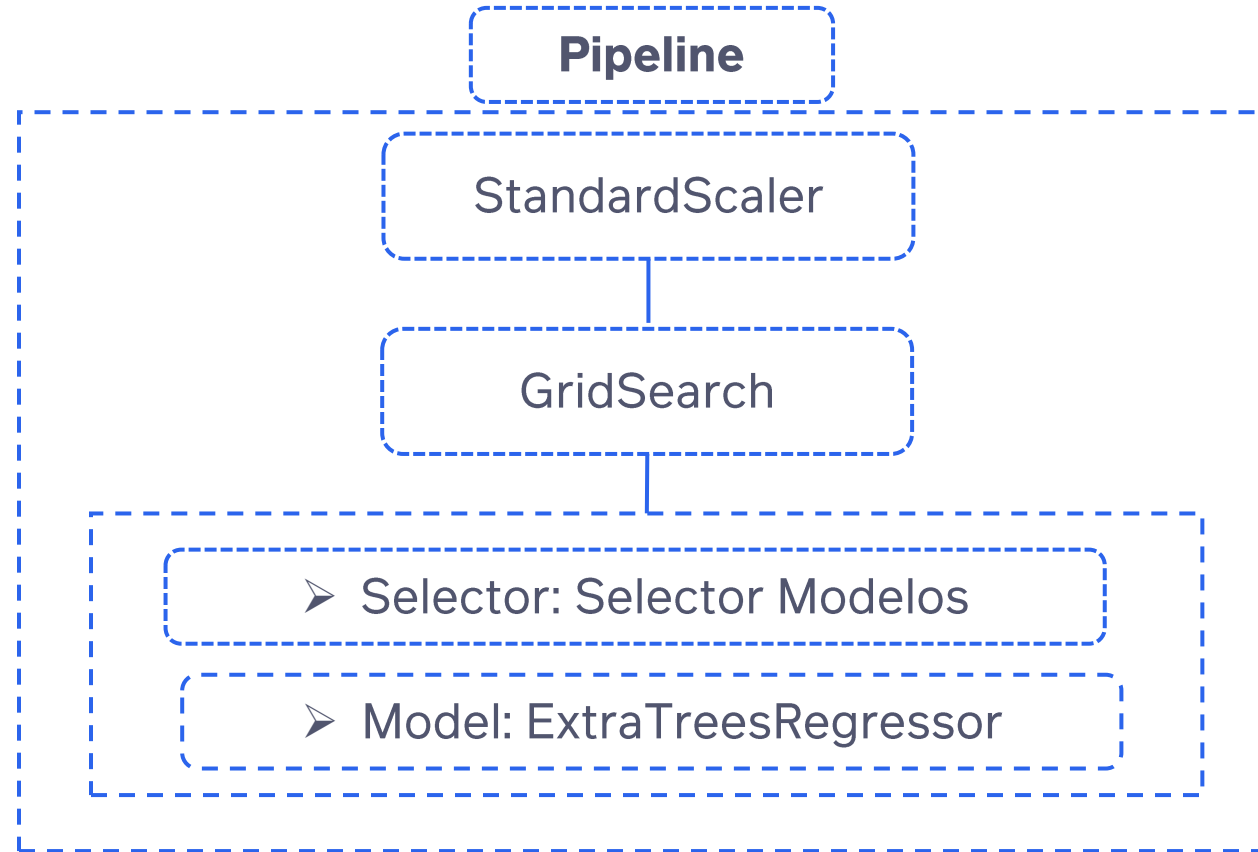


Conjunto de datos

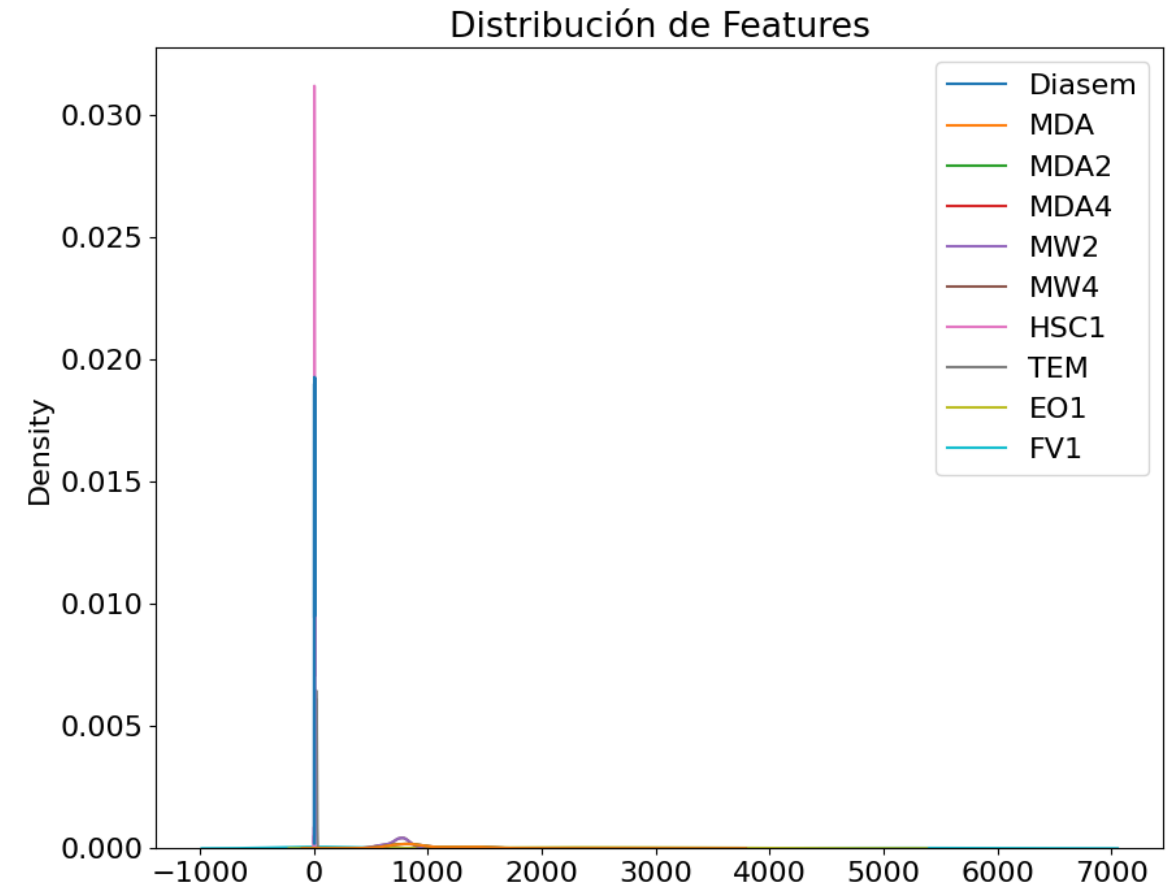
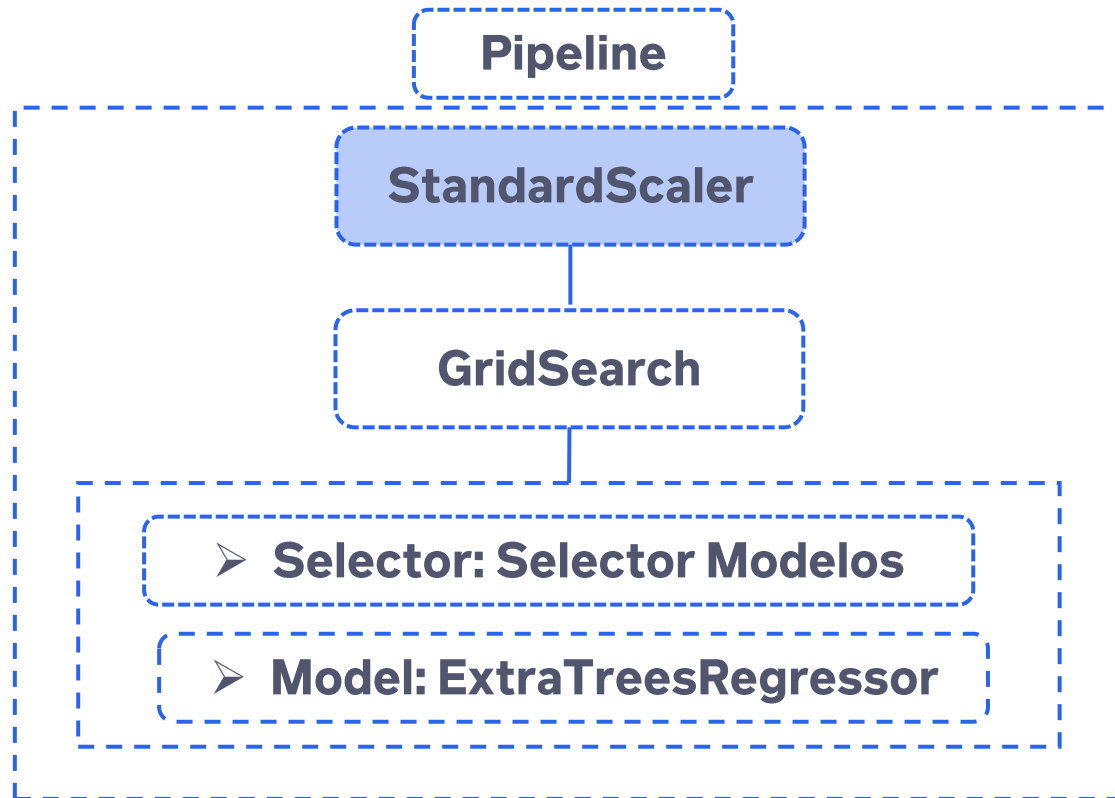
- El conjunto de datos está formado por:
 - 9 variables numéricas
 - 1 variable de fecha
 - 1 target



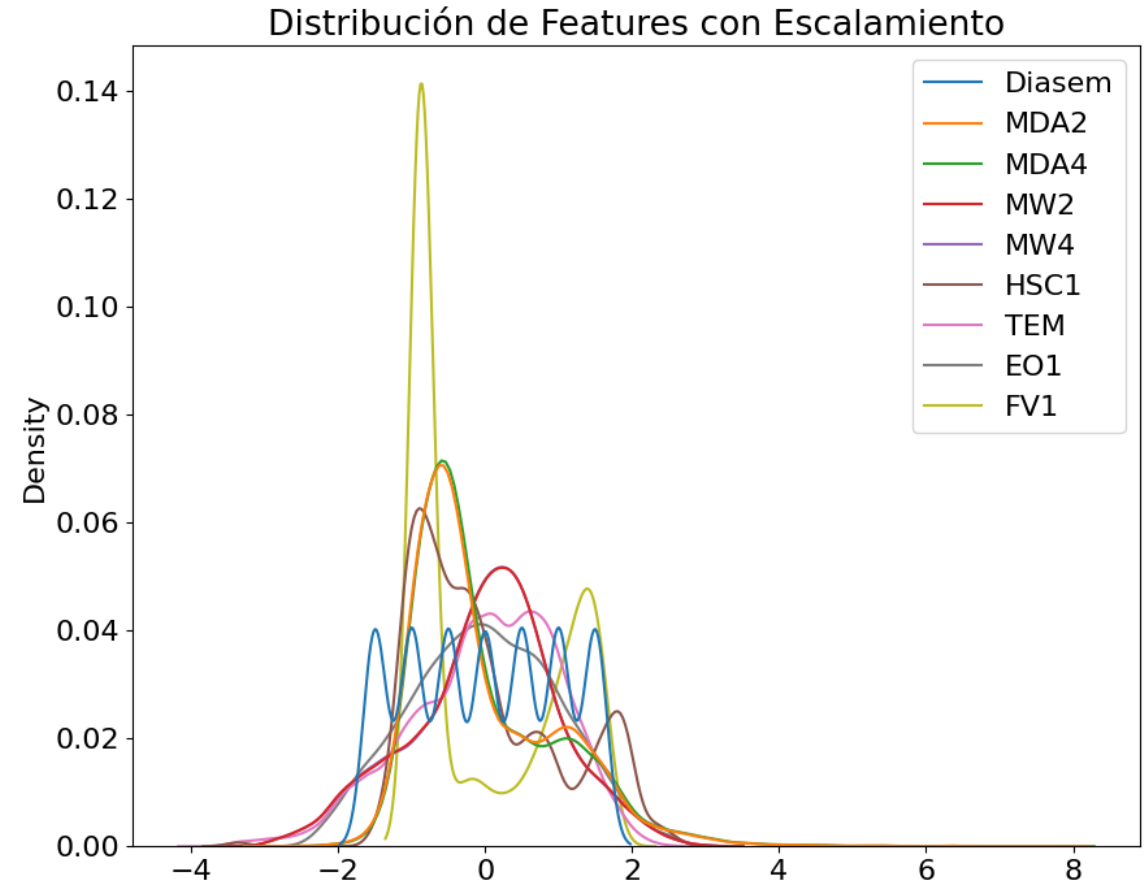
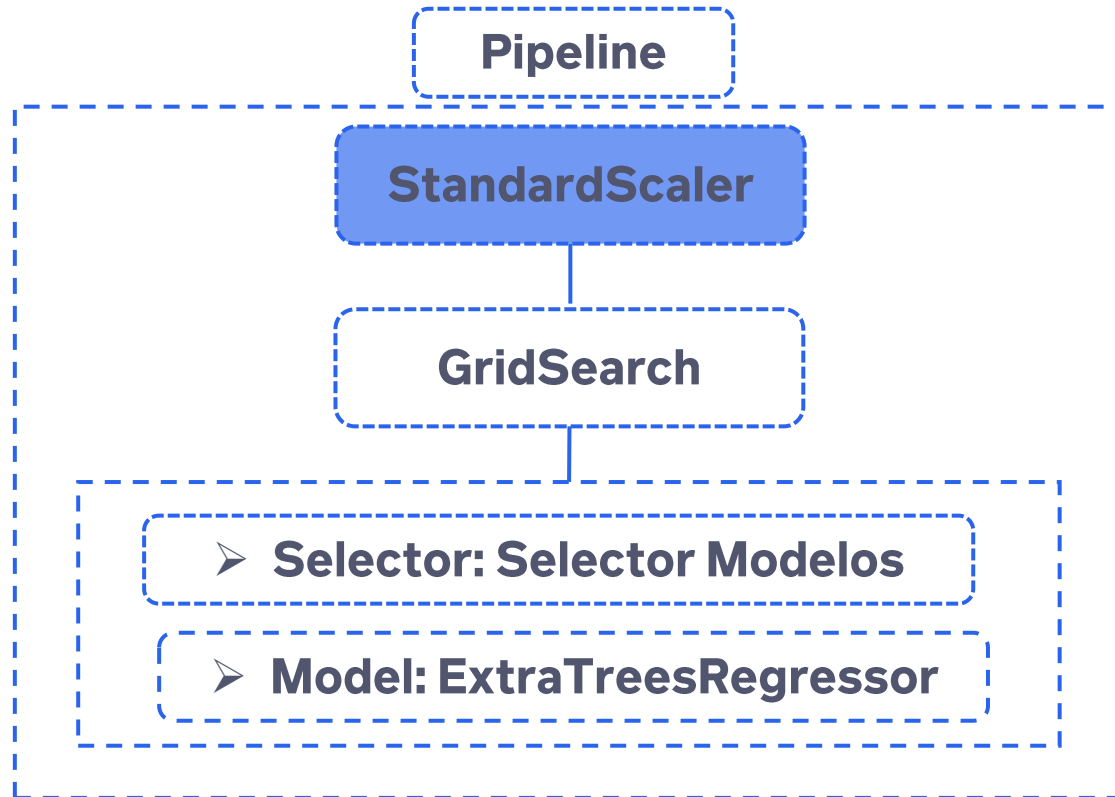
Proyecto: Evaluador de modelos



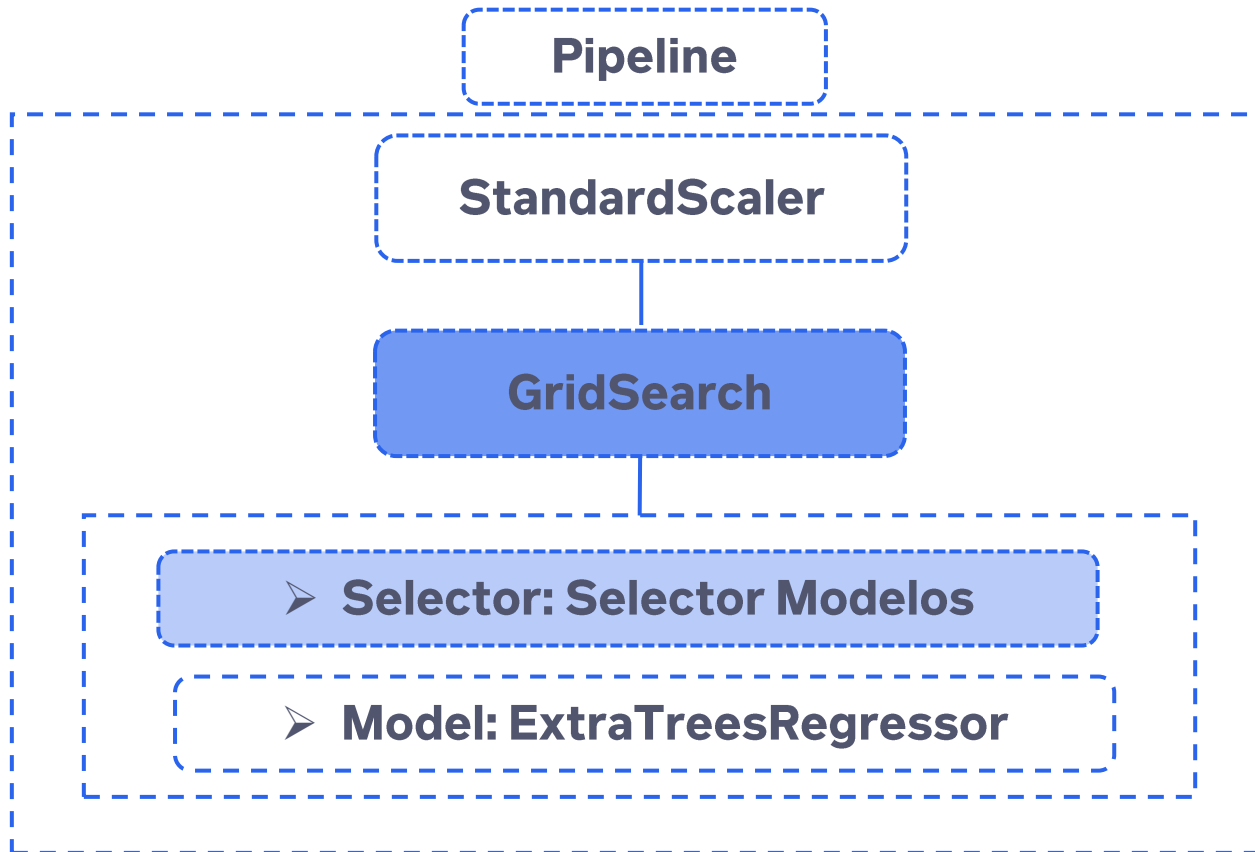
Proyecto: Evaluador de modelos



Proyecto: Evaluador de modelos



Proyecto: Evaluador de modelos



```
class SelectorModelos(BaseEstimator):
    def __init__(self, model):
        """
        A Custom BaseEstimator
        """
        self.model=model
        self.estimator = self.model

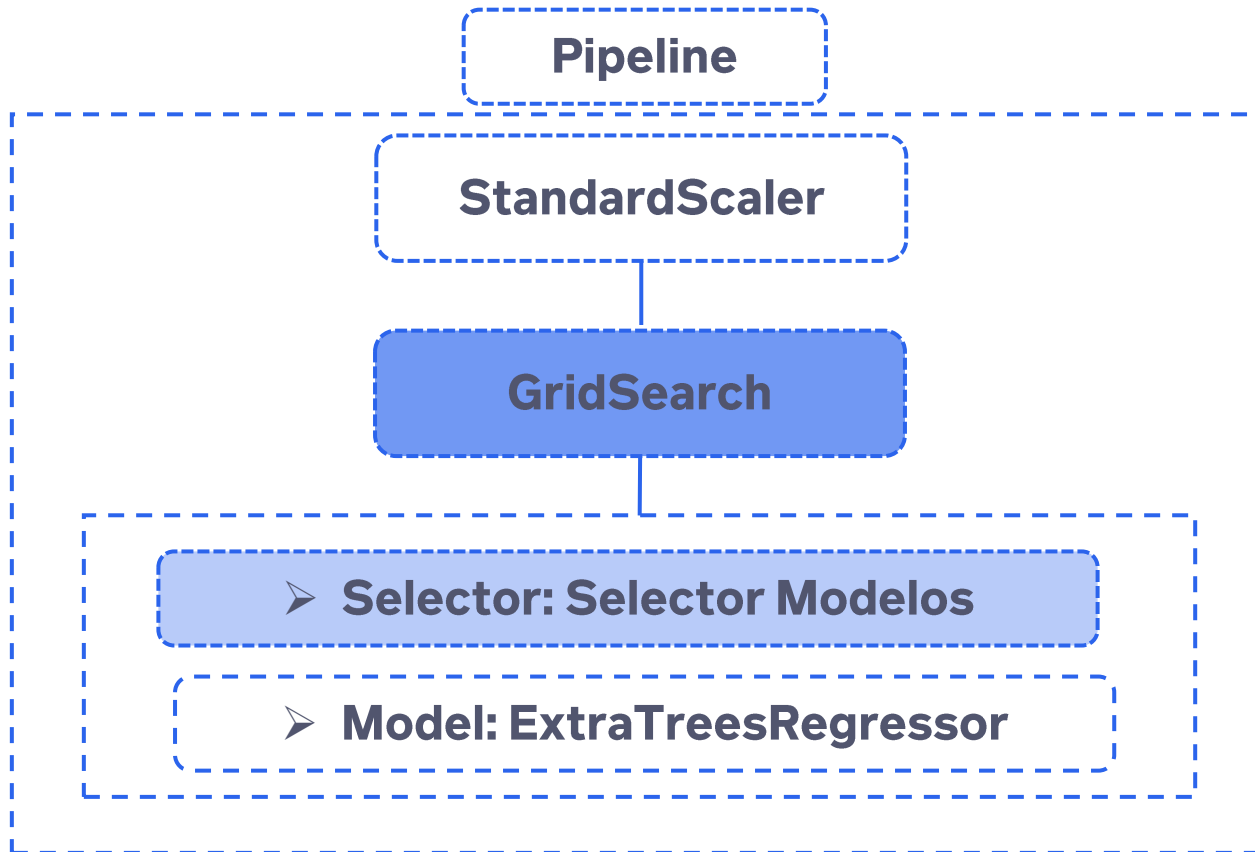
    def fit(self, X, y=None, **kwargs):
        self.estimator.fit(X, y)
        return self

    def predict(self, X, y=None):
        return self.estimator.predict(X)

    def predict_proba(self, X):
        return self.estimator.predict_proba(X)

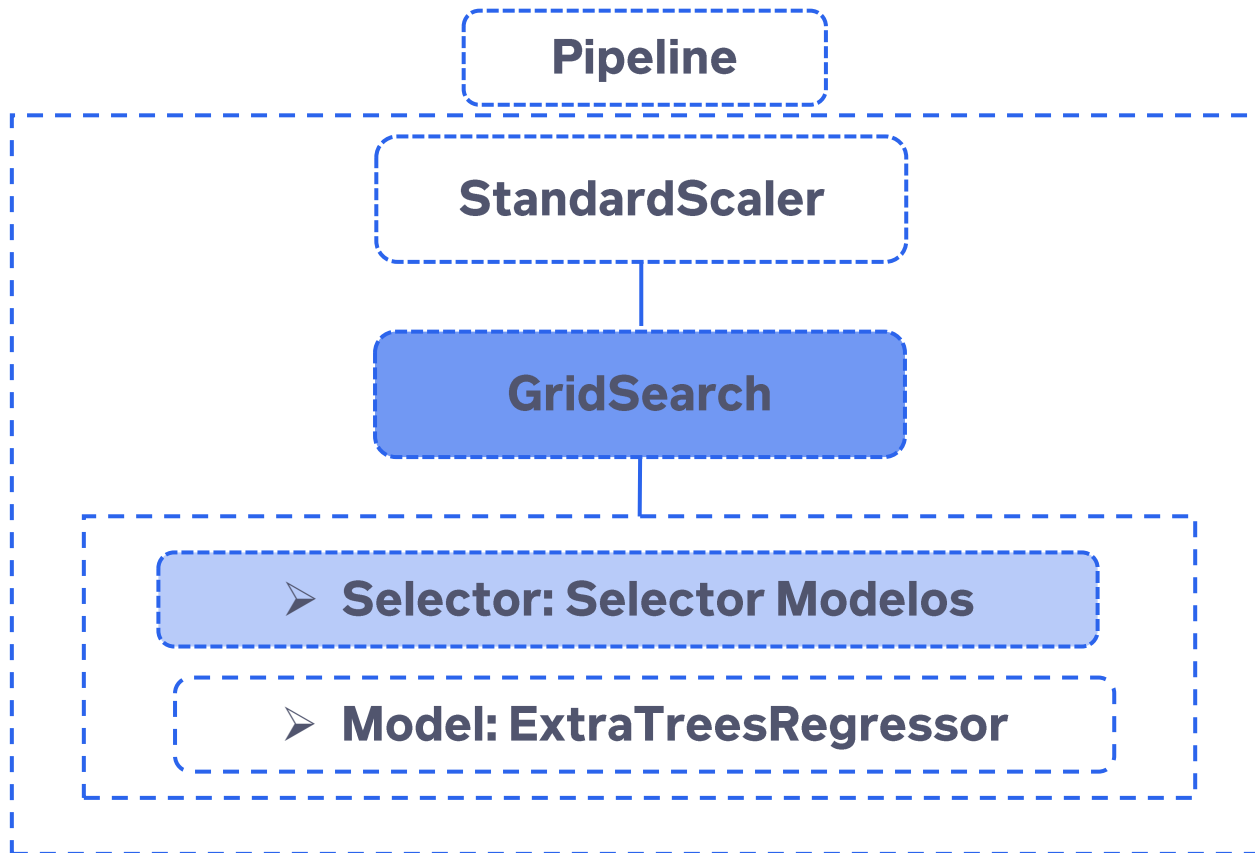
    def score(self, X, y):
        return self.estimator.score(X, y)
```


Proyecto: Evaluador de modelos



```
model_list=[Ridge(),Lasso(),ElasticNet(),
            RandomForestRegressor(),ExtraTreesRegressor(),
            XGBRegressor()]
params=[{'selector__model__alpha':np.logspace(-2,4,100)},
        {'selector__model__alpha':np.logspace(-2,4,100)},
        {'selector__model__alpha':np.logspace(-2,4,100),
         'selector__model__l1_ratio': list(np.linspace(0.01,0.99,25))},
        {'selector__model__n_estimators': np.arange(50,1050,100),
         'selector__model__max_depth': list(range(1,7))},
        {'selector__model__random_state':[16]},
        {'selector__model__n_estimators': np.arange(50,1050,100),
         'selector__model__max_depth': list(range(1,7))},
        {'selector__model__random_state':[16]},
        {'selector__model__n_estimators': np.arange(50,1050,100),
         'selector__model__max_depth': list(range(1,7))}]
]
```

Proyecto: Evaluador de modelos

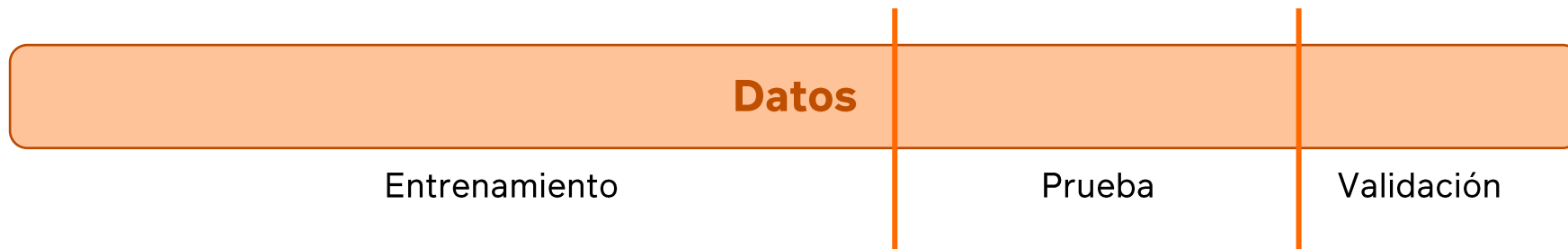


```
res=[]
for k,mdl in enumerate(model_list):
    pipe_tst=Pipeline(steps=[
        ('preprocessor',StandardScaler()),
        ('selector',SelectorModelos(mdl))
    ])
    gs=GridSearchCV(pipe_tst,params[k])
    gs.fit(X_train,y_train)
    res.append([gs.best_estimator_,
                gs.best_params_,
                gs.best_score_,
                gs.score(X_test,y_test)])
```

Resultados

Prueba: Entrenamiento con 6 meses de datos

01/07/2022 31/12/2022 15/01/2023



Modelo

(StandardScaler(), SelectorModelos(model=Ridge(alp...
(StandardScaler(), SelectorModelos(model=Lasso(alp...
(StandardScaler(), SelectorModelos(model=ElasticNe...
(StandardScaler(), SelectorModelos(model=RandomFor...
(StandardScaler(), SelectorModelos(model=ExtraTree...
(StandardScaler(), SelectorModelos(model=XGBRegres...

R² Entrenamiento

0.429034
0.434199
0.429853
-0.535183
-0.131433
-0.366488

R² Prueba

0.502714
0.507067
0.502119
0.447369
0.543862
0.546859

Datos de Validación

Mejor Modelo Lasso

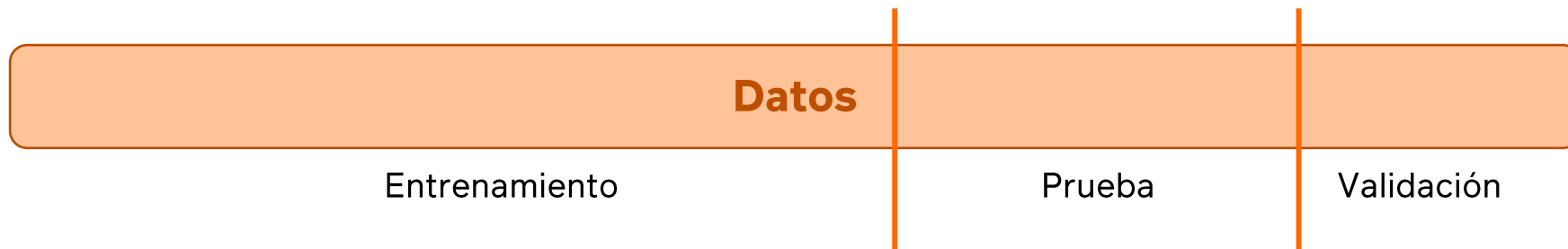
R² 0.27

MAPE 13.72 %

Resultados

Prueba: Entrenamiento con 12 meses de datos

01/01/2022 31/12/2022 15/01/2023



Modelo

(StandardScaler(), SelectorModelos(model=Ridge(alp...
(StandardScaler(), SelectorModelos(model=Lasso(alp...
(StandardScaler(), SelectorModelos(model=ElasticNe...
(StandardScaler(), SelectorModelos(model=RandomFor...
(StandardScaler(), SelectorModelos(model=ExtraTree...
(StandardScaler(), SelectorModelos(model=XGBRegres...

R² Entrenamiento

0.505590
0.502551
0.505596
0.321696
0.365138
0.449884

R² Prueba

0.590292
0.612079
0.583046
0.419766
0.273747
0.306916

Datos de Validación

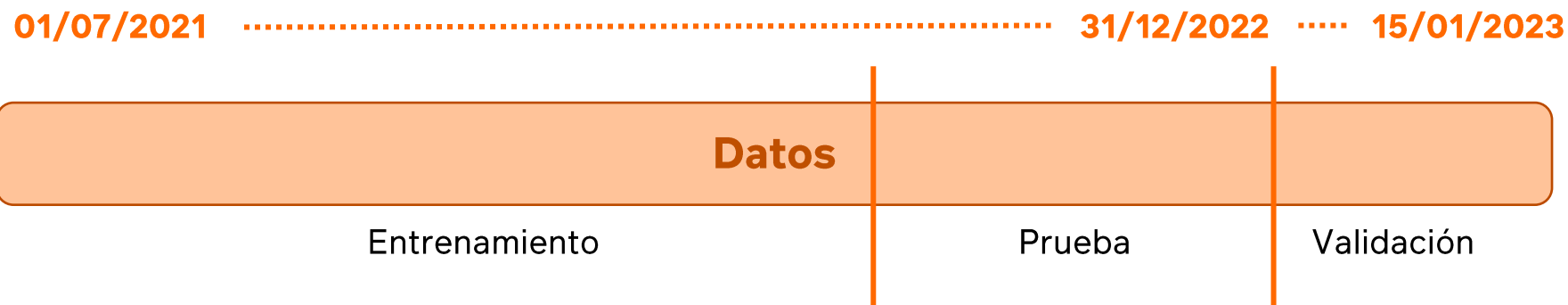
Mejor Modelo Lasso

R² 0.25

MAPE 14.06 %

Resultados

Prueba: Entrenamiento con 18 meses de datos



Modelo

(StandardScaler(), SelectorModelos(model=Ridge(alp...
(StandardScaler(), SelectorModelos(model=Lasso(alp...
(StandardScaler(), SelectorModelos(model=ElasticNe...
(StandardScaler(), SelectorModelos(model=RandomFor...
(StandardScaler(), SelectorModelos(model=ExtraTree...
(StandardScaler(), SelectorModelos(model=XGBRegres...

R² Entrenamiento

0.467684
0.467495
0.468531
0.029043
0.344083
0.099191

R² Prueba

0.884323
0.885452
0.883554
0.789817
0.820310
0.826105

Datos de Validación

Mejor Modelo	Lasso
R²	0.61
MAPE	10.72 %



¡Gracias!