UNIVERSIDAD SAN FRANCISCO DE QUITO

FUNDAMENTOS DE CIENCIA DE DATOS

PROYECTO 1

TEMA: IMPACTO DE INSIDER TRADING FRENTE A LAS VARIACIONES DEL PRECIO DE LA ACCIÓN DE APPLE QUE COTIZAN EN LA BOLSA DE NUEVA YORK DESDE EL AÑO 1 DE ENERO DEL 2015 HASTA EL 27 DE MARZO 2025.

ALUMNO: SANTIAGO RODRIGUEZ

ABRIL 2025

Contenido

1. FEATURE ENGINEERING - PROCESO DE LIMPIEZA Y AGREGACIÓN DE COLUMNAS 4

1.1 Limpieza de caracteres especiales de delta_owned 4

1.2 Agregar columnas “significant_transaction” e “impacto_negativo”. 5

2. MODEL RESEARCH 7

2.1 Preparación de final_ml_dataset 7

2.2 Evaluación de modelos 8

2.2.1 Regresión logística 8

2.2.2 Árbol de decisión 9

2.2.3 Árbol de decisión controlado 10

2.2.4 Random Forest 11

2.2.5 Bagging con Random Forest 12

2.2.6 Bagging con Decision tree 13

2.2.7 Gradient Boosting Classifier 14

2.3 Análisis de desempeño 14

CONCLUSIONES 15

RECOMENDACIONES 15

Ilustración 1: primera visualización de tabla fct de AAPL 4
Ilustración 2: tratamiento inicial como borrado de caracteres especiales 4
Ilustración 3: agregar columna significant_transaction 5
Ilustración 4: creación de columna impacto_negativo 6
Ilustración 5: Datos finales para análisis de ML AAPL 6
Ilustración 6: Conteo de valores de variable objetivo 7
Ilustración 7: obtención de dummies 7
Ilustración 8 separación de datos, extracción de características y variable objetivo 8
Ilustración 9 desarrollo y evaluación de modelo de regresión logística 8
Ilustración 10 aplicación de árbol de decisión 9
Ilustración 11cálculo de relación de datos predichos frente a datos reales 9
Ilustración 12 árbol de decisión controlado 10
Ilustración 13 relación con árbol de decisión controlado 10
Ilustración 14 metricas de arbol de decisión controlado 11
Ilustración 15 comparación de datos antes y despues de realizar sobremuestreo 11
Ilustración 16 comparación de métricas RF sobremuestreados vs original 12
Ilustración 17 aplicación y métricas de bagging con random forest 13
Ilustración 18 aplicación y métricas de bagging con decision tree 13
Ilustración 19 aplicación y métricas de GDB 14

REPORTE DE FEATURE ENGINEERING Y MODEL RESEARCH

1. FEATURE ENGINEERING - PROCESO DE LIMPIEZA Y AGREGACIÓN DE COLUMNAS

1.1 Limpieza de caracteres especiales de delta_owned

- Se importan los datos de la nueva carpeta denominada “clean”

```
# lectura de datos de tablas fct
fct_precios_insiders_AAPL = pd.read_csv(('../Dataset/clean/AAPL_fct_precios_insider.csv'))
fct_precios_insiders_AAPL.head()
```

✓ 0.0s

	Unnamed: 0	trade_date	ticker	insider_name	title	trade_type	quantity_of_shares	owned	delta_owned	value	Open	Close	movimiento
0	0	2015-01-23	AAPL	Riccio Daniel J.	SVP	S	3804	0	-100%	428955	24.98	25.13	0.15
1	1	2015-02-18	AAPL	Jung Andrea	Dir	S	40000	14595	-73%	5125200	28.50	28.75	0.25
2	2	2015-03-06	AAPL	Maestri Luca	CFO	S	3400	14124	-19%	437920	28.68	28.27	-0.41
3	3	2015-03-09	AAPL	Maestri Luca	CFO	S	2800	11324	-20%	361116	28.58	28.39	-0.19
4	4	2015-03-18	AAPL	Maestri Luca	CFO	S	10823	501	-96%	1394219	28.36	28.69	0.33

Ilustración 1: primera visualización de tabla fct de AAPL

- Necesitaba quitar el símbolo de % de la columna delta_owned porque quiero realizar una métrica para saber si el insider vendió una parte significativa de sus acciones. Por ejemplo, lo quiero estimar en un umbral del 10%.
- Hay que entender que la columna “delta_owned” representa el % de acciones que tiene esa persona **luego** de haber realizado una transacción, por ejemplo, si es venta el “delta_owned” será una disminución, si es compra será un aumento.
- Para el caso de las personas que su “delta_owned” es 0, no quiere decir que no hicieron transacciones, quiere decir que el porcentaje negociado es ínfimo frente al total de las acciones que poseen. Esto se puede evidenciar en la línea 220 de la tabla fct.

```
# eliminar el simbolo % y el guion - de la columna "delta_owned".
fct_precios_insiders_AAPL['delta_owned'] = fct_precios_insiders_AAPL['delta_owned'].str.replace('%', '').str.replace('-', '')
```

✓ 0.0s

```
fct_precios_insiders_AAPL
```

✓ 0.0s

	Unnamed: 0	trade_date	ticker	insider_name	title	trade_type	quantity_of_shares	owned	delta_owned	value	Open	Close	movimiento
0	0	2015-01-23	AAPL	Riccio Daniel J.	SVP	S	3804	0	100	428955	24.98	25.13	0.15
1	1	2015-02-18	AAPL	Jung Andrea	Dir	S	40000	14595	73	5125200	28.50	28.75	0.25
2	2	2015-03-06	AAPL	Maestri Luca	CFO	S	3400	14124	19	437920	28.68	28.27	-0.41
3	3	2015-03-09	AAPL	Maestri Luca	CFO	S	2800	11324	20	361116	28.58	28.39	-0.19
4	4	2015-03-18	AAPL	Maestri Luca	CFO	S	10823	501	96	1394219	28.36	28.69	0.33
...	...	...	...	...	...	...	...	...	...	...	...	...	...
216	216	2024-10-04	AAPL	Maestri Luca	CFO	S	59305	107788	35	13433769	227.40	226.30	-1.10
217	217	2024-11-15	AAPL	Levinson Arthur D	Dir	S	200000	4215576	5	45464500	226.15	224.75	-1.40
218	218	2024-11-18	AAPL	Kondo Chris	PAO	S	4130	15419	21	945233	225.00	227.77	2.77
219	219	2024-12-16	AAPL	Williams Jeffrey E	COO	S	100000	389944	20	24997395	247.72	250.76	3.04
220	220	2025-02-03	AAPL	Levinson Arthur D	Dir	S	1516	4215576	0	343147	229.74	227.76	-1.98

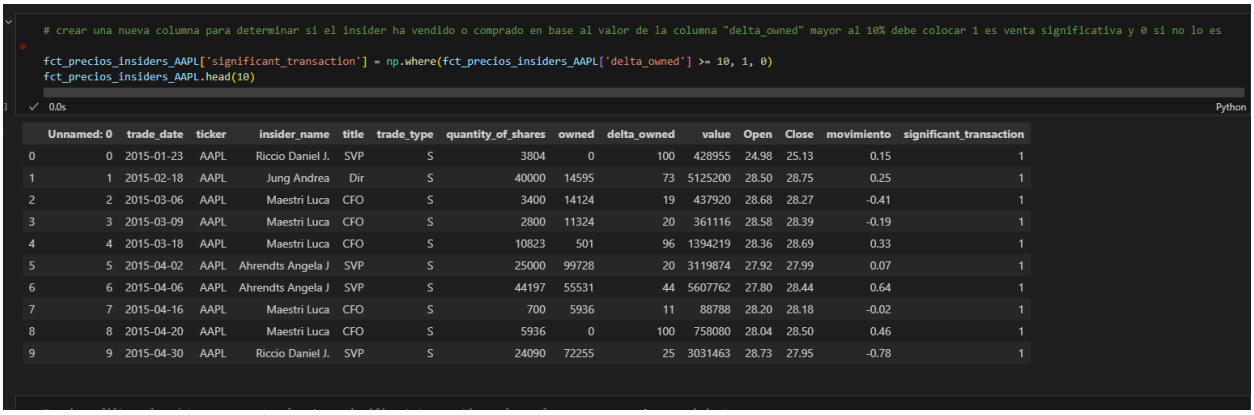
221 rows × 13 columns

Ilustración 2: tratamiento inicial como borrado de caracteres especiales

- Se transforma la columna “delta_owned” en tipo de dato enteros.

1.2 Agregar columnas “significant_transaction” e “impacto_negativo”.

- Agrego una columna para evaluar su impacto como transacción significativa, es decir, si la transacción que dio lugar a ese día fue una transacción mayor a 10% del delta_owned. Si es así, lo señala como 1, caso contrario como 0.



```
# crear una nueva columna para determinar si el insider ha vendido o comprado en base al valor de la columna "delta_owned" mayor al 10% debe colocar 1 es venta significativa y 0 si no lo es

fct_precios_insiders_AAPL['significant_transaction'] = np.where(fct_precios_insiders_AAPL['delta_owned'] >= 10, 1, 0)
fct_precios_insiders_AAPL.head(10)
```

0.0s Python

	Unnamed: 0	trade_date	ticker	insider_name	title	trade_type	quantity_of_shares	owned	delta_owned	value	Open	Close	movimiento	significant_transaction
0	0	2015-01-23	AAPL	Riccio Daniel J.	SVP	S	3804	0	100	428955	24.98	25.13	0.15	1
1	1	2015-02-18	AAPL	Jung Andrea	Dir	S	40000	14595	73	5125200	28.50	28.75	0.25	1
2	2	2015-03-06	AAPL	Maestri Luca	CFO	S	3400	14124	19	437920	28.68	28.27	-0.41	1
3	3	2015-03-09	AAPL	Maestri Luca	CFO	S	2800	11324	20	361116	28.58	28.39	-0.19	1
4	4	2015-03-18	AAPL	Maestri Luca	CFO	S	10823	501	96	1394219	28.36	28.69	0.33	1
5	5	2015-04-02	AAPL	Ahrendts Angela J	SVP	S	25000	99728	20	3119874	27.92	27.99	0.07	1
6	6	2015-04-06	AAPL	Ahrendts Angela J	SVP	S	44197	55531	44	5607762	27.80	28.44	0.64	1
7	7	2015-04-16	AAPL	Maestri Luca	CFO	S	700	5936	11	88788	28.20	28.18	-0.02	1
8	8	2015-04-20	AAPL	Maestri Luca	CFO	S	5936	0	100	758080	28.04	28.50	0.46	1
9	9	2015-04-30	AAPL	Riccio Daniel J.	SVP	S	24090	72255	25	3031463	28.73	27.95	-0.78	1

Ilustración 3: agregar columna significant_transaction

- A partir de aquí agrego una columna denominada “impacto_negativo” para identificar todas las transacciones que fueron **SIGNIFICATIVAS Y QUE MOVIMIENTO TENGA VALORES NEGATIVOS.** Con esto quiero determinar si el insider sabía lo que iba a ocurrir y por eso decidió negociar sus acciones en ese momento antes de que el precio baje.

  Aquí quisiera aclarar algo:

1. Cuando alguien compra una acción es porque considera que el desempeño de esa empresa será positivo, en base a resultados financieros de años anteriores, muestra solidez y es una marca posicionada en el mercado, entre los factores más comunes. Con este análisis, se “espera” que cuando se compra una acción el precio suba y se pueda obtener ganancias al vender al largo plazo.
2. Por otro lado, cuando alguien vende acciones, considera todo lo contrario, hay pésimos resultados financieros, corren rumores dentro de la industria, cambio de directores, productos con fallas, etc. Con la venta de una acción viene atado el concepto de “especulación de mercado” que hace que el precio baje en el corto plazo. Muy difícilmente ese precio vuelve a recuperarse en el corte plazo y deberán esperar varios meses, incluso años, para que el precio regreso a niveles anteriores.

  Por lo tanto:

- Determiné un umbral del 10% porque tomando en cuenta que son directores de la compañía tienen bastante influencia y un movimiento de acciones por más pequeño que sea debe ser considerado como una alerta.

- Mi análisis será a partir de verificar el impacto con el cambio del precio de la acción en negativo, es decir, si un director que haya vendido por lo menos el 10% de sus acciones afecta al precio de la acción en negativo, se consideraría un resultado positivo.

```
# crear columna para determinar el impacto negativo si el insider ha vendido en base al valor de movimiento y significant_transaction, que el output sea true o false, con inplace=true
fct_precios_insiders_AAPL['impacto_negativo'] = fct_precios_insiders_AAPL.apply(
    lambda x: True if (x['movimiento'] < 0 and x['significant_transaction'] == 1) else False, axis=1
)
```

✓ 0.0s Python

```
fct_precios_insiders_AAPL
```

✓ 0.0s Python

	Unnamed: 0	trade_date	ticker	insider_name	title	trade_type	quantity_of_shares	owned	delta_owned	value	Open	Close	movimiento	significant_transaction	impacto_negativo
0	0	2015-01-23	AAPL	Riccio Daniel J.	SVP	S	3804	0	100	428955	24.98	25.13	0.15	1	False
1	1	2015-02-18	AAPL	Jung Andrea	Dir	S	40000	14595	73	5125200	28.50	28.75	0.25	1	False
2	2	2015-03-06	AAPL	Maestri Luca	CFO	S	3400	14124	19	437920	28.68	28.27	-0.41	1	True
3	3	2015-03-09	AAPL	Maestri Luca	CFO	S	2800	11324	20	361116	28.58	28.39	-0.19	1	True
4	4	2015-03-18	AAPL	Maestri Luca	CFO	S	10823	501	96	1394219	28.36	28.69	0.33	1	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
216	216	2024-10-04	AAPL	Maestri Luca	CFO	S	59305	107788	35	13433769	227.40	226.30	-1.10	1	True
217	217	2024-11-15	AAPL	Levinson Arthur D	Dir	S	200000	4215576	5	45464500	226.15	224.75	-1.40	0	False
218	218	2024-11-18	AAPL	Kondo Chris	PAO	S	4130	15419	21	945233	225.00	227.77	2.77	1	False
219	219	2024-12-16	AAPL	Williams Jeffrey E	COO	S	100000	389944	20	24997395	247.72	250.76	3.04	1	False
220	220	2025-02-03	AAPL	Levinson Arthur D	Dir	S	1516	4215576	0	343147	229.74	227.76	-1.98	0	False

221 rows × 15 columns

Ilustración 4: creación de columna impacto_negativo

- Separo la fecha por mes y año.

Así quedaría:

```
fct_precios_insiders_AAPL['año'] = pd.to_datetime(fct_precios_insiders_AAPL['trade_date']).dt.year
fct_precios_insiders_AAPL['mes'] = pd.to_datetime(fct_precios_insiders_AAPL['trade_date']).dt.month
```

✓ 0.0s Python

```
fct_precios_insiders_AAPL
```

✓ 0.0s Python

	Unnamed: 0	trade_date	ticker	insider_name	title	trade_type	quantity_of_shares	owned	delta_owned	value	Open	Close	movimiento	significant_transaction	impacto_negativo	año	mes
0	0	2015-01-23	AAPL	Riccio Daniel J.	SVP	S	3804	0	100	428955	24.98	25.13	0.15	1	False	2015	1
1	1	2015-02-18	AAPL	Jung Andrea	Dir	S	40000	14595	73	5125200	28.50	28.75	0.25	1	False	2015	2
2	2	2015-03-06	AAPL	Maestri Luca	CFO	S	3400	14124	19	437920	28.68	28.27	-0.41	1	True	2015	3
3	3	2015-03-09	AAPL	Maestri Luca	CFO	S	2800	11324	20	361116	28.58	28.39	-0.19	1	True	2015	3
4	4	2015-03-18	AAPL	Maestri Luca	CFO	S	10823	501	96	1394219	28.36	28.69	0.33	1	False	2015	3
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
216	216	2024-10-04	AAPL	Maestri Luca	CFO	S	59305	107788	35	13433769	227.40	226.30	-1.10	1	True	2024	10
217	217	2024-11-15	AAPL	Levinson Arthur D	Dir	S	200000	4215576	5	45464500	226.15	224.75	-1.40	0	False	2024	11
218	218	2024-11-18	AAPL	Kondo Chris	PAO	S	4130	15419	21	945233	225.00	227.77	2.77	1	False	2024	11
219	219	2024-12-16	AAPL	Williams Jeffrey E	COO	S	100000	389944	20	24997395	247.72	250.76	3.04	1	False	2024	12
220	220	2025-02-03	AAPL	Levinson Arthur D	Dir	S	1516	4215576	0	343147	229.74	227.76	-1.98	0	False	2025	2

221 rows × 17 columns

Ilustración 5: Datos finales para análisis de ML AAPL

- Los datos están listos para avanzar con el proceso de machine learning. Guardo esta tabla en la carpeta ml_dataset como un nuevo archivo .csv

2. MODEL RESEARCH

A partir de ahora voy a Construir un modelo que prediga si una venta significativa de insider generará un movimiento negativo en el precio — es decir, si "sabían algo" antes de que la acción baje.

2.1 Preparación de final_ml_dataset

- Empiezo con contar los valores que se encuentra en la columna “impacto_negativo”, de ahora en adelante, mi variable objetivo.

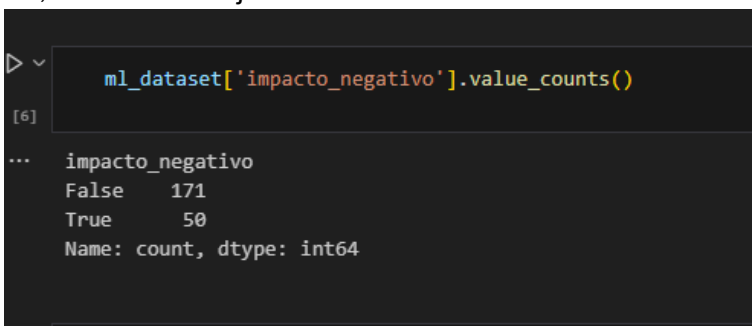
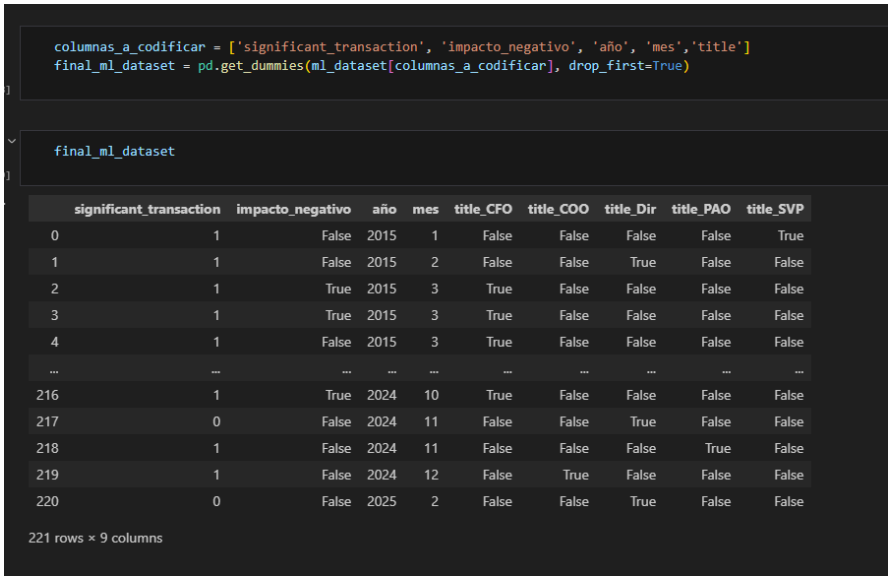


Ilustración 6: Conteo de valores de variable objetivo

- Transformo a dummies las columnas de significant_transaction, impacto_negativo, año, mes, title.



	significant_transaction	impacto_negativo	año	mes	title_CFO	title_COO	title_Dir	title_PAO	title_SVP
0	1	False	2015	1	False	False	False	False	True
1	1	False	2015	2	False	False	True	False	False
2	1	True	2015	3	True	False	False	False	False
3	1	True	2015	3	True	False	False	False	False
4	1	False	2015	3	True	False	False	False	False
...	...	...	...	...	...	...	...	...	...
216	1	True	2024	10	True	False	False	False	False
217	0	False	2024	11	False	False	True	False	False
218	1	False	2024	11	False	False	False	True	False
219	1	False	2024	12	False	True	False	False	False
220	0	False	2025	2	False	False	True	False	False

Ilustración 7: obtención de dummies

- Divido los datos en 3 grupos, entrenamiento, validación y prueba y separo las características de mi variable objetivo.

```
entrenamiento_validacion, prueba = train_test_split(final_ml_dataset, test_size=0.2, random_state=12345)

entrenamiento, validacion = train_test_split(entrenamiento_validacion, test_size=0.2, random_state=12345)
```

```
entrenamiento['impacto_negativo'].value_counts()
```

```
impacto_negativo
False    105
True      35
Name: count, dtype: int64
```

```
entrenamiento_caracteristicas = entrenamiento.drop(['impacto_negativo'], axis=1)
entrenamiento_objetivo = entrenamiento['impacto_negativo']

validacion_caracteristicas = entrenamiento.drop(['impacto_negativo'], axis=1)
validacion_objetivo = entrenamiento['impacto_negativo']

prueba_caracteristicas = entrenamiento.drop(['impacto_negativo'], axis=1)
prueba_objetivo = entrenamiento['impacto_negativo']
```

Ilustración 8 separación de datos, extracción de características y variable objetivo

2.2 Evaluación de modelos

2.2.1 Regresión logística

```
reg_log = LogisticRegression(random_state=12345)
```

```
reg_log.fit(entrenamiento_caracteristicas, entrenamiento_objetivo)
```

```
LogisticRegression
LogisticRegression(random_state=12345)
```

```
prediccion_entrenamiento = reg_log.predict(entrenamiento_caracteristicas)
```

```
pd.Series(prediccion_entrenamiento).value_counts()
```

```
False    140
Name: count, dtype: int64
```

Ilustración 9 desarrollo y evaluación de modelo de regresión logística

- Aquí vemos que el RL no me sirve ya que no me está arrojando datos TRUE que quiere decir que no impactó el precio de la acción. Por lo tanto, lo descarto y sigo probando modelos.

2.2.2 Árbol de decisión

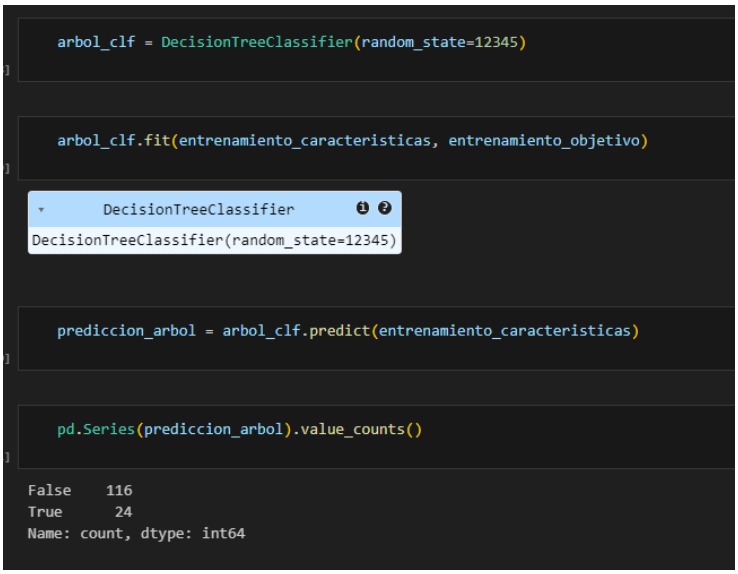


Ilustración 10 aplicación de árbol de decisión

- Con este modelo se observa que me arroja resultados como true y realizo la relación frente a los falsos, ya que me parece que están siendo muy pocos los datos que se predijeron.

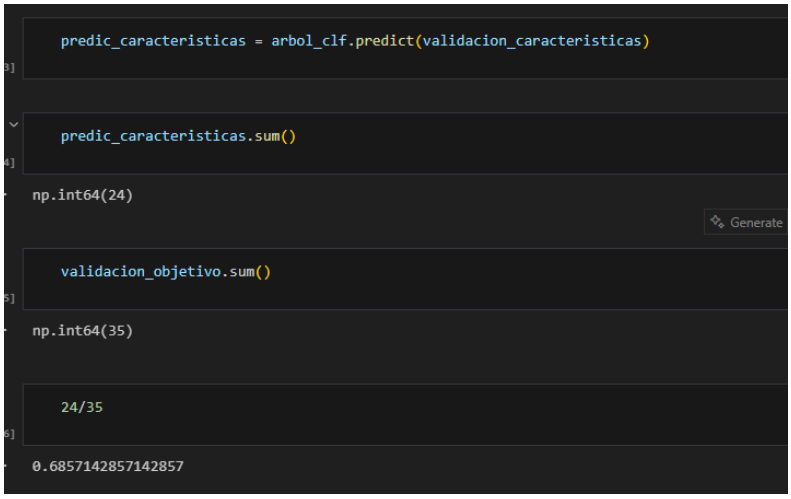


Ilustración 11cálculo de relación de datos predichos frente a datos reales

2.2.3 Árbol de decisión controlado

- Escojo hacer un árbol en donde determino el número de capas máximas a 10 ya que el modelo anterior tenía un total de 17 capas.

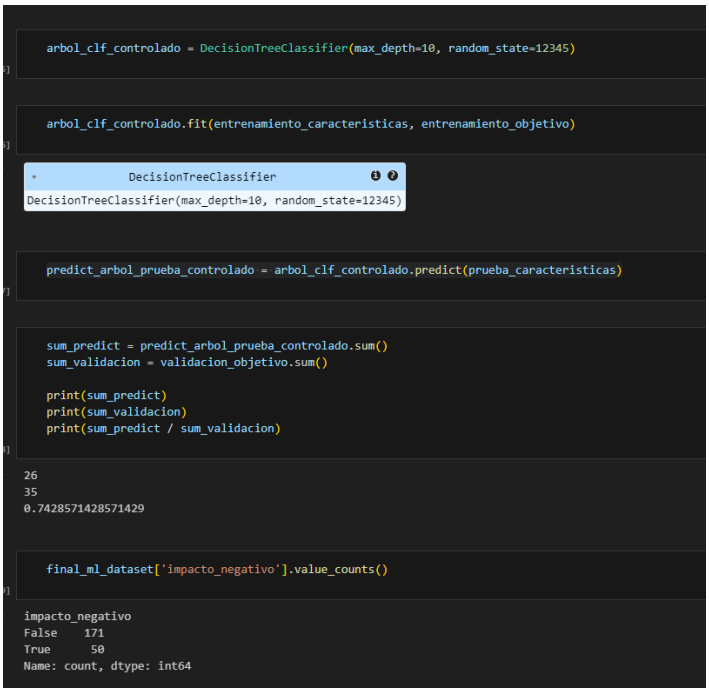


Ilustración 12 árbol de decisión controlado

- En esta parte me detengo a evaluar si debo realizar un sobremuestreo, ya que, 50 datos tengo como verdaderos y 171 como falsos, aquí nuevamente realizo un cálculo de la relación.

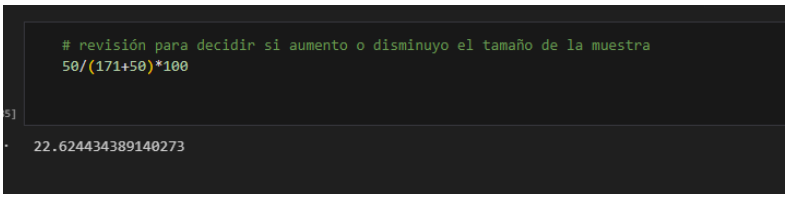


Ilustración 13 relación con árbol de decisión controlado

- Esto quiere decir que solo el 22% de mis valores verdaderos representan la totalidad de mis registros. Acompaño este análisis con las métricas para determinar si hago sobremuestreo.

```
El total de registros en validacion es: (140,)
El total de impactos negativos reales es: 35
El total de impactos negativos que se predice es: 26

La recuperacion con datos de validacion es de: 0.6
La precision con datos de validacion es de: 0.8076923076923077
El f1-score con datos de validacion es de: 0.6885245901639344
```

Ilustración 14 metricas de arbol de decisión controlado

- Como los resultados que arroja son bajos, decido hacer el sobremuestreo.

  Mi tasa de crecimiento es 2, ya que antes de aplicar sobremuestreo, el grupo de resultados de mi variable objetivo representaba el 22%, ahora representa el 66%. Con esto los datos aumentaron a 35 nuevos registros sintéticos, se puede ver la compración con .shape

  No quise aplicar una tasa de crecimiento = 3 ya que igualaría la totalidad de los verdaderos y positivos, según mi punto de vista aumentaría en mayor forma el sesgo. Mayor tasa de crecimiento, mayor sesgo.

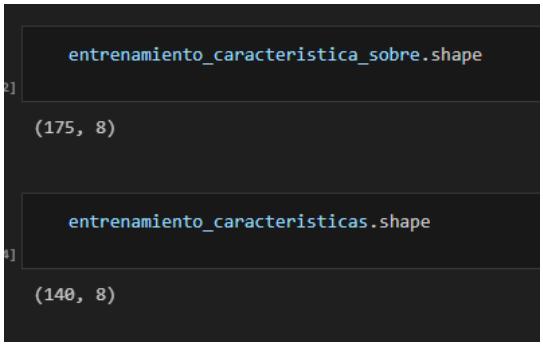


Ilustración 15 comparación de datos antes y despues de realizar sobremuestreo

2.2.4 Random Forest

- Para el caso del modelo Random Forest, decidí aplicar el modelo tanto para los datos sobremuestreados como para los datos originales. Aquí están las métricas de cada uno.

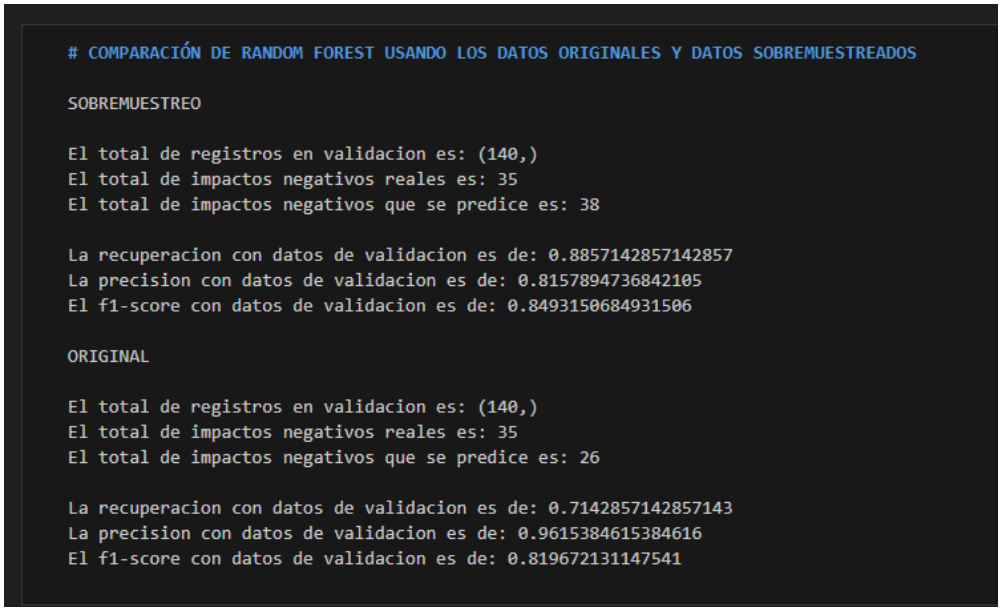


Ilustración 16 comparación de métricas RF sobremuestreados vs original

- Se observa que las métricas para el caso de los datos sobremuestreados arrojan un mejor desempeño frente a los datos normales.
- Estaba probando el número de estimadores y al encontrar que con 500 estimadores, el modelo arrojó los mejores resultados. Por lo tanto decido aplicar esta cantidad de estimadores con los datos originales y sobremuestreados por igual.

    1. Hay que entender que:
    2. Recuperación: De todos los positivos reales, cuántos o qué porcentaje fueron detectados por el modelo
    3. Precisión: De todos los positivos que predijo el modelo, cuántos o qué porcentaje eran realmente positivos.
    4. F1 score: Arroja un balance entre precisión y recall.

- Los datos que se predice es de 38, con una recuperación del 88%, precisión del 81% y f1 de 85%. De momento es el mejor modelo.

2.2.5 Bagging con Random Forest

- Por testear los límites, realicé un modelo bagging con un estimador de random forest. Aquí los resultados.

```
bag_clf = BaggingClassifier(
    estimator=RandomForestClassifier(),
    n_estimators=500,
    max_samples=150,
    random_state=12345

)
```

```
bag_clf.fit(entrenamiento_caracteristica_sobre, entrenamiento_objetivo_sobre)
```

BaggingClassifier

estimator: RandomForestClassifier

RandomForestClassifier

```
pred_bag_clf = bag_clf.predict(validacion_caracteristicas)
metricas(bag_clf, pred_bag_clf, validacion_objetivo)
```

```
El total de registros en validacion es: (140,)
El total de impactos negativos reales es: 35
El total de impactos negativos que se predice es: 43

La recuperacion con datos de validacion es de: 0.9142857142857143
La precision con datos de validacion es de: 0.7441860465116279
El f1-score con datos de validacion es de: 0.8205128205128205
```

Ilustración 17 aplicación y métricas de bagging con random forest

2.2.6 Bagging con Decision tree

```
bag_clf_arbol = BaggingClassifier(
    estimator=DecisionTreeClassifier(max_features='sqrt', splitter="random"),
    n_estimators=500,
    max_samples=150,
    random_state=12345,
    n_jobs=-1

)

bag_clf_arbol.fit(entrenamiento_caracteristica_sobre, entrenamiento_objetivo_sobre)
pred_bag_clf_arbol = bag_clf_arbol.predict(validacion_caracteristicas)
metricas(bag_clf_arbol, pred_bag_clf_arbol, validacion_objetivo)
```

```
El total de registros en validacion es: (140,)
El total de impactos negativos reales es: 35
El total de impactos negativos que se predice es: 38

La recuperacion con datos de validacion es de: 0.8857142857142857
La precision con datos de validacion es de: 0.8157894736842105
El f1-score con datos de validacion es de: 0.8493150684931506
```

Ilustración 18 aplicación y métricas de bagging con decision tree

2.2.7 Gradient Boosting Classifier

```
grad_clf = GradientBoostingClassifier(
    n_estimators=500,
    learning_rate=0.15,
    max_depth=30,
    random_state=12345,
)

grad_clf.fit(entrenamiento_caracteristica_sobre, entrenamiento_objetivo_sobre)
pred_grad_clf = grad_clf.predict(validacion_caracteristicas)
metricas(grad_clf, pred_grad_clf, validacion_objetivo)
```

```
El total de registros en validacion es: (140,)
El total de impactos negativos reales es: 35
El total de impactos negativos que se predice es: 38

La recuperacion con datos de validacion es de: 0.8857142857142857
La precision con datos de validacion es de: 0.8157894736842105
El f1-score con datos de validacion es de: 0.8493150684931506
```

Ilustración 19 aplicación y métricas de GDB

2.3 Análisis de desempeño

Árbol de decisión controlado:

- Tiene el peor recall (0.60), lo que significa que omite muchos impactos negativos reales.
- Precisión decente (0.808), pero el F1 es bajo (0.689).
- No recomendable si el objetivo es detectar la mayor cantidad de impactos negativos.

Random Forest con sobremuestreo, Bagging y Boosting:

- Recall alto (0.886–0.914)
- Precisión sólida (~0.816)
- F1-score más alto (≈0.849), lo que indica buen equilibrio entre precisión y sensibilidad.
- Recomendables si el objetivo fuera detectar la mayoría de los impactos negativos como reducir falsos positivos.

Random Forest con datos originales:

- Tiene la mayor precisión (0.962): cuando predice impacto negativo, casi siempre acierta.
- Pero el recall cae (0.714): omite varios casos reales.
- Ideal si se prefiere evitar falsos positivos, pero aceptar que no va a detectar todos los casos negativos.

CONCLUSIONES

- Los modelos **Boosting por Descenso de Gradiente**, **Random Forest con sobremuestreo** y **Bagging con Árbol de Decisión** obtienen los **f1-score más altos (alrededor de 0.85)**, lo que refleja un equilibrio adecuado entre precisión y recall. Estos modelos demuestran un rendimiento consistente y robusto en la detección de impactos negativos.
- En base a los resultados expuestos, personalmente considero que el modelo que mejor ha desplegado las predicciones es el de RANDOM FOREST con la mejor calificación con relación a la métrica f1. Obviamente se puede determinar otro modelo que se acerque con más precisión a las predicciones, sin embargo, como un primer acercamiento a modelos predictivos me pareció interesante dar un repaso por todos los modelos e ir evaluando el impacto de cada uno de ellos.

RECOMENDACIONES

- Si el objetivo principal del análisis es **maximizar la detección de impactos negativos**, incluso a costa de aceptar algunos falsos positivos, se recomienda utilizar modelos como **Bagging con Random Forest** o **Boosting**.
- Si en cambio se desea **minimizar los falsos positivos** y se prioriza la certeza en cada predicción, el modelo más apropiado es **Random Forest con datos originales**.
- Para un **equilibrio general entre precisión y recall**, los modelos como **Boosting** y **Bagging con árbol de decisión** ofrecen resultados sólidos y consistentes, siendo opciones adecuadas para aplicaciones en producción.