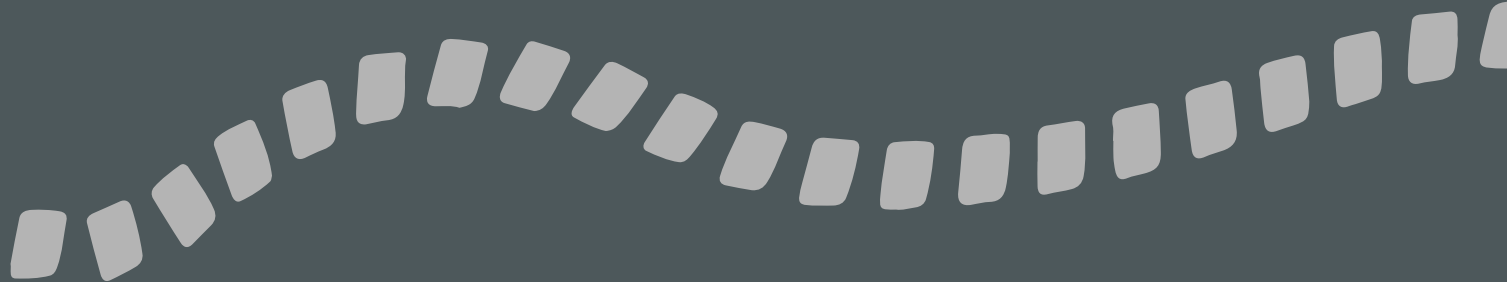




Propuesta de **Modelo de Detección de Fraude y Scorecard**

Sergio Maldonado Rodriguez



Objetivo: Desarrollar un modelo predictivo para estimar probabilidad de fraude

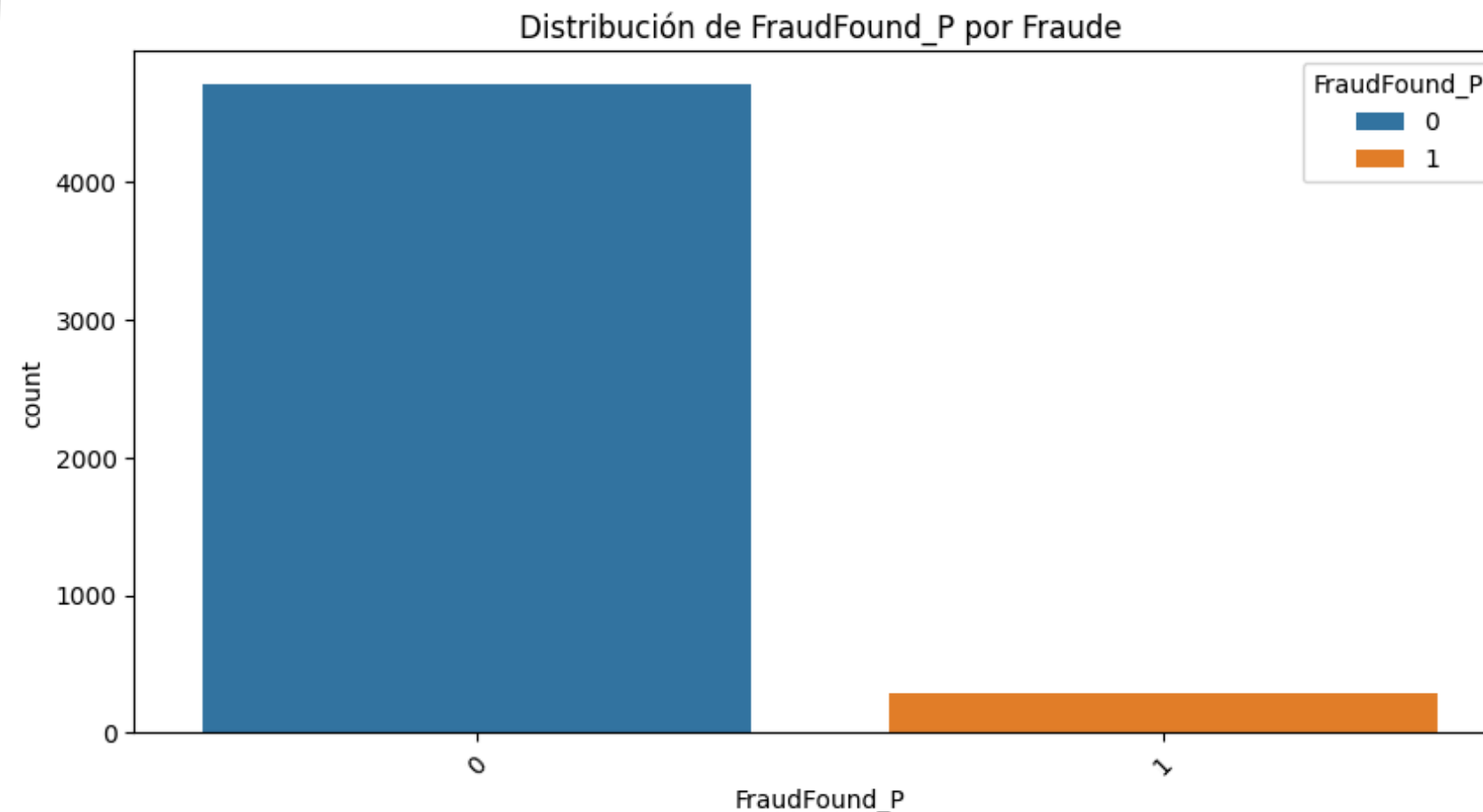
Alcance: Preparación de datos, entrenamiento, scorecard y API local.

Beneficio: Reducción de pérdidas y priorización de casos.



Datos y Preparación

- Fuente de datos: Dataset histórico de transacciones. Con un fuerte desbalanceo



Técnicas ocupadas:

- Limpieza de datos principalmente se encontraron algunos valores nulos
- Al ser datos categoricos se aplico WOE/IV para la seleccion de variables y transformacion de datos

Puntos a resaltar del EDA

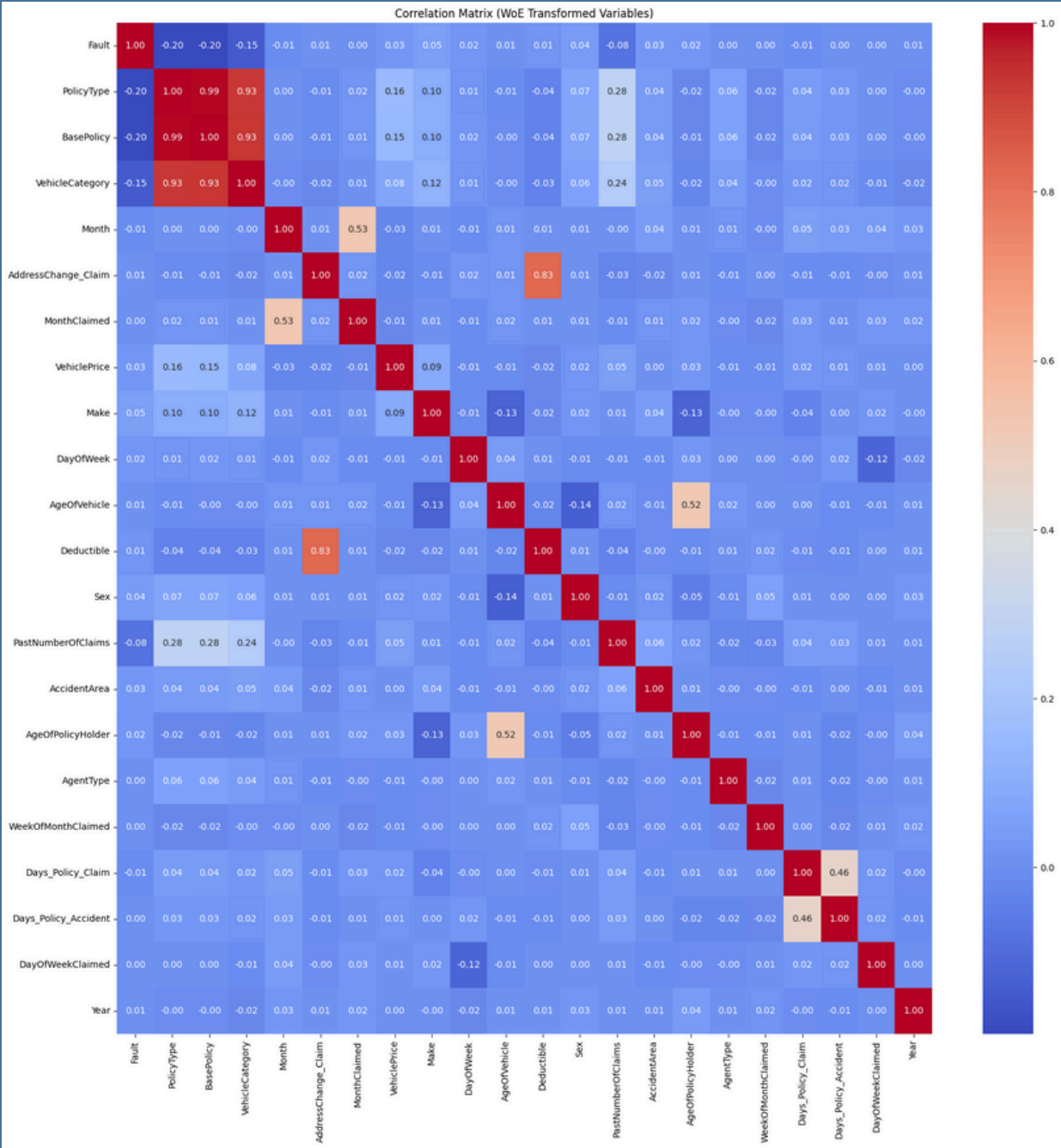
- Casi todas las pólizas tiene mas de 30 días de política de días de accidente
- La mayoría de propietarios de las pólizas están en un rango de edad entre los 31 a 40
- El deducible de casi todas las pólizas es de 400
- El tipo de póliza mas común es para vehículos tipo sedan
- El cambio de dirección no es común en las pólizas
- El numero de quejas se comporta bien variado por lo cual puede ser una buena variable a ocupar ya que no tiene tanta variabilidad
- los vehículos viejos son mas comunes en las pólizas
- Existen mas pólizas de hombres que de mujeres
- las personas casadas tienen mas pólizas
- los vehículos con valor de 20,000 a 29,000 son los mas comunes en estas pólizas
- el tipo de póliza esta bien distribuido de igual forma lo cual puede ser también una buena variable predicadora
- Existen modelos como honda y pontiac que son mas representativos en las pólizas ya que existen mas casos
- la mayoría de reclamos no tienen testigos ni reporte policial de campo
- La mayoría de pólizas involucran solo 1 carro
- la gran mayoría de siniestros son culpa del propietario de la póliza y son en áreas urbanas.

Elección de características

=== RESUMEN DE INFORMATION VALUES ===

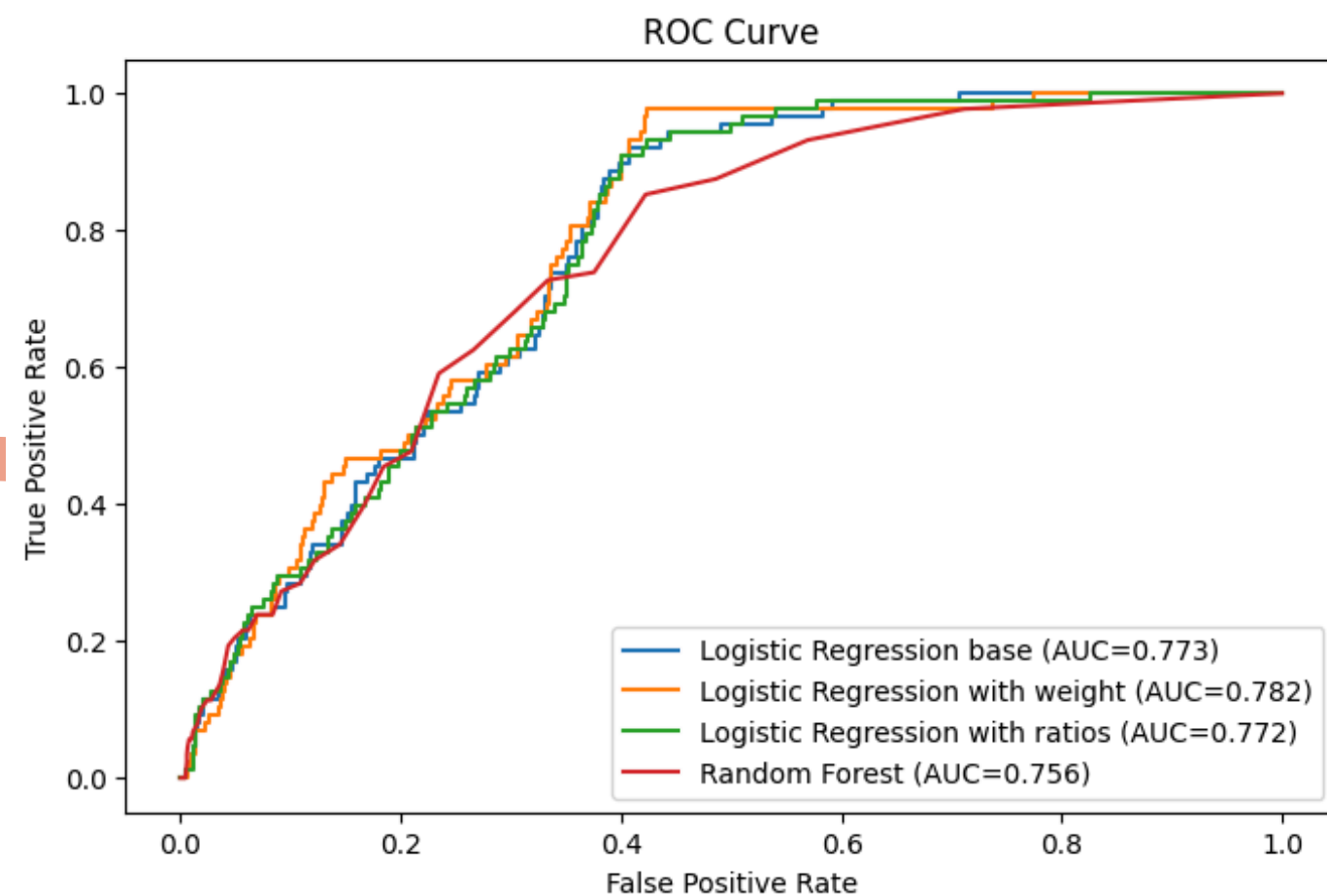
Variable	IV
Fault	0.727217
PolicyType	0.589421
BasePolicy	0.586460
VehicleCategory	0.454021
Month	0.087971
AddressChange_Claim	0.083941
MonthClaimed	0.079986
VehiclePrice	0.077972
Make	0.075414
DayOfWeek	0.071244
AgeOfVehicle	0.065891
Deductible	0.065244
Sex	0.059022
PastNumberOfClaims	0.045636
AccidentArea	0.042930
AgeOfPolicyHolder	0.041880
AgentType	0.039841
WeekOfMonthClaimed	0.031794
Days_Policy_Claim	0.029300
Days_Policy_Accident	0.025491
DayOfWeekClaimed	0.024184
Year	0.021723
PoliceReportFiled	0.014036
MaritalStatus	0.012648
WitnessPresent	0.009457
WeekOfMonth	0.004999
NumberOfCars	-0.000793

Basada en IV de las variables y la correlacion de esta

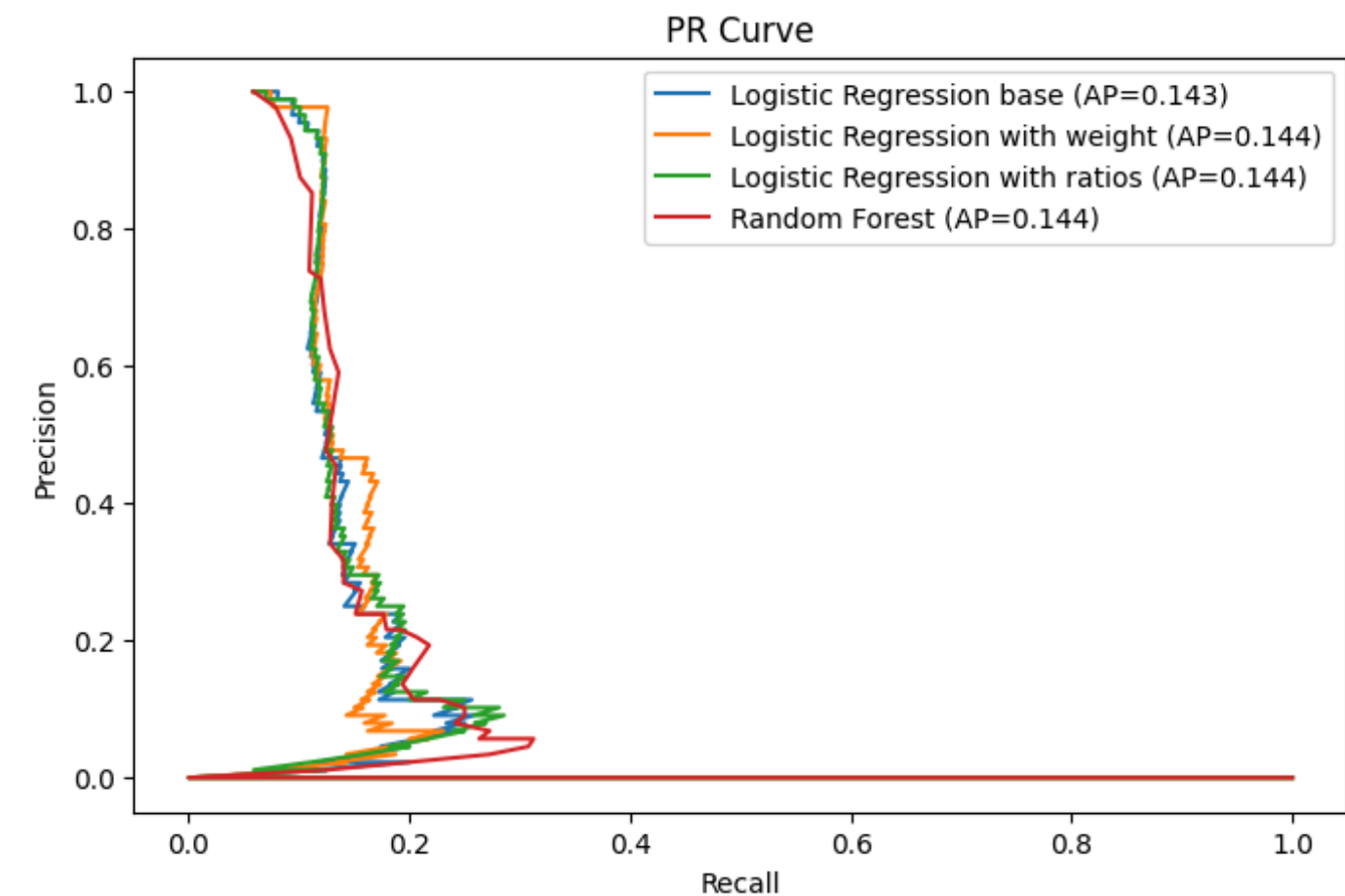


Modelo base

```
models = {  
    "Logistic Regression base": LogisticRegression(max_iter=1000,random_state=42),  
    "Logistic Regression with weight": LogisticRegression(max_iter=1000,random_state=42,class_weight='balanced'),  
    "Logistic Regression with ratios": LogisticRegression(max_iter=1000,random_state=42,class_weight={0: 1, 1: 15}),  
    "Random Forest":RandomForestClassifier(random_state=42,class_weight='balanced',n_estimators=100),  
}
```



Modelos que se compararon para buscar el base, el cual pueda lidiar de mejor manera con el desbalance. De igual forma se determino que la curva PR es la mejor para decidir cual ocupar.



Ajuste de hiperparametros del modelo base

models

```
"Logistic Regression base": LogisticRegression(max_iter=1000,random_state=42,class_weight='balanced'),
"Logistic Regression 1": LogisticRegression(max_iter=1000,random_state=42,class_weight='balanced',penalty= 'l2',solver='liblinear',C=0.01),
"Logistic Regression 2": LogisticRegression(max_iter=1000,random_state=42,class_weight='balanced',penalty= 'l2',solver='liblinear',C=0.1),
"Logistic Regression 3": LogisticRegression(max_iter=1000,random_state=42,class_weight='balanced',penalty= 'l2',solver='liblinear',C=0.10),
"Logistic Regression 4": LogisticRegression(max_iter=1000,random_state=42,class_weight='balanced',penalty= 'l1',solver='liblinear',C=0.1),
"Logistic Regression 5": LogisticRegression(max_iter=1000,random_state=42,class_weight='balanced',penalty= 'l1',solver='liblinear',C=1),
"Logistic Regression 6": LogisticRegression(max_iter=1000,random_state=42,class_weight='balanced',penalty= 'l1',solver='liblinear',C=10),
```

- Con el modelo base se modificaron algunos hiperprametros
- Se hizo un split de los datos en entrenamiento, prueba y evaluacion para poder validar los hiperparametros
- Se centro la atencion en la metrica recall y en el f1-score por ser mas acordes al caso

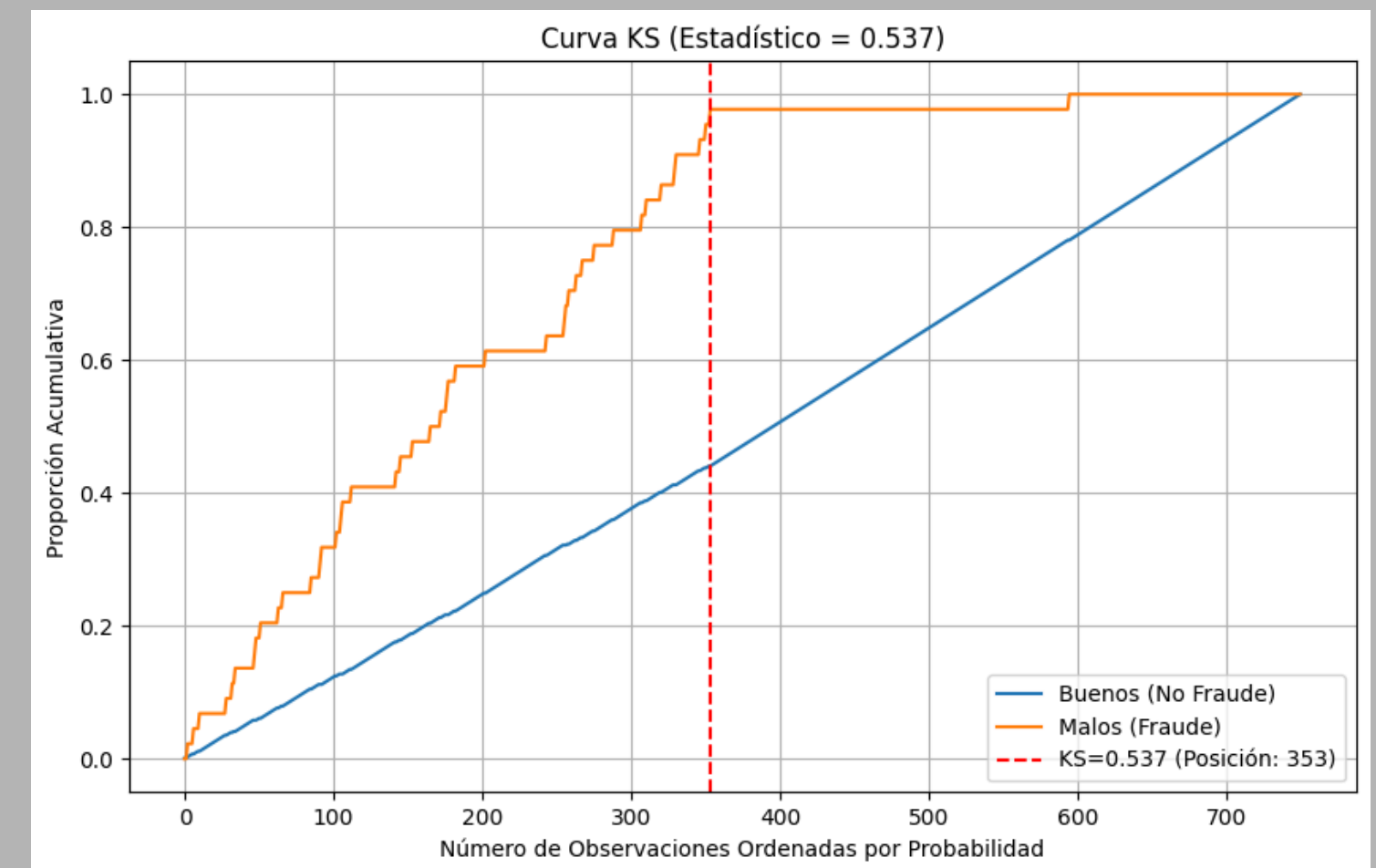
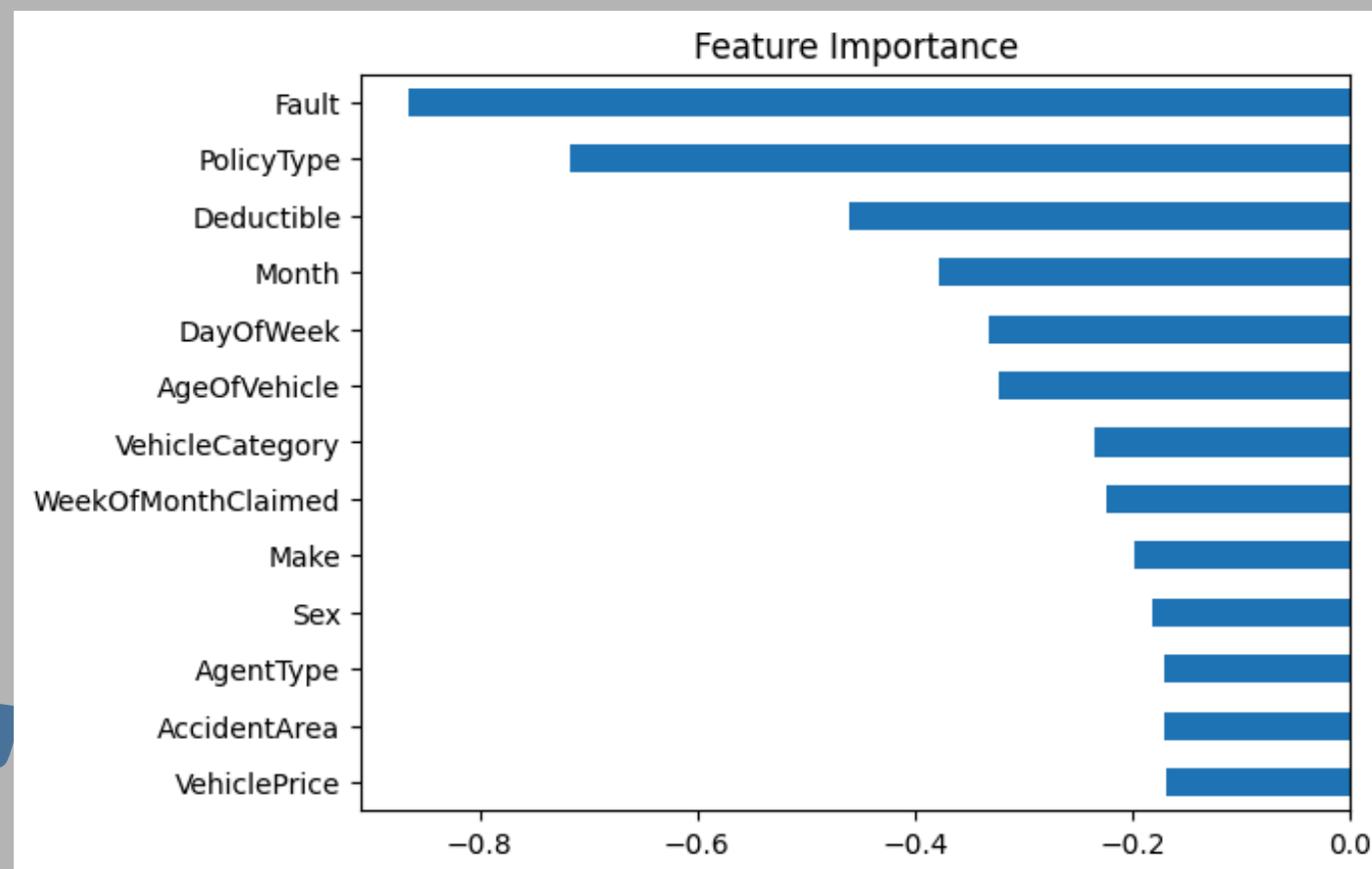
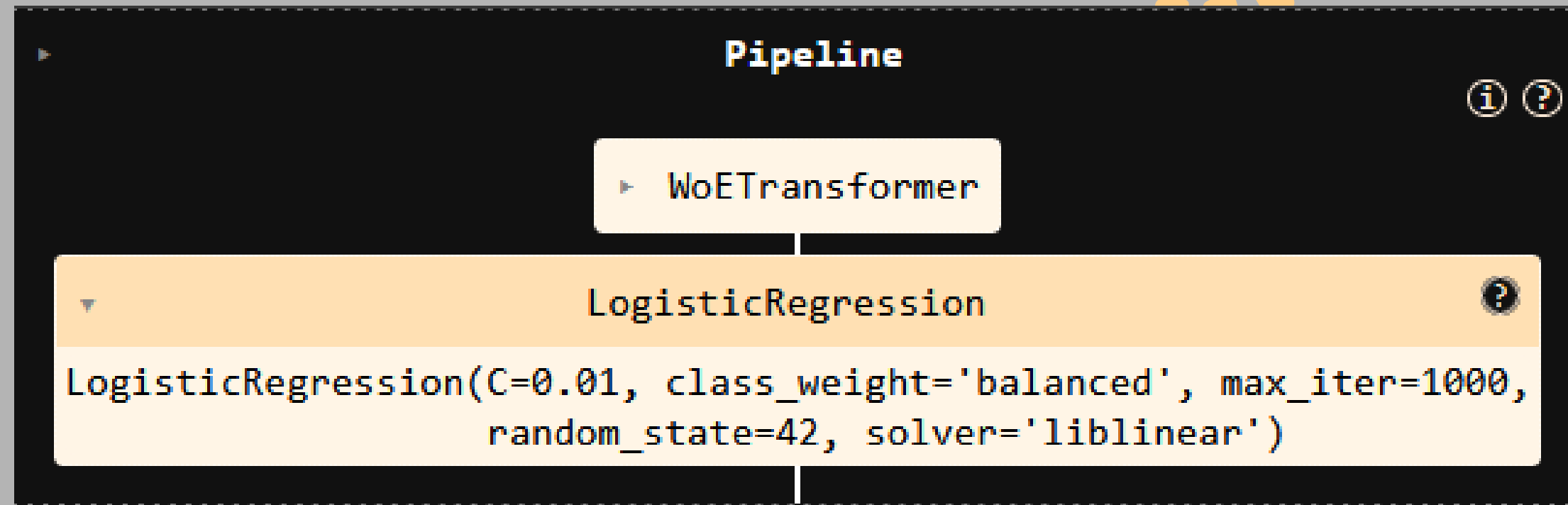
```
Modelo Usado: Logistic Regression base
AUC: 0.7770731393252639
Confusion Matrix:
[[474 232]
 [ 14  30]]
Precision Score: 0.11450381679389313
Recall Score: 0.6818181818181818
F1 Score: 0.19607843137254902
```

```
Modelo Usado: Logistic Regression 2
AUC: 0.7888552665464847
Confusion Matrix:
[[461 245]
 [ 11  33]]
Precision Score: 0.11870503597122302
Recall Score: 0.75
F1 Score: 0.20496894409937888
```

```
Modelo Usado: Logistic Regression 1
AUC: 0.7914305949008499
Confusion Matrix:
[[424 282]
 [  2  42]]
Precision Score: 0.12962962962962962
Recall Score: 0.9545454545454546
F1 Score: 0.22826086956521738
```

```
Modelo Usado: Logistic Regression 4
AUC: 0.7936840072109194
Confusion Matrix:
[[454 252]
 [  9  35]]
Precision Score: 0.12195121951219512
Recall Score: 0.7954545454545454
F1 Score: 0.21148036253776434
```

Modelo seleccionado



Scorecard

$$Score = Offset + Factor \times \ln(Odds)$$

$$Odds = \frac{P(\text{Good})}{P(\text{Bad})}$$

$$Factor = \frac{PDO}{\ln(2)}$$

$$Offset = BaseScore - Factor \times \ln(BaseOdds)$$

$$Score = Offset - Factor \cdot \text{logit}(p)$$

$$\text{logit}(p) = \beta_0 + \sum_{i=1}^n \beta_i \cdot WOE_i$$

- Para Odds que es La relación "bueno/malo" para el score base consideraremos la proporción de nuestro dataset que es 1:15 lo cual es 1 fraude cada 15 no fraudes
- PDO (Points to Double Odds): Número de puntos que representan duplicar la probabilidad de "buen cliente" vs "malo" (o viceversa). Ejemplo común: PDO = 50 Si un cliente tiene 50 puntos más, sus odds de no fraude son el doble.
- Score (Score en Odds): Puntaje asignado a un conjunto de odds de referencia, como referencia se ocupa mayormente 600

Interpretación:

- Puntaje alto: Menor riesgo de fraude
- Puntaje bajo: Mayor riesgo de fraude

Implementación en API

- Tecnología: FastAPI.
- Endpoints: /score (POST)
- Salida: probabilidad, score.

- 'Fault' -> 'Policy Holder', 'Third Party',
- 'PolicyType' -> 'Sedan - All Perils', 'Sedan - Collision', 'Sport - Liability', 'Sport - Collision', 'Utility - All Perils', 'Utility - Collision', 'Utility - Liability', 'Sport - All Perils'
- 'VehicleCategory' -> 'Sedan', 'Sport', 'Utility'
- 'Month' -> 'Aug', 'Dec', 'Feb', 'Jun', 'Jan', 'Nov', 'Jul', 'May', 'Oct', 'Sep', 'Mar', 'Apr'
- 'VehiclePrice' -> '30000 to 39000', '20000 to 29000', 'less than 20000', 'more than 69000', '40000 to 59000', '60000 to 69000'
- 'Make' -> 'Honda', 'Chevrolet', 'Pontiac', 'Toyota', 'Mazda', 'Ford', 'Accura', 'Mercury', 'VW', 'Saturn', 'Dodge', 'Saab', 'BMW', 'Nissan', 'Porsche', 'Ferrari', 'Jaguar', 'Mecedes'
- 'DayOfWeek' -> 'Friday', 'Tuesday', 'Sunday', 'Monday', 'Thursday', 'Wednesday', 'Saturday'
- 'AgeOfVehicle' -> '7 years', '5 years', '6 years', '3 years', 'more than 7', 'new', '2 years', '4 years'
- 'Deductible' -> 400, 500, 700, 300
- 'Sex' -> 'Male', 'Female'
- 'AccidentArea' -> 'Urban', 'Rural'
- 'AgentType' -> 'External', 'Internal'
- 'WeekOfMonthClaimed' -> 5, 3, 4, 2, 1

Comentarios finales

- El Modelo se puede mejorar, ocupando un poco de feature engineering y un optimizador de hiperparametros
- A pesar de que se aplico un scorecard este necesita validarse para ver si esta reflejando lo que se necesita para el negocio
- El umbral para determinar si es fraude o no fraude se puede ajustar, para mejorar la relación que existe entre recall y precision. En datasets desbalanceados, un modelo con alta precisión pero bajo recall podría “verse bien” en métricas generales pero no sirve: detecta pocos fraudes y deja escapar la mayoría. En cambio, un modelo con alto recall pero baja precisión detectará casi todos los fraudes, pero molestará a muchos clientes legítimos con revisiones innecesarias.
- Una interfaz limitada para el consumo de el API es recomendable, ya que al ser variables categóricas solo permiten ciertos valores