

Más vale prevenir que curar: un remedio en forma de pipeline de ML

BBVA - AI Factory
Data4Risk

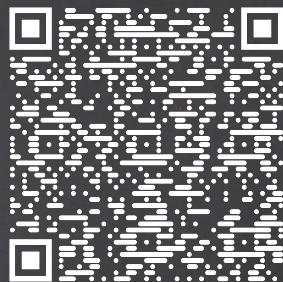
Pablo Quicios Asenjo
Carlos González Álvarez



AI Factory

BBVA

¿Nos conocemos?



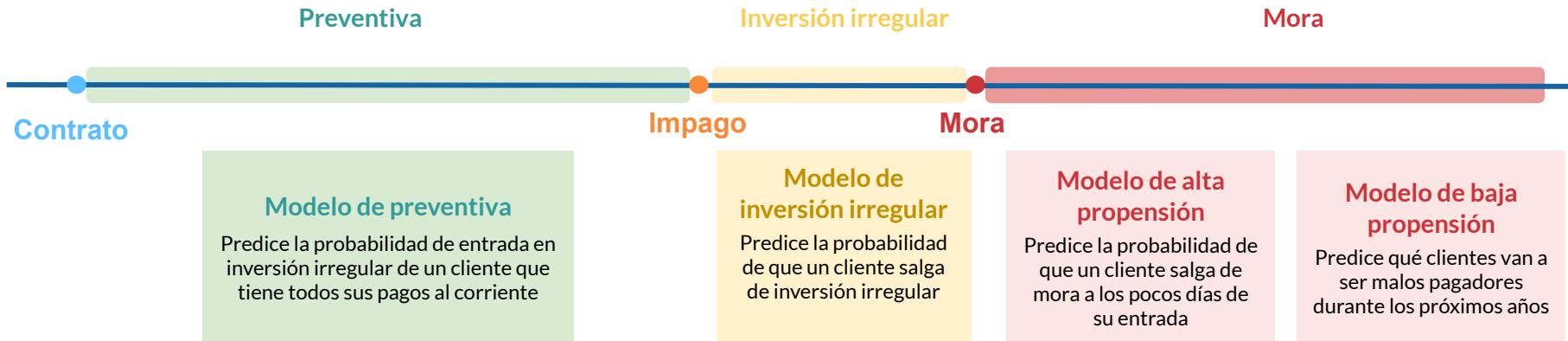
En BBVA AI Factory siempre estamos buscando talento excepcional

bbvaaifactory.com

Síguenos en Twitter y LinkedIn



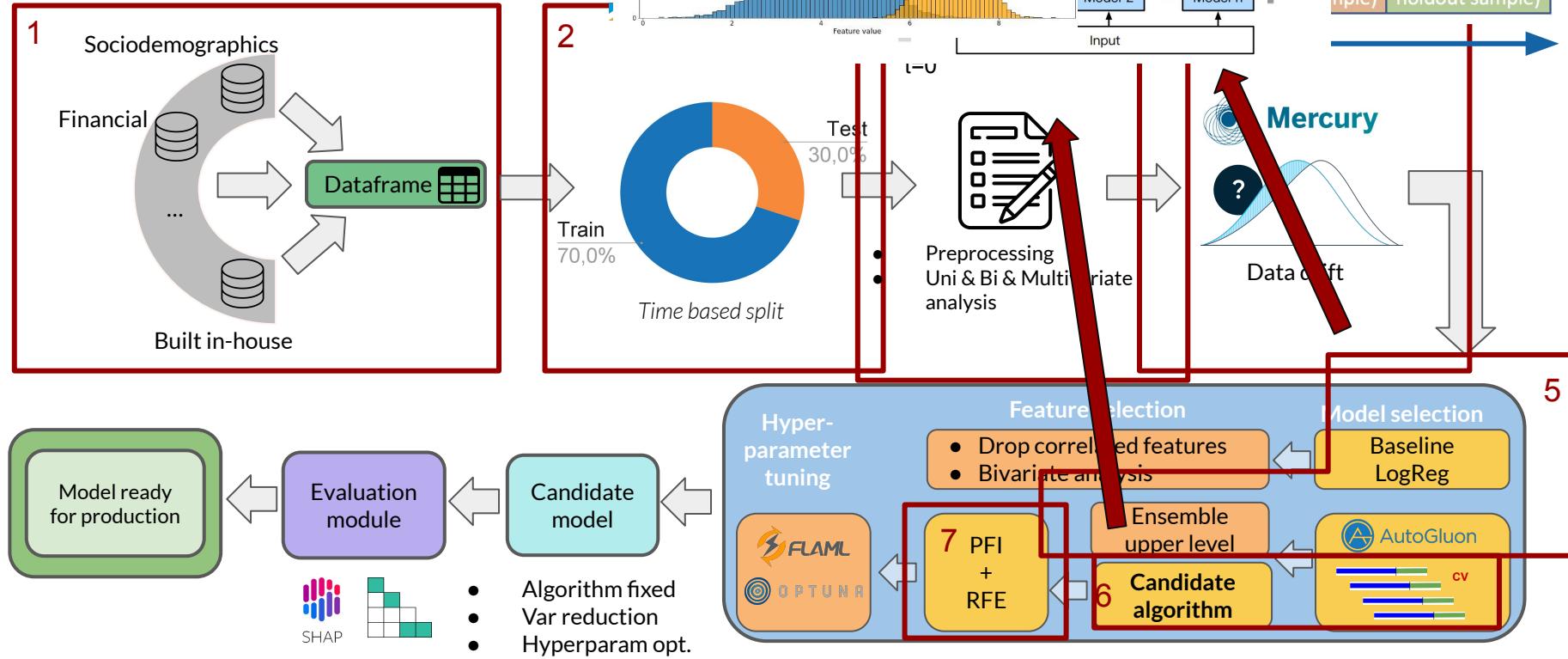
Construimos modelos predictivos para la gestión temprana de la deuda buscando mejorar la situación de los clientes y del banco



Similitudes de los diferentes proyectos:

1. Tabular ML
2. Target binario
3. Gran volumen inicial de variables
4. Deadlines apretados
5. Entorno productivo específico
6. Tramificación del score

Modelling pipeline





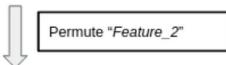
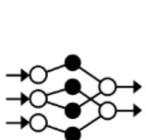
Feature Selection

PFI + RFE

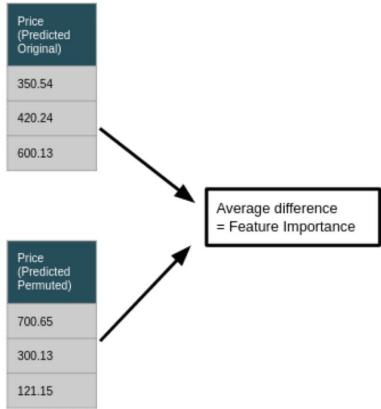
Se parte de un data set con alta dimensionalidad y es necesario seleccionar un subconjunto de variables que mantenga o mejore el poder predictivo respetando las limitaciones de memoria y tiempo .

Permutation Feature Importance (PFI)

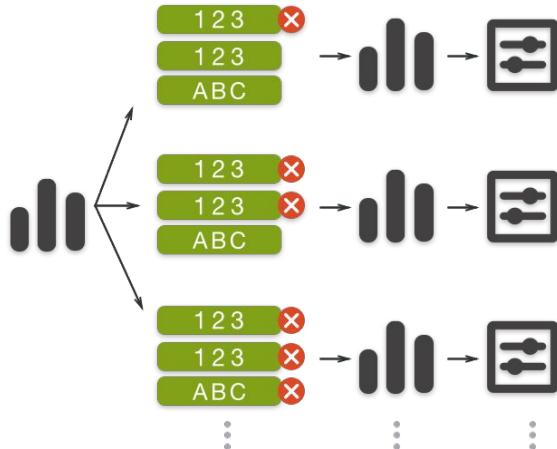
	Feature_1	Feature_2	Feature_3	Label
Sample_1	1970	10.5	1	403.12
Sample_2	2020	14.9	2	412.15
Sample_3	1910	17.7	3	564.46



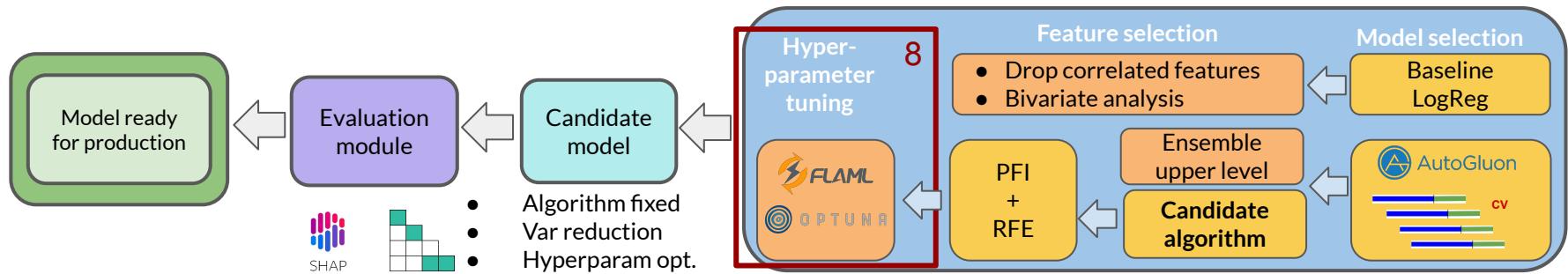
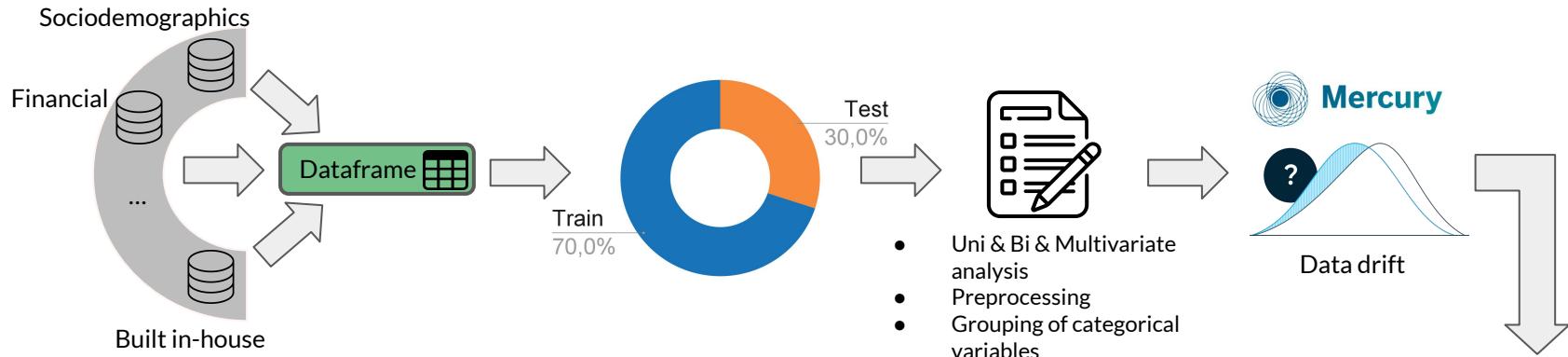
	Feature_1	Feature_2	Feature_3	Label
Sample_1	1970	17.7	1	403.12
Sample_2	2020	10.5	2	412.15
Sample_3	1910	14.9	3	564.46



Recursive Feature Elimination (RFE)



Modelling pipeline

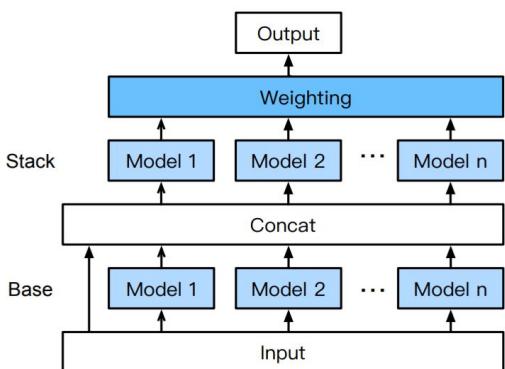


Hyperparameter optimization

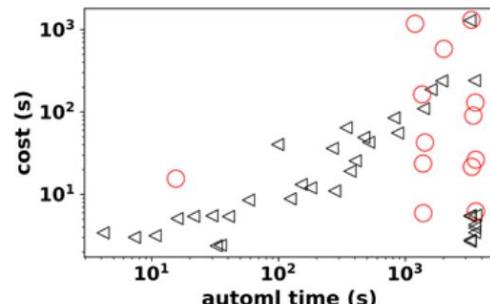


AutoGluon

- Transfer learning at starting point
- Different search strategies (RL, Bayesian, Gaussian process surrogate model...)



- Time budget
- Incremental complexity for each iteration



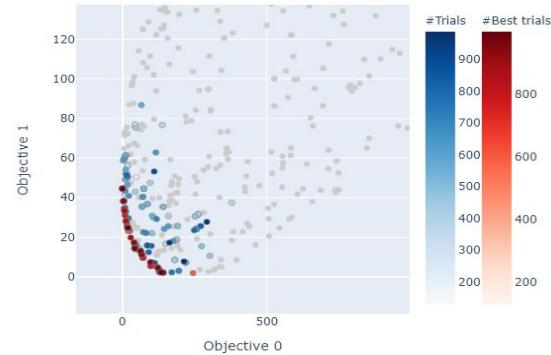
Trial cost vs. total time spent in automl. Each marker corresponds to one trial of configuration evaluation. Triangles mark FLAML; circles mark a typical existing AutoML library.



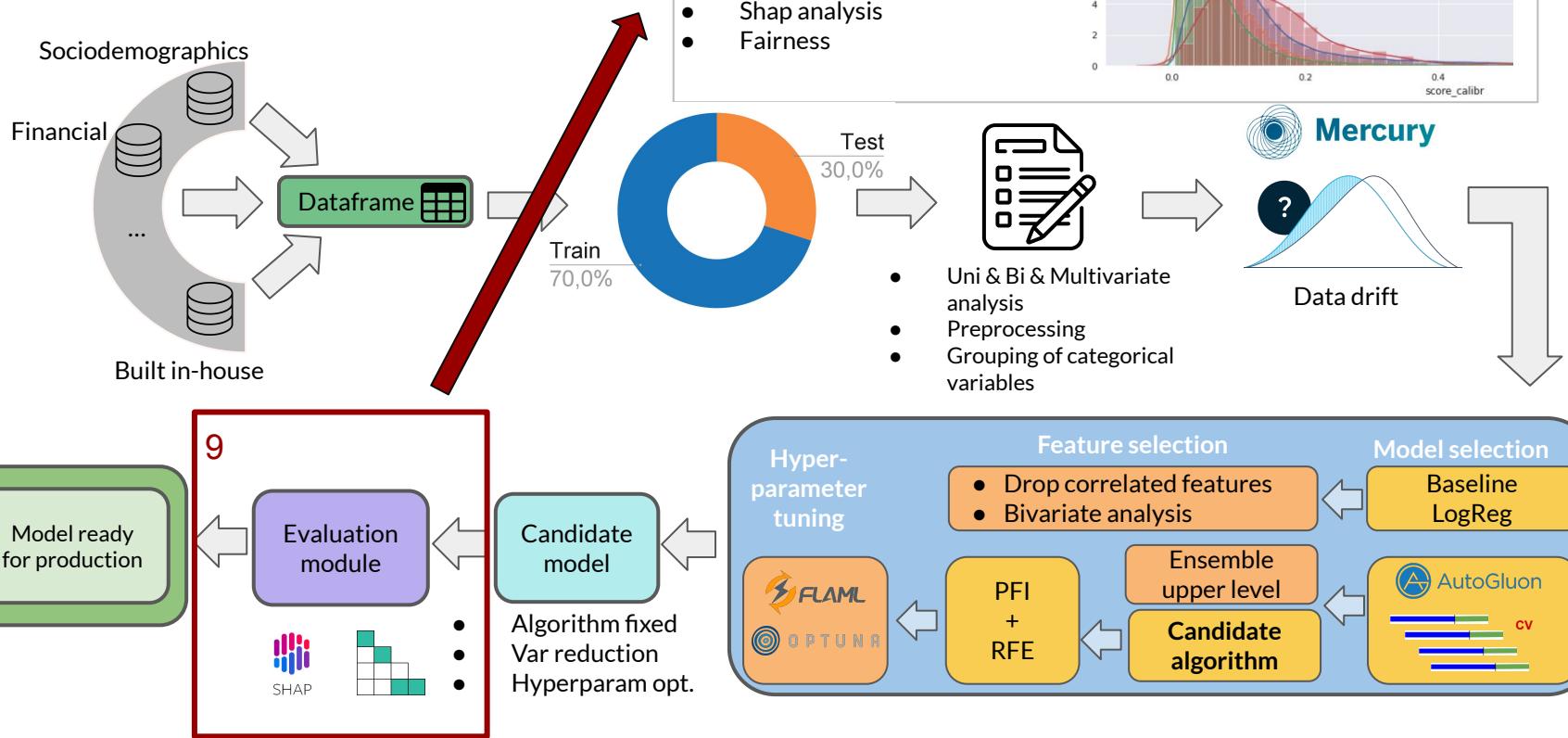
OPTUNA

- Customizable objective function → Multiobjective
- Time-series cross-validation
- Different search strategies (RL, Bayesian, TPSE, Genetic...)

Pareto-front Plot

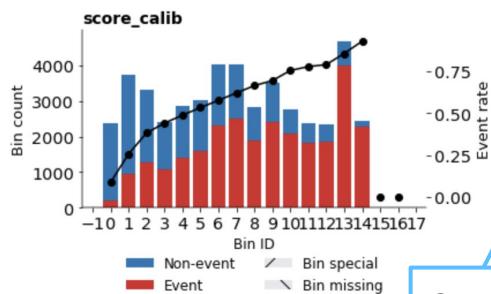


Modelling pipeline



Model Evaluation

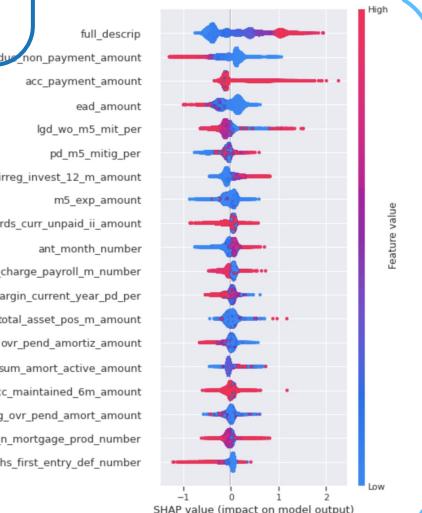
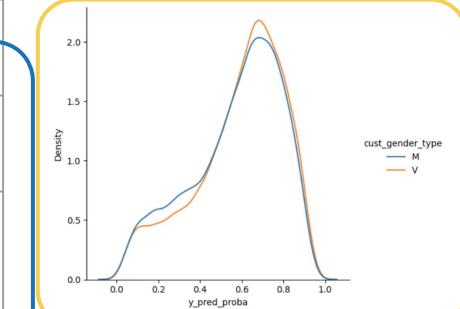
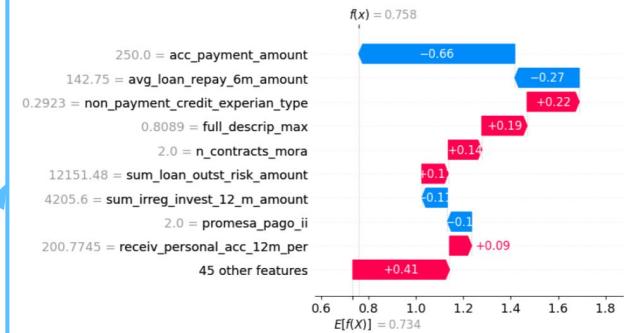
	Bin	Count	Count (%)	Non-event	Event	Event rate
0	[-inf, 14.79]	2378	0.051022	2171	207	0.087048
1	[14.79, 29.42]	3716	0.079731	2763	953	0.256459
2	[29.42, 39.62)	3304	0.070891	2042	1262	0.381961
3	[39.62, 45.53)	2417	0.051859	1352	1065	0.440629
4	[45.53, 51.07)	2871	0.061600	1470	1401	0.487983
5	[51.07, 55.66)	3025	0.064904	1413	1612	0.532893
6	[55.66, 60.81)	4010	0.086039	1699	2311	0.576309
7	[60.81, 65.19)	4009	0.086017	1521	2488	0.620604
8	[65.19, 68.02)	2827	0.060656	950	1877	0.663955
9	[68.02, 71.54)	3488	0.074839	1067	2421	0.694094
10	[71.54, 74.47)	2760	0.059219	678	2082	0.754348
11	[74.47, 77.05)	2365	0.050743	526	1839	0.777590
12	[77.05, 79.85)	2333	0.050057	491	1842	0.789541
13	[79.85, 86.86)	4652	0.099813	669	3983	0.856191
14	[86.86, inf)	2452	0.052610	173	2279	0.929445



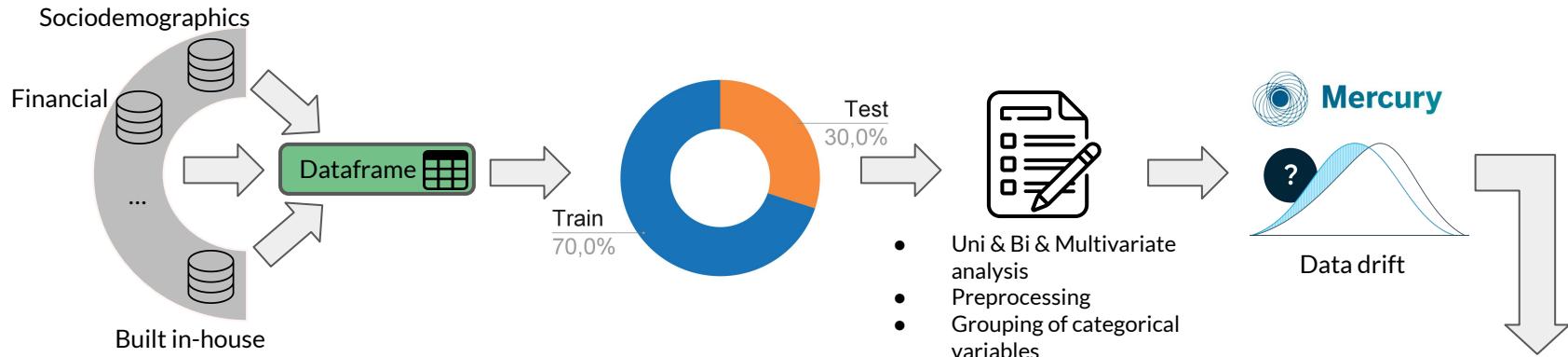
SHAP

Métricas por segmentos
Optimal binning

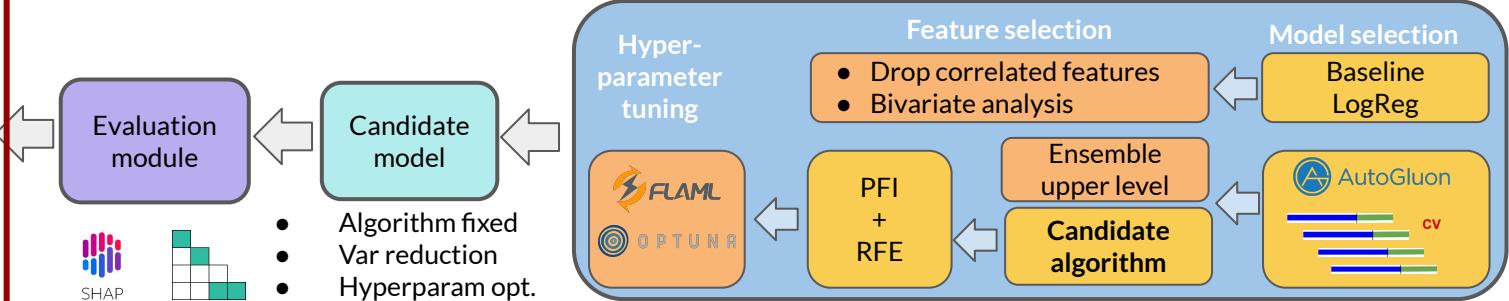
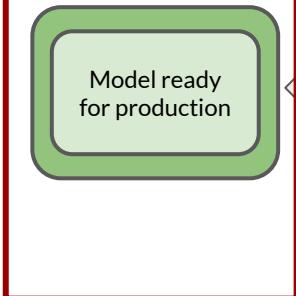
Hipoteca en tenencia	% BP test	Acc Test	AUC Test	GINI Test
Hipoteca en mora	40.0%	0.62	0.68	0.36
NO hipoteca en mora	36.3%	0.72	0.76	0.53



Modelling pipeline



10



Takeaways

Model selection

AutoGluon:

<https://auto.gluon.ai/stable/index.html>



AutoGluon

Feature selection

PFI + RFE (AutoGluon):

https://github.com/autogluon/autogluon/blob/master/core/src/autogluon/core/utils/feature_selection.py



AutoGluon

Hyperparameter optimization

FLAML: <https://microsoft.github.io/FLAML/>



Optuna: <https://optuna.org/>

AutoGluon: <https://auto.gluon.ai/stable/index.html>



Evaluation

SHAP: <https://shap.readthedocs.io/en/latest/>



OptBinning: <http://gnpalencia.org/optbinning/>

Monitoring

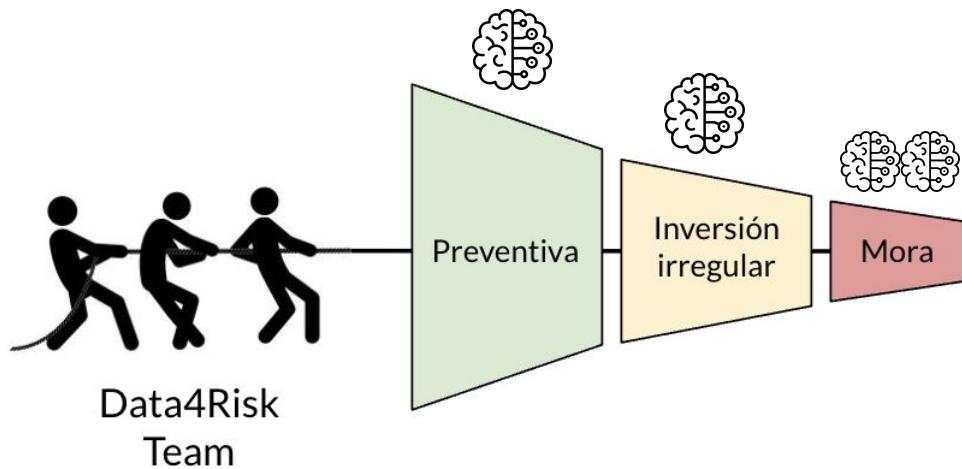
Mercury BBVA:

https://www.bbvaifactory.com/mercury/#structure_section



Mercury

Takeaways



The whole team!

LEADS



DATA SCIENTISTS



DATA ENGINEERS





AI Factory

¡Gracias!

En BBVA AI Factory siempre estamos buscando talento excepcional

Conoce nuestras ofertas



bbvaifactory.com

Síguenos en Twitter y LinkedIn





Creating Opportunities

Anexo



You

¿Por qué en un modelo de aprendizaje supervisado no se suele optimizar la elección del algoritmo, la selección de variables y la elección de los hiperparámetros en un mismo proceso común?



ChatGPT

En el aprendizaje supervisado, la optimización de la elección del algoritmo, la selección de variables y la elección de hiperparámetros son tareas distintas pero interrelacionadas. Aunque en algunos casos se pueden abordar de manera conjunta, por lo general se realizan en procesos separados por varias razones:

1. **Complejidad del espacio de búsqueda:** La combinación de algoritmos, variables y hiperparámetros genera un espacio de búsqueda extremadamente grande y complejo. Explorar todas las posibles combinaciones de manera exhaustiva puede ser computacionalmente costoso y requerir mucho tiempo.



AutoGluon

Factores diferenciales:

- Cota superior de performance
- Metodología altamente ‘customizable’

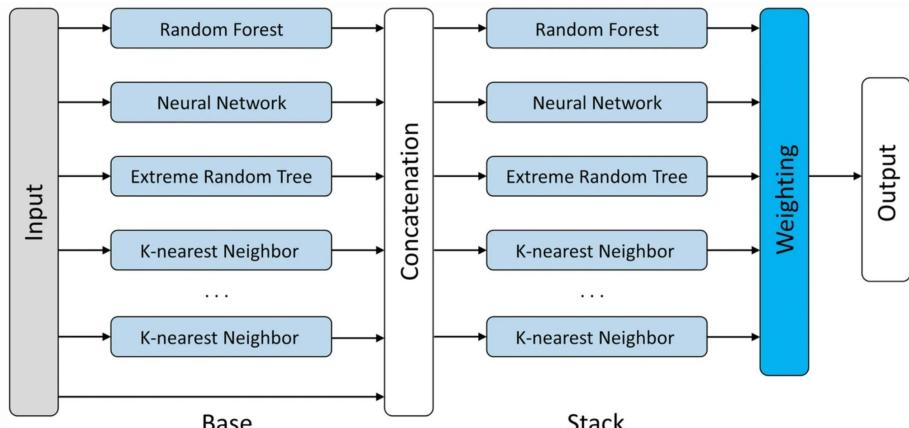
Limitaciones:

- No es compatible con time series split CV
- Optimización de hiperparámetros vía Random Search
- Preprocesado automático o ‘customizable’ sin opción al tratamiento propio de los modelos
- Output de modelo no compatible con el módulo de evaluación

Algorithm 1 AutoGluon-Tabular Training Strategy
(multi-layer stack ensembling + n -repeated k -fold bagging).

Require: data (X, Y) , family of models \mathcal{M} , # of layers L

```
1: Preprocess data to extract features
2: for  $l = 1$  to  $L$  do {Stacking}
3:   for  $i = 1$  to  $n$  do { $n$ -repeated}
4:     Randomly split data into  $k$  chunks  $\{X^j, Y^j\}_{j=1}^k$ 
5:     for  $j = 1$  to  $k$  do { $k$ -fold bagging}
6:       for each model type  $m$  in  $\mathcal{M}$  do
7:         Train a type- $m$  model on  $X^{-j}, Y^{-j}$ 
8:         Make predictions  $\hat{Y}_{m,i}^j$  on OOF data  $X^j$ 
9:       end for
10:      end for
11:    end for
12:    Average OOF predictions  $\hat{Y}_m = \{\frac{1}{n} \sum_i \hat{Y}_{m,i}^j\}_{j=1}^k$ 
13:     $X \leftarrow$  concatenate( $X, \{\hat{Y}_m\}_{m \in \mathcal{M}}$ )
14:  end for
```



A two-stacking layers example of the multi-layer stacking framework of AutoGluon

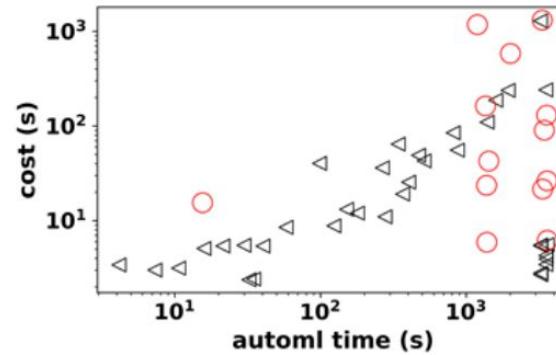


Factores diferenciales:

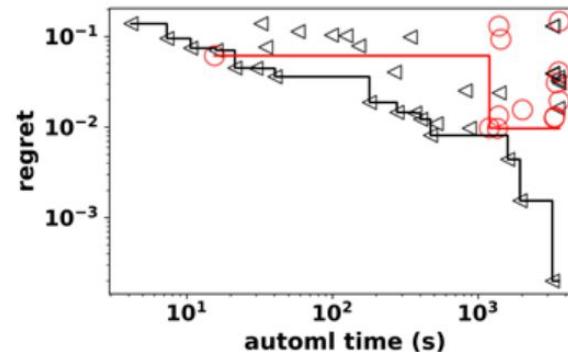
- Cálculo de candidatos en función del time budget
- Búsqueda de hiperparámetros incremental
- Puede partir de una combinación de hiperparámetros inicial asignada por el usuario

Limitaciones:

- No es compatible con time series split CV
- No permite que LightGBM use su propio tratamiento de categóricas
- Solo permite optimizar algunos hiperparámetros de cada modelo



Trial cost vs. total time spent in automl. Each marker corresponds to one trial of configuration evaluation. Triangles mark FLAML; circles mark a typical existing AutoML library.



Model auc regret vs. total time spent in automl. Each marker corresponds to one trial of configuration evaluation. Triangles mark FLAML; circles mark a typical existing AutoML library.



OPTUNA

Factores diferenciales:

- 100% customizable
- Posibilidad de optimización multiobjetivo
- Distintas metodologías de búsqueda de hiperparámetros
- Compatible con time series split CV

Limitaciones:

- No tiene AutoML
- No se ajusta a ningún time budget

```
def objective(trial):
    data, target = sklearn.datasets.load_breast_cancer(return_X_y=True)
    train_x, valid_x, train_y, valid_y = train_test_split(data, target, test_size=0.25)
    dtrain = lgb.Dataset(train_x, label=train_y)

    param = {
        "objective": "binary",
        "metric": "binary_logloss",
        "verbosity": -1,
        "boosting_type": "gbdt",
        "lambda_l1": trial.suggest_float("lambda_l1", 1e-8, 10.0, log=True),
        "lambda_l2": trial.suggest_float("lambda_l2", 1e-8, 10.0, log=True),
        "num_leaves": trial.suggest_int("num_leaves", 2, 256),
        "feature_fraction": trial.suggest_float("feature_fraction", 0.4, 1.0),
        "bagging_fraction": trial.suggest_float("bagging_fraction", 0.4, 1.0),
        "bagging_freq": trial.suggest_int("bagging_freq", 1, 7),
        "min_child_samples": trial.suggest_int("min_child_samples", 5, 100),
    }

    gbm = lgb.train(param, dtrain)
    preds = gbm.predict(valid_x)
    pred_labels = np.rint(preds)
    accuracy = sklearn.metrics.accuracy_score(valid_y, pred_labels)
    return accuracy

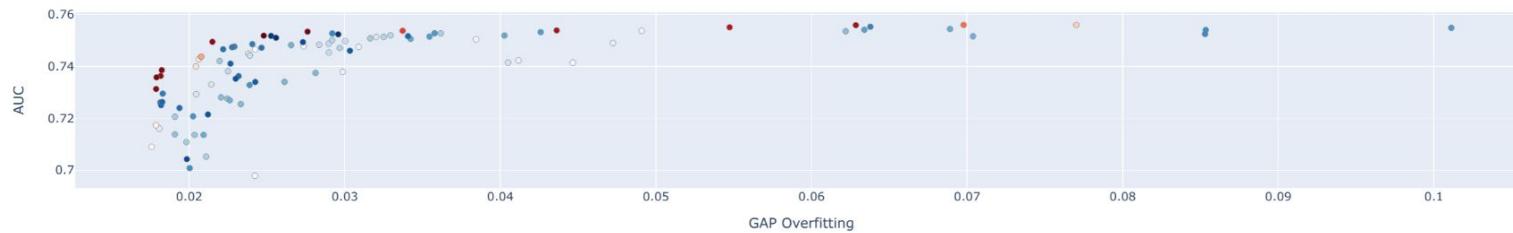
if __name__ == "__main__":
    study = optuna.create_study(direction="maximize")
    study.optimize(objective, n_trials=100)
```



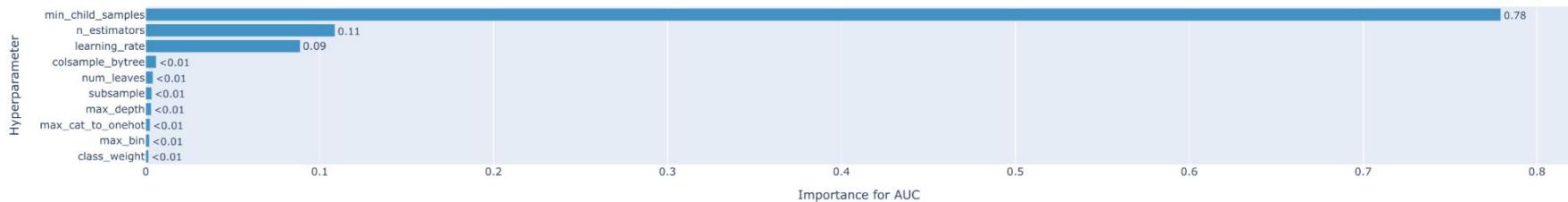
OPTUNA - Multiobjective

- Maximizar AUC en folds de validación
- Minimizar Gap de AUC entre folds de train y validación

Pareto-front Plot



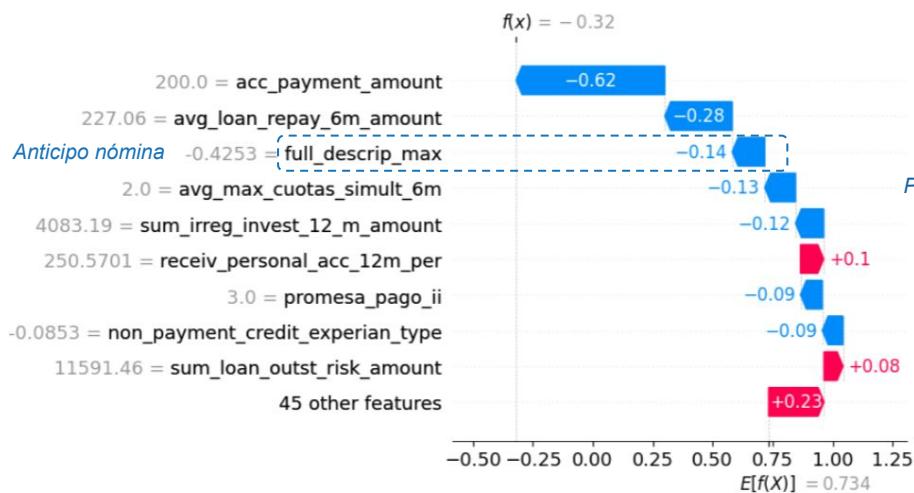
Hyperparameter Importances



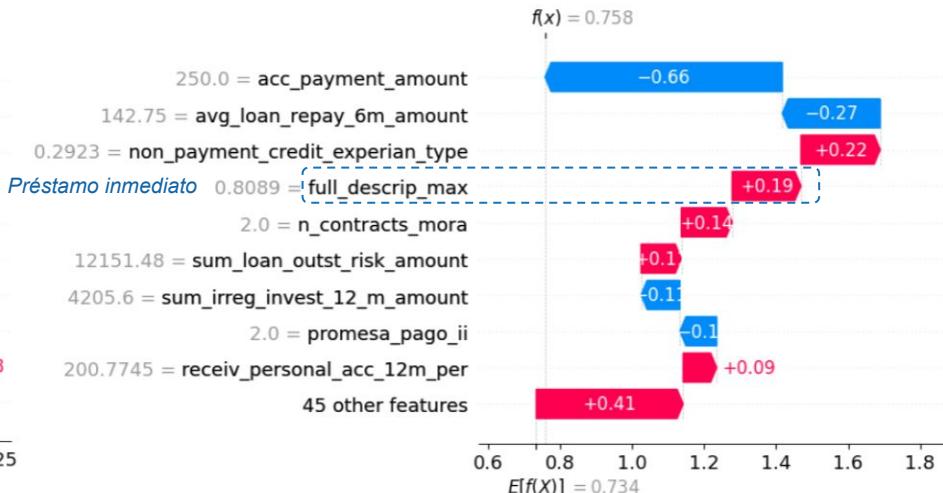
SHAP: Casos particulares de baja propensión

	customer_id	part_closing_date	n_contracts_mora	n_contracts_total	cliente_CF	casos_tenencia	cliente_CF	default_main_risk_amount	y_pred_proba	target_principal_perc_recovery_2y
0	013210441	2020-07-31	1	13	0	2.0	0	687.61	0.422216	0
1	013210441	2020-08-31	2	13	0	3.0	0	12151.48	0.685781	1

2020-07-31



2020-08-31



$$f(x) = \ln\left(\frac{\text{predict}_{\text{proba}}}{1-\text{predict}_{\text{proba}}}\right)$$