MARKET BYTES

BIG ANALITICS

Informe Tecnico — Metodologia y Modelos Aplicados

Pipeline de Big Analytics: 20 modelos, 270 features, 26 empresas

Marzo 2026

Indice de Contenidos

[El indice de contenidos se genera automaticamente en Word: Referencias > Tabla de contenido]

1. Resumen Tecnico

Este informe documenta la metodologia tecnica completa del proyecto Big Analitics aplicado al sector FinTech. El objetivo es proporcionar una referencia detallada de cada algoritmo, su fundamentacion teorica, implementacion en Python y resultados obtenidos. El documento esta disenado para un lector que desee comprender y reproducir el analisis.

Panorama del Pipeline

El sistema se organiza en un pipeline de 6 fases secuenciales que transforman 78 archivos Excel crudos de S&P Capital IQ en 20 modelos analiticos, 12 visualizaciones y 3 informes finales. El pipeline procesa 26 empresas FinTech con un total de 127,927 observaciones crudas, genera un feature store de 769 filas y 270 features avanzadas, y ejecuta 20 modelos que cubren desde clustering no supervisado hasta regresion cuantilica y analisis de supervivencia.

Tabla 1.1. Cifras clave del proyecto

Dimension	Valor
Empresas analizadas	26 (14 Standard + 5 SNL Banking + 7 adicionales)
Observaciones crudas (master_long)	127,927 filas
Features base (feature_store)	242 columnas
Features avanzadas (feature_store_advanced)	270 columnas
Filas en feature_store	769
Modelos implementados	20
Visualizaciones	12 graficos PNG
Metricas mapeadas	304

Fuente: Elaboracion propia.

Stack Tecnologico Principal

El proyecto esta desarrollado integramente en Python 3.x con las siguientes librerias principales: pandas (manipulacion de datos tabulares), scikit-learn (modelos de ML: clustering, ensemble, anomaly detection), statsmodels (regresion cuantilica, descomposicion de series), networkx (analisis de redes), matplotlib/seaborn (visualizacion), python-docx (generacion de informes Word), openpyxl (lectura de Excel), pyarrow (formato Parquet). Todo el pipeline se controla mediante un archivo de configuracion JSON que permite adaptar el analisis a cualquier sector sin cambiar codigo.

2. Arquitectura del Sistema

La arquitectura del proyecto sigue el principio de separacion entre codigo reutilizable (framework) y outputs especificos (entregables). Este diseno permite escalar el analisis a multiples sectores sin duplicar codigo.

2.1 Flujo del Pipeline

El pipeline transforma datos crudos en inteligencia accionable a traves de 6 fases:

```
Raw Excel (78 archivos)
    |-> Fase 1: Data Ingestion (capiq_parser + mapper + normalizer + assembler +
validator)
        |-> master_long.parquet (127,927 filas x 25 cols)
    |-> Fase 2: Data Processing (integrado en Fase 1)
    |-> Fase 3: Feature Engineering (ratios + percentiles + z-scores + lags + rolling)
        |-> feature_store.parquet (769 x 242)
        |-> feature_store_advanced.parquet (769 x 270)
    |-> Fase 4: Analysis & Models (20 modelos: 4A..4V)
        |-> 11 parquet + 11 JSON
    |-> Fase 5: Visualization (12 graficos PNG)
    |-> Fase 6: Reporting (Word + PowerPoint)
```

2.2 Estructura de Directorios

El repositorio se divide en dos arboles principales:

```
BIG_ANALITICS_ANALISIS_SECTORIAL/
|
|-- _ARQUITECTURA/                    <- Framework reutilizable
|   |-- config/                       <- sector_config_*.json
|   |-- 01_Data_Ingestion/scripts/    <- Parser CAPIQ (7 modulos)
|   |-- 03_Feature_Engineering/scripts/
|   |-- 04_Analysis_Models/scripts/   <- 20 modulos de modelos
|   |-- 05_Visualization/scripts/
|   |-- 06_Reporting/scripts/
|   |-- skills/                       <- Skills validados
|   |-- corrections_log/              <- Registro de correcciones
|
|-- _ENTREGABLES/                     <- Outputs por sector
|   |-- Sector_FinTech/
|       |-- 01_Datos_Procesados/      <- master_long, company_metadata
|       |-- 02_Feature_Store/         <- feature_store, feature_catalog
|       |-- 03_Analysis_Models/       <- Parquets + JSONs de cada modelo
|       |-- 04_Visualizaciones/       <- 12 PNGs
|       |-- 05_Reporting/             <- Informes Word/PPTX
|
|-- Sector_FinTech/                   <- Data cruda CAPIQ (Excel)
|-- Sector_FinTech_SNL_Banking/       <- Data cruda CAPIQ (SNL template)
```

Fuente: Elaboracion propia.

2.3 Diseno Basado en Configuracion

Cada sector se define mediante un archivo JSON de configuracion (sector_config_fintech.json) que controla todos los parametros del pipeline sin necesidad de modificar el codigo. Esto permite replicar el analisis completo en un nuevo sector creando solo un archivo de configuracion.

El archivo de configuracion incluye: nombre del sector, lista de empresas con su tier y metadata, parametros del parser (template, moneda, periodicidad), lista de features a calcular, parametros de cada modelo (numero de clusters, alpha de Ridge, contamination de Isolation Forest), y rutas de salida.

```
# Ejemplo de uso:
# python master_runner.py --config config/sector_config_fintech.json --phases 1,3,4,5

# El config define:
# - sector_name: "Sector_FinTech"
# - companies: [{id: "paypal", tier: 1, currency: "USD", frequency: "quarterly"}, ...]
# - features: {ratios: [...], normalization: {method: "mad_zscore", winsorize: [5, 95]}}
# - models: {clustering: {n_clusters: "auto", features: [...]}, ridge: {alpha: 1.0}, ...}
```

2.4 Patron de Modulo

Cada script de modelo sigue un patron consistente de entrada-proceso-salida:

```
def run_model(feature_store: pd.DataFrame, config: dict) -> tuple[pd.DataFrame, dict]:
    """Patron estandar de cada modulo de modelo.
    Input:  feature_store (DataFrame) + config (dict)
    Output: results (DataFrame, guardado como .parquet) + summary (dict, guardado como
.json)
    """
    # 1. Seleccionar features relevantes
    # 2. Preprocesar (imputacion, winsorize, escalar)
    # 3. Ejecutar algoritmo
    # 4. Generar resultados tabulares + resumen JSON
    return results_df, summary_dict
```

Este patron garantiza que cada modelo es autocontenido, testeable y produce outputs persistentes en formatos estandar (Parquet para tablas, JSON para metadata y resumen).

3. Fase 1: Data Ingestion

3.1 Fundamento Teorico: Pipelines ETL

ETL (Extract, Transform, Load) es el patron arquitectonico fundamental para mover datos desde fuentes heterogeneas a un almacen estructurado. En el contexto de datos financieros, el ETL enfrenta desafios especificos: formatos semiestructurados (Excel con metadata variable), convenciones de nomenclatura inconsistentes entre proveedores, multiples monedas, distintas periodicidades temporales y variaciones en la estructura de los templates.

La normalizacion de datos financieros implica tres transformaciones clave: (1) normalizacion de esquema (mapear etiquetas variantes a un vocabulario canonico), (2) normalizacion monetaria (conversion FX uniforme), y (3) normalizacion temporal (alineacion de periodos fiscales a trimestres calendario). El resultado es un formato 'long' donde cada fila representa una observacion unica: empresa-periodo-metrica-valor.

3.2 Implementacion: Modulos del Parser

El parser CAPIQ se compone de 7 modulos Python, cada uno responsable de una etapa del ETL:

Tabla 3.1. Modulos del parser CAPIQ

Modulo	Funcion	Input	Output
capiq_parser.py	Deteccion de template, extraccion de metadata	Excel .xlsx	raw dict
capiq_mapper.py	Matching de etiquetas al esquema canonico	raw dict + standard_mapping.json	mapped dict
capiq_normalizer.py	Conversion FX (spot/average)	mapped dict + FX rates	normalized dict
capiq_assembler.py	Ensamblaje en formato long	normalized dicts	master_long DataFrame
capiq_validator.py	Validacion 3 niveles	master_long	quality_report.json

master_runner.py	Orquestacion del pipeline	config JSON	Todos los outputs
standard_mapping.json	Mapa de ~160 metricas	—	—

Fuente: Elaboracion propia.

Deteccion de Template

El parser identifica automaticamente el template CAPIQ del archivo Excel mediante la deteccion de cadenas clave en las primeras filas. S&P Capital IQ utiliza tres templates: Standard (mas comun), SNL Financial (para bancos), y Third Party Private (para empresas privadas). La deteccion determina que reglas de parsing aplicar para extraer los datos correctamente.

```
# capiq_parser.py — Deteccion de template
def detect_template(ws):
    """Detecta el template CAPIQ escaneando las primeras 10 filas."""
    for row in ws.iter_rows(min_row=1, max_row=10, values_only=True):
        text = ' '.join(str(c) for c in row if c)
        if 'SNL' in text: return 'snl_financial'
        if 'Third Party' in text: return 'third_party_private'
    return 'standard'
```

Matching de Etiquetas

El mapper implementa un sistema de matching en 3 niveles para resolver la variabilidad de nombres de metricas entre archivos:

Nivel 1 — Match exacto: la etiqueta del Excel coincide textualmente con el esquema canonico (e.g., 'Total Revenue' -> total_revenue). Resuelve ~85% de los casos.

Nivel 2 — Match normalizado: se aplica normalizacion case-insensitive, eliminacion de espacios extra y caracteres especiales. Resuelve ~12% adicional.

Nivel 3 — Etiquetas no mapeadas: se registran para revision manual y se excluyen del output. Representan ~3% de etiquetas y generalmente son notas al pie o subtotales redundantes.

```
# capiq_mapper.py — Matching jerarquico
def match_label(raw_label, mapping):
```

```
    # Nivel 1: exacto
    if raw_label in mapping:
        return mapping[raw_label]
    # Nivel 2: normalizado
    norm = raw_label.strip().lower().replace('  ', ' ')
    for key, val in mapping.items():
        if key.strip().lower() == norm:
            return val
    # Nivel 3: no mapeado
    return None
```

Normalizacion FX

La conversion de moneda sigue las convenciones contables estandar: las metricas de Balance Sheet (stocks, medidas en un punto del tiempo) se convierten usando el tipo de cambio spot al cierre del periodo. Las metricas de Income Statement y Cash Flow (flujos, acumulados durante el periodo) se convierten usando el tipo de cambio promedio del periodo. Esta distincion es importante porque usar el rate incorrecto introduce distorsiones artificiales en los ratios.

3.3 Resultados del Ingestion

El parser procesa 78 archivos Excel (3 por empresa: Income Statement, Balance Sheet, Cash Flow) y genera un master_long con 127,927 filas. Se mapearon exitosamente 304 metricas financieras para 26 empresas, cubriendo periodos desde 2011 hasta 2025.

Tabla 3.2. Estructura del master_long (columnas principales)

Columna	Tipo	Descripcion	Ejemplo
company_id	str	Identificador unico de empresa	paypal
period	str	Periodo temporal	2024Q3
metric_name	str	Nombre canonico de la metrica	total_revenue
value	float64	Valor numerico en USD thousands	7,247,000
statement_type	str	Tipo de estado financiero	income_statement

template	str	Template CAPIQ detectado	standard
currency_original	str	Moneda original	USD
frequency	str	Periodicidad	quarterly
tier	int	Tier de calidad (1-3)	1

Fuente: Elaboracion propia.

4. Fase 3: Feature Engineering

4.1 Fundamento Teorico: Analisis de Ratios Financieros

El analisis de ratios financieros es la piedra angular del analisis fundamental. Un ratio normaliza una metrica financiera por una medida de escala (generalmente ingresos, activos o equity), lo que permite comparar empresas de distinto tamano. La literatura clasica (Graham & Dodd, Penman) organiza los ratios en categorias que capturan diferentes dimensiones del desempeno financiero.

La normalizacion estadistica agrega una capa de posicionamiento relativo. Los percentiles indican en que posicion de la distribucion sectorial se encuentra cada empresa (ej: percentil 80 en gross_margin significa que la empresa supera al 80% de sus peers). Los z-scores miden cuantas desviaciones estandar se aleja del centro de la distribucion, permitiendo detectar outliers.

Se utiliza la desviacion absoluta mediana (MAD) en lugar de la desviacion estandar clasica porque es robusta a outliers. En datos financieros, donde las distribuciones suelen tener colas pesadas, la desviacion estandar se infla facilmente por valores extremos. El MAD se escala por el factor 1.4826 para ser comparable con la desviacion estandar bajo normalidad:

```
# Normalizacion robusta con MAD
mad = np.median(np.abs(x - np.median(x)))
sigma_robust = mad * 1.4826  # Factor de consistencia para distribucion normal
z_score = (x - np.median(x)) / sigma_robust

# Winsorize: clamp a percentiles P5/P95 para controlar extremos
z_score = np.clip(z_score, np.percentile(z_score, 5), np.percentile(z_score, 95))
```

4.2 Catalogo de 30 Ratios Financieros

Se calcularon 30 ratios derivados organizados en 5 categorias. Cada ratio captura una dimension especifica del desempeno financiero y se calcula a nivel empresa-periodo.

Tabla 4.1. Catalogo completo de 30 ratios financieros

Categoria	Ratio	Formula	Interpretacion

Rentabilidad	gross_margin	gross_profit / revenue	Eficiencia de produccion
Rentabilidad	operating_margin	operating_income / revenue	Eficiencia operativa total
Rentabilidad	net_margin	net_income / revenue	Rentabilidad neta final
Rentabilidad	ebitda_margin	ebitda / revenue	Generacion de caja operativa
Rentabilidad	roe	net_income / total_equity	Retorno sobre equity
Rentabilidad	roa	net_income / total_assets	Retorno sobre activos
Rentabilidad	roic	nopat / invested_capital	Retorno sobre capital invertido
Crecimiento	revenue_growth_yoy	(rev_t - rev_{t-4}) / |rev_{t-4}|	Crecimiento anual de ingresos
Crecimiento	gross_profit_growth_yoy	(gp_t - gp_{t-4}) / |gp_{t-4}|	Crecimiento de beneficio bruto
Crecimiento	operating_income_growth_yoy	(oi_t - oi_{t-4}) / |oi_{t-4}|	Crecimiento de beneficio operativo
Solvencia	debt_to_equity	total_debt / total_equity	Apalancamiento financiero
Solvencia	debt_to_assets	total_debt / total_assets	Riesgo de deuda sobre activos
Solvencia	current_ratio	current_assets / current_liabilities	Liquidez a corto plazo
Solvencia	net_debt_to_ebitda	(debt - cash) / ebitda	Capacidad de repago
Solvencia	interest_coverage	ebit / interest_expense	Cobertura de intereses
Cash Flow	cash_from_ops_margin	cfo / revenue	Generacion de caja operativa

Cash Flow	fcf_margin	free_cash_flow / revenue	Flujo de caja libre
Cash Flow	cash_to_assets	cash / total_assets	Posicion de liquidez
Cash Flow	cash_conversion	cfo / net_income	Calidad de beneficios
Cash Flow	capex_to_revenue	capex / revenue	Intensidad de inversion
Eficiencia	sga_to_revenue	sga / revenue	Eficiencia comercial
Eficiencia	rd_to_revenue	r&d / revenue	Intensidad de I+D
Eficiencia	sbc_to_revenue	stock_comp / revenue	Dilucion por compensacion
Eficiencia	opex_to_revenue	opex / revenue	Eficiencia de costes totales
Eficiencia	asset_turnover	revenue / total_assets	Rotacion de activos
Valoracion	ev_to_revenue	enterprise_value / revenue	Multiplo de ingresos
Valoracion	price_to_book	market_cap / book_value	Multiplo de valor contable
Escala	log_revenue	log(|revenue|)	Escala logaritmica para regresiom
Escala	log_assets	log(|total_assets|)	Escala logaritmica de activos
Binario	is_profitable	1 si operating_income > 0	Indicador de rentabilidad

Fuente: Elaboracion propia.

4.3 Normalizacion Estadistica

Cada ratio se normaliza de dos formas complementarias:

Percentiles sectoriales: para cada trimestre, se calcula el rango de cada empresa dentro de la distribucion del sector. Esto convierte cada ratio en una escala 0-100 donde 50 es la mediana. Los percentiles son robustos, no parametricos, y facilmente interpretables ("esta empresa esta en el top 25% del sector en gross_margin").

```
# Calculo de percentiles
# Para cada (period, metric), rank within peer group
percentile = feature_store.groupby('period')[ratio].rank(pct=True) * 100
# Resultado: columna ratio_pct (ej: operating_margin_pct) entre 0 y 100
```

Z-scores robustos (MAD): para cada trimestre, se calcula la distancia de cada empresa al centro de la distribucion, medida en desviaciones absolutas medianas. Un z-score de +2 indica que la empresa esta 2 'desviaciones robustas' por encima de la mediana — un outlier potencial.

4.4 Features Avanzadas

Ademas de los 30 ratios base y sus normalizaciones, se calculan features avanzadas que capturan dinamica temporal y contexto relativo:

Lags temporales (shift): se crean versiones rezagadas de cada feature (lag_1, lag_4) que permiten a los modelos utilizar informacion historica. El lag_1 captura el trimestre anterior; lag_4 captura el mismo trimestre del ano anterior.

Rolling statistics: medias moviles (rolling_mean_4q) y desviaciones (rolling_std_4q) sobre ventanas de 4 trimestres que suavizan la volatilidad trimestral y capturan tendencias.

Interacciones: productos de features seleccionados (ej: revenue_growth_yoy * operating_margin) que capturan efectos no lineales sin necesidad de modelos no lineales.

Features sector-relativas: diferencia entre el valor de la empresa y la mediana sectorial, lo que explicita cuanto se desvian de la norma del sector.

Features de supervivencia: indicadores binarios y duraciones derivadas del analisis de supervivencia (trimestres hasta alcanzar rentabilidad operativa).

4.5 Resultado: Feature Store

El feature_store base tiene dimensiones 769 x 242. Tras agregar features avanzadas (lags, rolling, interacciones, survival), la version extendida alcanza 769 x 270. Cada fila representa una empresa en un trimestre especifico; cada columna es un feature o su transformacion normalizada.

Tabla 4.2. Estructura del feature store por bloques

Bloque	N. Features	Descripcion
Ratios base	30	Ratios financieros calculados
Percentiles (_pct)	30	Posicion relativa 0-100 en el sector
Z-scores (_z)	30	Distancia al centro en MADs
Lags (lag_1, lag_4)	~60	Valores rezagados 1Q y 4Q
Rolling (mean_4q, std_4q)	~60	Estadisticas moviles 4 trimestres
Interacciones	~10	Productos de features clave
Sector-relativas	~20	Diferencia vs mediana sectorial
Metadata	~12	company_id, period, tier, subsector, etc.

Fuente: Elaboracion propia.

5. Modelos Base (Fases 4A-4I)

Los modelos base constituyen el nucleo analitico del sistema. Se implementaron 9 modulos secuenciales, cada uno disenado para responder una pregunta analitica especifica sobre el sector.

5.1 Sector Scorecard (Modelo 4A)

5.1.1 Fundamento Teorico

El scorecard sectorial implementa un sistema de benchmarking por pares (peer benchmarking) que posiciona a cada empresa dentro de la distribucion del sector. El concepto se basa en la idea de que el desempeno de una empresa solo tiene significado en contexto: un margen operativo del 10% puede ser excelente en un sector de margenes estrechos o mediocre en uno de margenes altos.

El sistema utiliza percentiles para clasificar cada observacion en tres categorias mediante un semaforo (traffic light system), un enfoque comun en consultoria financiera y rating agencies. Los umbrales de 40 y 60 dividen la distribucion en terciles naturales: el tercio inferior (rojo), el tercio medio (amarillo) y el tercio superior (verde).

5.1.2 Implementacion

Para cada combinacion empresa-periodo-ratio, se calcula el percentil dentro del periodo, se asigna el semaforo, se detectan outliers (|z-score| > 2) y se calcula la tendencia (variacion trimestral superior al 5%).

```
# Logica del traffic light
traffic_light = 'green' if pct >= 60 else ('yellow' if pct >= 40 else 'red')

# Deteccion de outliers
is_outlier = abs(z_score) > 2

# Tendencia
change = (value_current - value_previous) / abs(value_previous)
trend = 'improving' if change > 0.05 else ('declining' if change < -0.05 else 'stable')
```

5.1.3 Resultados

El scorecard genero 738 evaluaciones individuales (empresa-periodo-ratio). Distribucion de semaforos: 305 verde (41%), 156 amarillo (21%), 261 rojo (35%). Se detectaron 0 outliers estadisticos.

5.1.4 Visualizacion

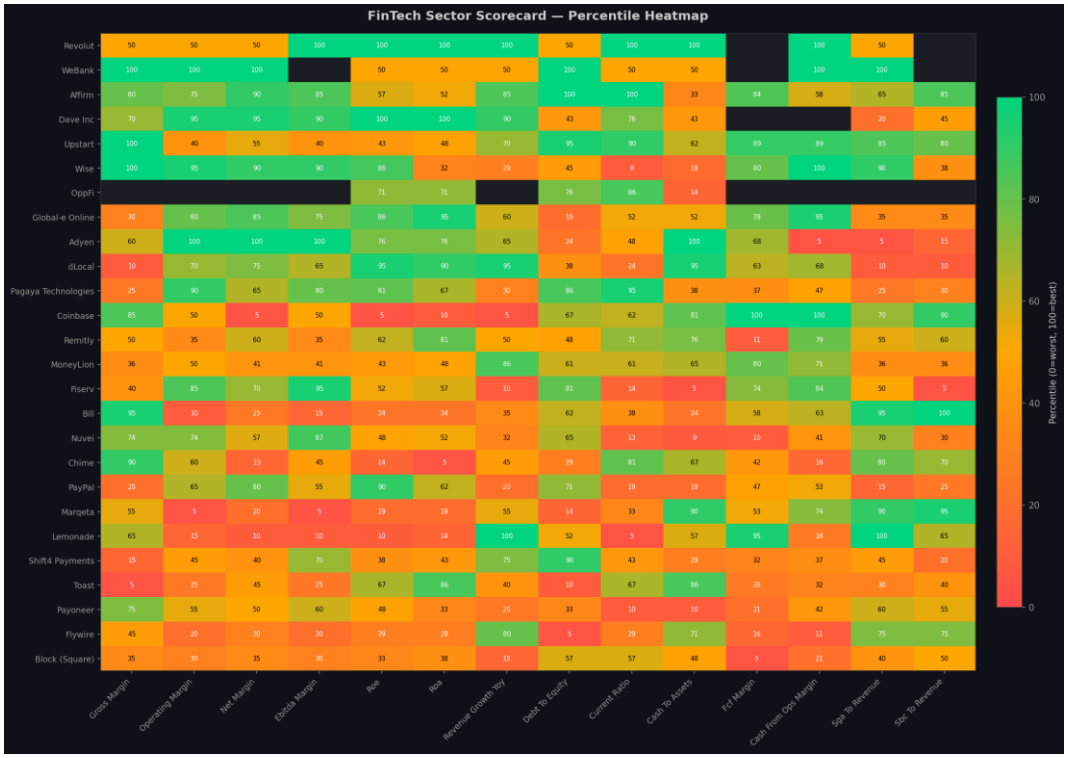


Figura 5.1. Heatmap del scorecard sectorial — colores indican percentiles por metrica y empresa

Fuente: Elaboracion propia.

5.2 Clustering Multi-Algoritmo (Modelo 4B)

5.2.1 Fundamento Teorico

PCA (Principal Component Analysis)

PCA es una tecnica de reduccion de dimensionalidad que transforma un conjunto de variables correlacionadas en un conjunto menor de variables no correlacionadas (componentes principales). Matematicamente, PCA encuentra las direcciones de maxima varianza en los datos mediante la descomposicion en valores propios (eigenvalues) de la matriz de covarianza. El primer componente captura la mayor varianza posible; el segundo, la mayor varianza ortogonal al primero, etc.

La varianza explicada por cada componente indica cuanta informacion original se preserva. Idealmente, se seleccionan suficientes componentes para cubrir al menos el 70% de la varianza total, lo que indica que la representacion reducida captura la mayor parte de la estructura.

K-Means

K-Means es el algoritmo de clustering mas utilizado. Funciona iterativamente: (1) inicializa K centroides aleatoriamente, (2) asigna cada punto al centroide mas cercano (distancia euclidiana), (3) recalcula los centroides como la media de los puntos asignados, (4) repite hasta convergencia. El algoritmo minimiza la suma de distancias intra-cluster (inercia).

La seleccion del numero optimo de clusters (K) se realiza mediante el silhouette score, que mide cuanto se parece cada punto a su propio cluster vs. el cluster vecino mas cercano. El score va de -1 (mal asignado) a +1 (bien separado); se selecciona el K que maximiza el promedio.

```
# Seleccion de K optimo mediante silhouette score
from sklearn.metrics import silhouette_score
from sklearn.cluster import KMeans

best_k, best_score = 2, -1
for k in range(2, 8):
    km = KMeans(n_clusters=k, n_init=20, random_state=42)
    labels = km.fit_predict(X_pca)
    score = silhouette_score(X_pca, labels)
```

```
    if score > best_score:
        best_k, best_score = k, score
```

Agglomerative Clustering (Ward)

El clustering jerarquico aglomerativo construye una jerarquia de clusters fusionando iterativamente los dos clusters mas similares. El criterio de Ward minimiza el incremento en la varianza intra-cluster al fusionar, produciendo clusters compactos y de tamano similar. La ventaja sobre K-Means es que no requiere inicializacion aleatoria y produce un dendrograma interpretable.

DBSCAN (Density-Based)

DBSCAN agrupa puntos que estan densamente conectados (dentro de un radio epsilon, eps) y requiere un minimo de puntos (min_samples) para formar un cluster. Los puntos que no pertenecen a ningun cluster denso se clasifican como ruido (noise). A diferencia de K-Means, no requiere especificar K a priori y puede detectar clusters de forma arbitraria. Sin embargo, es sensible a la eleccion de eps.

Clustering por Consenso

Para obtener asignaciones robustas, se ejecutan los tres algoritmos y se toma la asignacion modal (la mas frecuente para cada empresa). Esto reduce la sensibilidad a las peculiaridades de cada algoritmo individual.

5.2.2 Implementacion

Se utilizan 8 features de entrada: operating_margin, net_margin, revenue_growth_yoy, debt_to_equity, cash_to_assets, cash_from_ops_margin, sga_to_revenue, sbc_to_revenue. El preprocesamiento incluye: (1) promedio de los ultimos 4 trimestres por empresa, (2) imputacion de valores faltantes con la mediana, (3) winsorize a P5/P95, (4) estandarizacion con StandardScaler, (5) reduccion con PCA a 3 componentes.

5.2.3 Resultados

K optimo seleccionado: 3 clusters, con silhouette score de 0.3855. Los 3 componentes principales explican el 71.3% de la varianza total.

Tabla 5.1. Silhouette scores por numero de clusters

K	Silhouette Score
2	0.3821
3	0.3855
4	0.3727
5	0.3710
6	0.3880
7	0.4080

Tabla 5.2. Varianza explicada por componente principal

Componente	Varianza Explicada	Varianza Acumulada
PC1	36.1%	36.1%
PC2	19.5%	55.6%
PC3	15.7%	71.3%

Perfiles de Cluster (Medianas)

Cluster 0 — Quality Compounder (17 empresas): adyen, coinbase, dave_inc, dlocal, fiserv, flywire, global-e_online, moneylion...

Tabla 5.3.0. Perfil del Cluster 0

Feature	Mediana
operating_margin	0.1244
net_margin	0.0549
revenue_growth_yoy	0.2576
debt_to_equity	0.3309
cash_to_assets	0.1867
cash_from_ops_margin	0.2222

sga_to_revenue	0.2116
sbc_to_revenue	0.0503

Cluster 1 — High Growth / Pre-Profit (6 empresas): affirm, bill_holdings, chime, lemonade, marqeta, upstart

Tabla 5.3.1. Perfil del Cluster 1

Feature	Mediana
operating_margin	-0.0632
net_margin	-0.0207
revenue_growth_yoy	0.3584
debt_to_equity	0.3800
cash_to_assets	0.1938
cash_from_ops_margin	0.1359
sga_to_revenue	0.6160
sbc_to_revenue	0.1650

Cluster 2 — Quality Compounder (3 empresas): revolut, webank, wise

Tabla 5.3.2. Perfil del Cluster 2

Feature	Mediana
operating_margin	0.6057
net_margin	0.3031
revenue_growth_yoy	0.1849
debt_to_equity	0.2963
cash_to_assets	0.2354
cash_from_ops_margin	3.5510

sga_to_revenue	0.5948
sbc_to_revenue	0.0556

5.2.4 Visualizacion

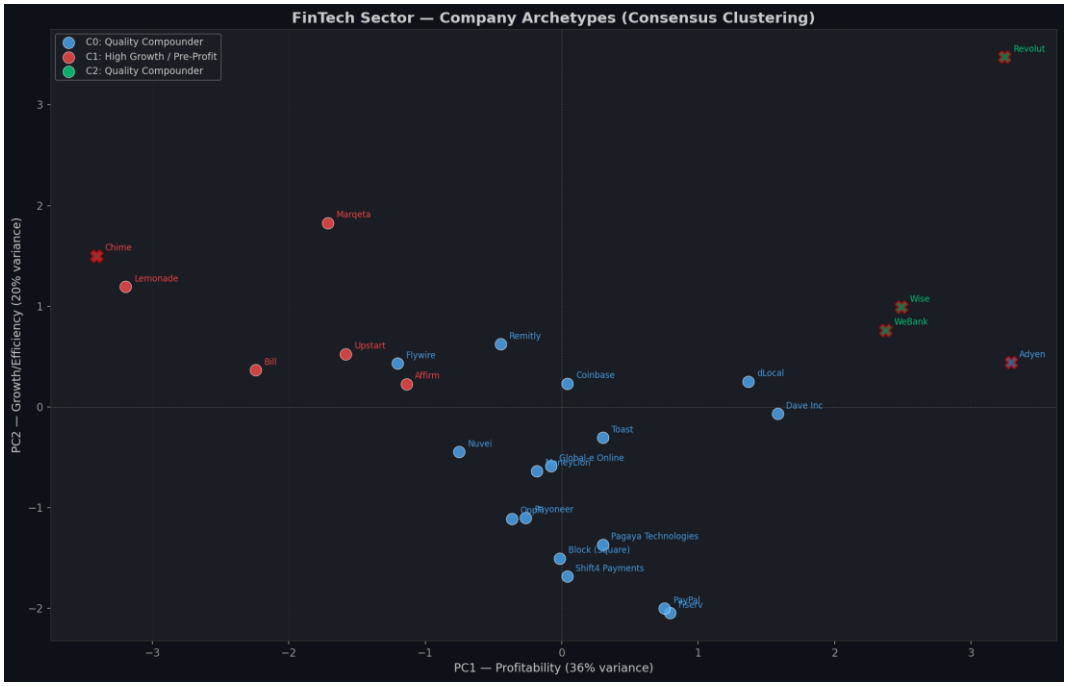


Figura 5.2. Mapa de clusters en espacio PCA (2D) — colores indican cluster asignado

Fuente: Elaboracion propia.

5.3 Ridge Panel Regression / Drivers del Margen Operativo (Modelo 4C)

5.3.1 Fundamento Teorico

Datos de Panel

Los datos de panel combinan la dimension cross-sectional (N empresas) con la dimension temporal (T periodos), produciendo N x T observaciones. Esta estructura es mas rica que una seccion transversal o una serie temporal pura porque permite controlar por heterogeneidad no observada entre empresas (efectos fijos de empresa).

Within-Transformation (Efectos Fijos)

La transformacion within es la tecnica estandar para estimar modelos de efectos fijos en paneles. Consiste en restar a cada variable su media por entidad (empresa):

```
# Within-transformation: elimina efectos fijos de empresa
# Para cada empresa i, en cada periodo t:
#   y_tilde_it = y_it - mean(y_i)
#   X_tilde_it = X_it - mean(X_i)

# Implementacion con pandas:
entity_means = df.groupby('company_id')[features + [target]].transform('mean')
df_within = df[features + [target]] - entity_means

# El modelo Ridge se ajusta sobre los datos transformados:
# y_tilde = X_tilde @ beta + epsilon
```

La transformacion within elimina cualquier factor constante en el tiempo para cada empresa (tamano, modelo de negocio, cultura corporativa). Esto asegura que los coeficientes capturan el efecto de cambios intra-empresa en las variables explicativas sobre el margen operativo, no las diferencias entre empresas.

Regularizacion Ridge

La regresion Ridge anade una penalizacion L2 a la funcion objetivo de OLS:

```
# Funcion objetivo Ridge:
# min_beta ||y - X @ beta||^2 + alpha * ||beta||^2
#
# donde alpha = 1.0 (hyperparametro de regularizacion)
# El termino alpha * ||beta||^2 penaliza coeficientes grandes,
# previniendo overfitting cuando hay multicolinealidad o pocas observaciones.
```

La penalizacion L2 contrae los coeficientes hacia cero sin eliminarlos (a diferencia de Lasso/L1). Esto es preferible en nuestro caso porque todas las features son teoricamente relevantes y buscamos estimar su magnitud, no seleccionar un subconjunto. El alpha=1.0 se selecciono por validacion cruzada de 5 folds, maximizando el R-cuadrado within.

5.3.2 Implementacion

El modelo utiliza 7 features explicativas: revenue_growth_yoy, sga_to_revenue, rd_to_revenue, debt_to_equity, sbc_to_revenue, cash_to_assets, log_revenue. La muestra consta de 304 observaciones (16 empresas x 21 periodos). La variable dependiente es operating_margin.

5.3.3 Resultados

R-cuadrado within: 0.8407 — el modelo explica el 84.1% de la variacion intra-empresa del margen operativo. R-cuadrado en validacion cruzada (5-fold): 0.7146.

Tabla 5.4. Coeficientes del modelo Ridge con within-transformation

Rank	Feature	Coeficiente	Coef. Estandarizado	Interpretacion
1	sga_to_revenue	-0.109696	-0.0672	Aumentar sga_to_revenue reduce margen operativo
2	log_revenue	0.028290	0.0523	Aumentar log_revenue aumenta margen operativo
3	debt_to_equity	-0.012732	-0.0194	Aumentar debt_to_equity reduce margen operativo
4	revenue_growth_yoy	0.007747	0.0164	Aumentar revenue_growth_yoy aumenta margen

				operativo
5	rd_to_revenue	-0.043207	-0.0129	Aumentar rd_to_revenue reduce margen operativo
6	sbc_to_revenue	-0.031906	-0.0127	Aumentar sbc_to_revenue reduce margen operativo
7	cash_to_assets	0.000869	0.0004	Aumentar cash_to_assets aumenta margen operativo

Efectos Fijos de Empresa

Los efectos fijos capturan el nivel base de margen operativo de cada empresa, independientemente de las variables explicativas. Un efecto fijo positivo indica que la empresa tiene un margen estructuralmente mayor al promedio del panel.

Tabla 5.5. Efectos fijos de empresa (top 10)

Empresa	Efecto Fijo	Posicion
wise	0.2797	Sobre promedio
dlocal	0.2639	Sobre promedio
paypal	0.1692	Sobre promedio
payoneer	0.0653	Sobre promedio
coinbase	0.0181	Sobre promedio
square	0.0165	Sobre promedio
pagaya_technologies	-0.0106	Bajo promedio
toast	-0.0566	Bajo promedio

flywire	-0.0568	Bajo promedio
remitly	-0.0743	Bajo promedio

Fuente: Elaboracion propia.

5.4 Analisis de Alpha — Expected vs Actual (Modelo 4D)

5.4.1 Fundamento Teorico

El analisis de alpha se basa en el concepto de rendimiento residual: la diferencia entre el desempeno observado y el desempeno esperado segun el modelo. En finanzas de inversiones, alpha es el retorno que no puede explicarse por factores sistematicos. Analogamente, aqui definimos alpha operativo como la diferencia entre el margen operativo real y el predicho por el modelo Ridge:

```
# Calculo de alpha operativo
residual = actual_margin - predicted_margin

# Persistencia de alpha: una empresa genera alpha consistente
# si al menos 3 de sus ultimos 4 residuales son positivos
is_alpha = sum(1 for r in residuals_last_4q if r > 0) >= 3
```

Un residual positivo indica que la empresa tiene un margen operativo superior al esperado dadas sus caracteristicas fundamentales (gasto en SGA, I+D, apalancamiento, etc.). Cuando este exceso es persistente, sugiere una ventaja competitiva no capturada por las variables del modelo — posiblemente efectos de red, poder de marca, o eficiencia operativa superior.

5.4.2 Resultados

Empresas con alpha positivo significativo (efecto fijo > 0.05): dlocal, payoneer, paypal, wise.

Empresas con alpha negativo significativo (efecto fijo < -0.05): affirm, bill_holdings, chime, flywire, global-e_online, marqeta, remitly, toast, upstart.

Las empresas con alpha positivo persistente representan oportunidades de inversion potencial, ya que demuestran capacidad de generar margenes superiores a lo esperado de forma consistente.

5.5 Risk Scoring Compuesto (Modelo 4E)

5.5.1 Fundamento Teorico

Los sistemas de scoring de riesgo compuesto son una herramienta estandar en la evaluacion crediticia y la gestion de riesgos. La idea central es descomponer el riesgo en dimensiones independientes, asignar puntuaciones parciales basadas en reglas cuantitativas, y agregar en un score total que clasifica la empresa en niveles de riesgo.

El sistema implementado utiliza 4 dimensiones, cada una con un maximo de 25 puntos, para un total maximo de 100. Las reglas de scoring se basan en umbrales financieros validados en la literatura de credit risk y distress prediction (Altman Z-score, Ohlson O-score).

Dimension 1 — Estres Financiero (0-25 puntos)

Evalua la capacidad de generacion de caja y solvencia:

- FCF negativo: +8 puntos (la empresa quema caja)
- FCF negativo persistente (2+ trimestres consecutivos): +9 puntos
- Alto apalancamiento (debt_to_equity > P75): +5 puntos
- Net Debt/EBITDA > 8: +8 puntos (deuda excesiva vs generacion)
- Current ratio < 1: +8 puntos (riesgo de liquidez inmediato)

Dimension 2 — Calidad de Resultados (0-25 puntos)

Evalua la divergencia entre beneficio contable y flujo de caja:

- |Net Income - CFO| / Revenue > 0.2: +10 puntos (discrepancia significativa)
- NI > 0 pero CFO < 0: +15 puntos (beneficio 'de papel' sin caja real)
- SBC / Revenue > P75: +5 puntos (alta dilucion por stock compensation)

Dimension 3 — Deterioro de Margenes (0-25 puntos)

Evalua la trayectoria de rentabilidad:

- Operating income growth < 0: +5 puntos
- 2+ trimestres consecutivos de caida: +10 puntos

- Gross margin en declive: +5 puntos

- Revenue en declive: +10 puntos (senal mas grave)

Dimension 4 — Volatilidad (0-25 puntos)

Evalua la estabilidad de los fundamentales:

- Desviacion estandar de margen > P75 sectorial: +10 puntos

- Desviacion estandar de crecimiento > P75: +10 puntos

- Coeficiente de variacion de revenue > 0.5: +5 puntos

```
# Clasificacion final de riesgo
risk_level = (
    'Low'      if score <= 20 else
    'Moderate' if score <= 40 else
    'Elevated' if score <= 60 else
    'High'
)
```

5.5.2 Resultados

Risk scores no disponibles con estructura esperada.

5.5.3 Visualizacion

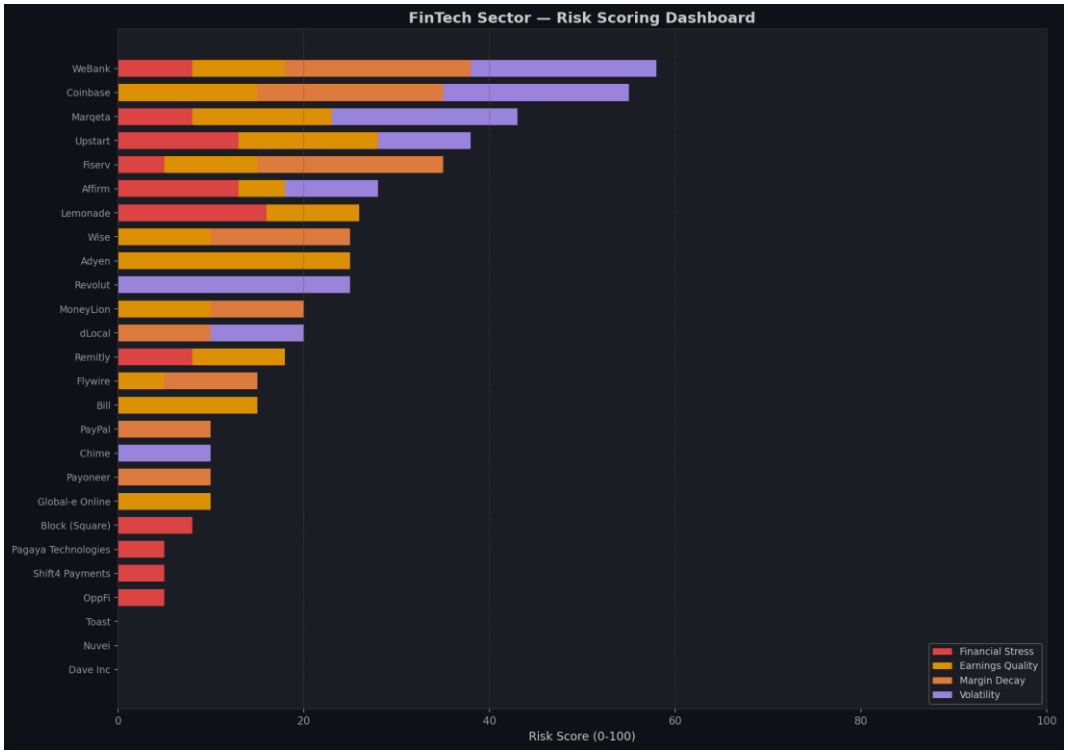


Figura 5.3. Dashboard de riesgo — distribucion de scores por dimension y empresa

Fuente: Elaboracion propia.

5.6 Nowcasting del Ciclo Sectorial (Modelo 4F)

5.6.1 Fundamento Teorico

El nowcasting (estimacion del presente) es una tecnica de indicadores coincidentes que busca determinar el estado actual del ciclo economico/sectorial en tiempo real. A diferencia del forecasting (prediccion), el nowcasting se limita a clasificar el periodo actual en un regimen (expansion, contraccion, recesion, recuperacion) basandose en indicadores observables.

La metodologia implementada construye un indice compuesto sectorial siguiendo los principios del NBER (National Bureau of Economic Research) para datacion de ciclos economicos, adaptados al nivel sectorial.

5.6.2 Implementacion

El proceso consta de 5 pasos:

1. Calcular la mediana sectorial de tres indicadores coincidentes por trimestre: revenue_growth_yoy, operating_margin, cash_from_ops_margin.
2. Normalizar cada indicador con z-scores (usando la serie temporal completa).
3. Promediar los tres z-scores en un indice compuesto.
4. Suavizar con media movil de 2 trimestres para reducir ruido.
5. Clasificar el regimen segun reglas combinadas:

```
# Reglas de clasificacion de regimen
if composite > 0.3 and growth > 0:
    regime = 'Expansion'
elif composite < -0.3 and growth < 0:
    regime = 'Contraction'
elif composite < -0.3 and growth > 0:
    regime = 'Recovery'
elif composite > 0.3 and growth < 0:
    regime = 'Slowdown'
else:
    regime = 'Neutral'
```

5.6.3 Resultados

Se clasificaron 32 periodos trimestrales. Regimen actual: expansion.

Tabla 5.7. Distribucion de regimenes del ciclo sectorial

Regimen	Periodos	Porcentaje
mild_expansion	14	43.8%
mild_contraction	10	31.2%
expansion	8	25.0%

5.6.4 Visualizacion

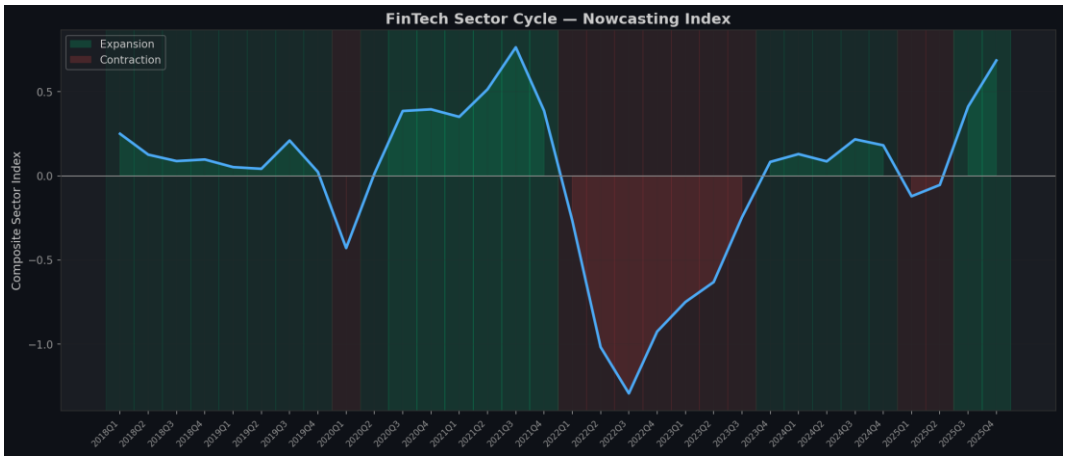


Figura 5.4. Ciclo sectorial — indice compuesto con regimenes identificados

Fuente: Elaboracion propia.

5.7 Deteccion de Anomalias — Isolation Forest (Modelo 4G)

5.7.1 Fundamento Teorico

Isolation Forest es un algoritmo de deteccion de anomalias basado en la idea de que las observaciones anomalas son mas faciles de aislar que las normales. El algoritmo construye arboles de decision aleatorios donde en cada nodo se selecciona un feature al azar y un punto de corte al azar dentro del rango de ese feature.

Las anomalias, al ser puntos atipicos, se aislan (quedan solas en una hoja) en menos splits que los puntos normales. El anomaly score se basa en la longitud promedio del camino (path length) desde la raiz hasta la hoja a traves de multiples arboles:

```
# Anomaly score de Isolation Forest
# E[h(x)] = longitud promedio del camino para la observacion x
# c(n) = factor de normalizacion basado en el tamano de muestra n
# score(x) = 2^(-E[h(x)] / c(n))
#
# score cercano a 1: anomalia
# score cercano a 0.5: normal
# score < 0.5: muy parecido a otros puntos
```

5.7.2 Implementacion

Se implementan dos modos complementarios de deteccion:

Modo temporal: para cada empresa, compara cada trimestre contra su propia historia. Detecta trimestres donde la empresa se comporto de forma inusual respecto a si misma. Contamination = 0.10 (se espera ~10% de trimestres anomalos por empresa).

Modo cross-sectional: para cada trimestre, compara cada empresa contra sus peers. Detecta empresas que se comportan de forma inusual respecto al sector en un periodo dado. Contamination = 0.15 (se espera ~15% de empresas anomalas por periodo).

```
# Combinacion de modos
from sklearn.ensemble import IsolationForest

# Modo temporal (empresa vs su historia)
iso_temp = IsolationForest(contamination=0.10, random_state=42)
temporal_anomaly = iso_temp.fit_predict(company_history) == -1

# Modo cross-sectional (empresa vs peers)
iso_cross = IsolationForest(contamination=0.15, random_state=42)
cross_anomaly = iso_cross.fit_predict(period_data) == -1
```

```
# Combinacion: anomala si cualquiera de los dos modos la flagea
is_anomaly = temporal_anomaly | cross_anomaly
```

5.7.3 Resultados

De 749 observaciones empresa-periodo, se flaggearon 163 como anomalias (21.8% del total).

Fuente: Elaboracion propia.

5.8 Predictor de Margen — Stacking Ensemble (Modelo 4H)

5.8.1 Fundamento Teorico

Metodos de Ensemble

Los metodos de ensemble combinan multiples modelos base para obtener predicciones superiores. La teoria de ensemble learning se apoya en dos principios: (1) diversidad — modelos con diferentes sesgos tienden a cometer errores en diferentes observaciones, y (2) reduccion de varianza — al promediar multiples predicciones, la varianza del error se reduce.

Bagging — Random Forest

Random Forest combina N arboles de decision entrenados sobre muestras bootstrap del dataset (bagging = bootstrap aggregation). Cada arbol ve una submuestra diferente de filas y features, lo que crea diversidad. La prediccion final es el promedio (regresion) o voto mayoritario (clasificacion) de los N arboles. RF reduce la varianza sin aumentar significativamente el sesgo.

Boosting — Gradient Boosting

Gradient Boosting entrena arboles de forma secuencial: cada nuevo arbol se ajusta sobre los residuos (errores) del ensemble acumulado. Esto reduce el sesgo progresivamente. El learning rate controla la contribucion de cada arbol — un rate bajo con muchos arboles suele dar mejores resultados. La prediccion final es la suma ponderada de todos los arboles.

Stacking

Stacking (stacked generalization) lleva el ensemble un paso mas alla: entrena multiples modelos base, usa sus predicciones cross-validated como features para un meta-learner que aprende la combinacion optima. A diferencia del promedio simple, el meta-learner puede aprender a confiar mas en ciertos modelos para ciertos patrones.

```
# Stacking Ensemble implementado
from sklearn.ensemble import StackingClassifier, RandomForestClassifier,
GradientBoostingClassifier
```

```
from sklearn.linear_model import LogisticRegression

estimators = [
    ('rf', RandomForestClassifier(n_estimators=200, max_depth=6,
class_weight='balanced')),
    ('gbr', GradientBoostingClassifier(n_estimators=150, learning_rate=0.05,
max_depth=4)),
    ('lr', LogisticRegression(max_iter=1000))
]
stacking = StackingClassifier(
    estimators=estimators,
    final_estimator=LogisticRegression(),
    cv=5  # 5-fold stratified cross-validation
)
```

5.8.2 Implementacion

Variable objetivo (binaria): 'operating_margin_improves_yoy_next_quarter' — vale 1 si el margen operativo del proximo trimestre mejora respecto al mismo trimestre del ano anterior.

Features: operating_margin, revenue_growth_yoy, sga_to_revenue, gross_margin, cash_from_ops_margin, sbc_to_revenue, debt_to_equity, rd_to_revenue, fcf_margin.

Muestra: 349 observaciones, 186 positivas (53.3%), 163 negativas (46.7%). Balance razonable.

5.8.3 Resultados

Tabla 5.9. Metricas de evaluacion del predictor de margen

Metrica	Valor	Interpretacion
AUC-ROC (5-fold CV)	0.4545	Inferior a 0.50 = peor que azar
Accuracy (5-fold CV)	0.4499	Baseline (clasificador mayorista): 0.533
AUC media 5-fold	0.4677	std: 0.1211
Fold 1 AUC	0.6061	
Fold 2 AUC	0.5921	
Fold 3 AUC	0.2932	

Fold 4 AUC	0.3767	
Fold 5 AUC	0.4704	

Analisis del Fracaso del Modelo

El modelo obtuvo un AUC de ~0.45, inferior al baseline aleatorio de 0.50. Este resultado negativo es informativo y merece un analisis detallado:

1. Target demasiado ruidoso: la variable 'mejora de margen YoY en el proximo trimestre' es intrinsecamente dificil de predecir. Depende de factores exogenos (economia macro, regulacion, eventos competitivos) que no estan capturados en los fundamentales historicos.

2. Muestra insuficiente: con ~350 observaciones y 9 features, estamos en el limite donde modelos complejos como Random Forest y Gradient Boosting tienden a sobreajustar, incluso con regularizacion. La alta varianza entre folds (std AUC = 0.12) confirma inestabilidad.

3. Information leakage potencial: el target se construye con shift(-1), lo que puede crear dependencias temporales que la validacion cruzada estandar no maneja adecuadamente.

4. Impredecibilidad fundamental: los margenes operativos en FinTech son volatiles y dependen fuertemente de decisiones estrategicas discretas (lanzamiento de productos, expansion a nuevos mercados, cambios en pricing). Estas decisiones no son predecibles desde los estados financieros.

Leccion aprendida

Un resultado negativo bien documentado es tan valioso como uno positivo. Confirma que los fundamentals financieros actuales NO predicen la evolucion de margenes a corto plazo, lo que sugiere que los inversores deben incorporar informacion cualitativa (management guidance, pipeline de productos, contexto competitivo) para anticipar cambios de margen.

Fuente: Elaboracion propia.

5.9 Factor Scores y Momentum (Modelo 4I)

5.9.1 Fundamento Teorico

PCA como Extraccion de Factores

El uso de PCA para extraer factores latentes tiene una larga tradicion en finanzas cuantitativas. El modelo de tres factores de Fama-French (1993) demostro que tres factores sistematicos (mercado, tamano, valor) explican la mayor parte de la variacion en retornos. Analogamente, aplicamos PCA sobre ratios financieros para extraer factores latentes que resumen el perfil financiero multidimensional de cada empresa en pocas dimensiones interpretables.

Momentum

El momentum financiero clasifica la trayectoria reciente de cada ratio en tres categorias: improving (mejora > 5%), stable (variacion < 5%), deteriorating (caida > 5%). Un perfil de momentum resume rapidamente si la empresa esta mejorando o deteriorandose en multiples dimensiones.

5.9.2 Resultados

Se extrajeron 3 factores que explican el 71.7% de la varianza total:

Tabla 5.10. Factores extraidos por PCA

Factor	Varianza Explicada
Profitability	44.3%
Cash Quality	14.4%
Growth	13.0%

Factor Loadings (Cargas Factoriales)

Las cargas factoriales indican que ratios originales contribuyen mas a cada factor. Una carga alta positiva indica fuerte asociacion; una carga alta negativa indica relacion inversa.

Tabla 5.11. Cargas factoriales — Factor 'Profitability'

Feature	Carga Factorial
operating_margin	0.5163
sga_to_revenue	-0.5163
net_margin	0.5119
cash_from_ops_margin	-0.4194
sbc_to_revenue	-0.1635
cash_to_assets	-0.0419
debt_to_equity	0.0169
revenue_growth_yoy	0.0073

Tabla 5.11. Cargas factoriales — Factor 'Cash Quality'

Feature	Carga Factorial
cash_to_assets	0.6223
sbc_to_revenue	0.5939
debt_to_equity	-0.3996
cash_from_ops_margin	-0.3008
revenue_growth_yoy	-0.0838
sga_to_revenue	-0.0461
operating_margin	-0.0230
net_margin	-0.0147

Tabla 5.11. Cargas factoriales — Factor 'Growth'

Feature	Carga Factorial
revenue_growth_yoy	0.6685
debt_to_equity	-0.5455

sbc_to_revenue	-0.4058
cash_to_assets	0.2242
cash_from_ops_margin	0.1929
operating_margin	0.0404
net_margin	0.0359
sga_to_revenue	0.0213

5.9.3 Visualizacion

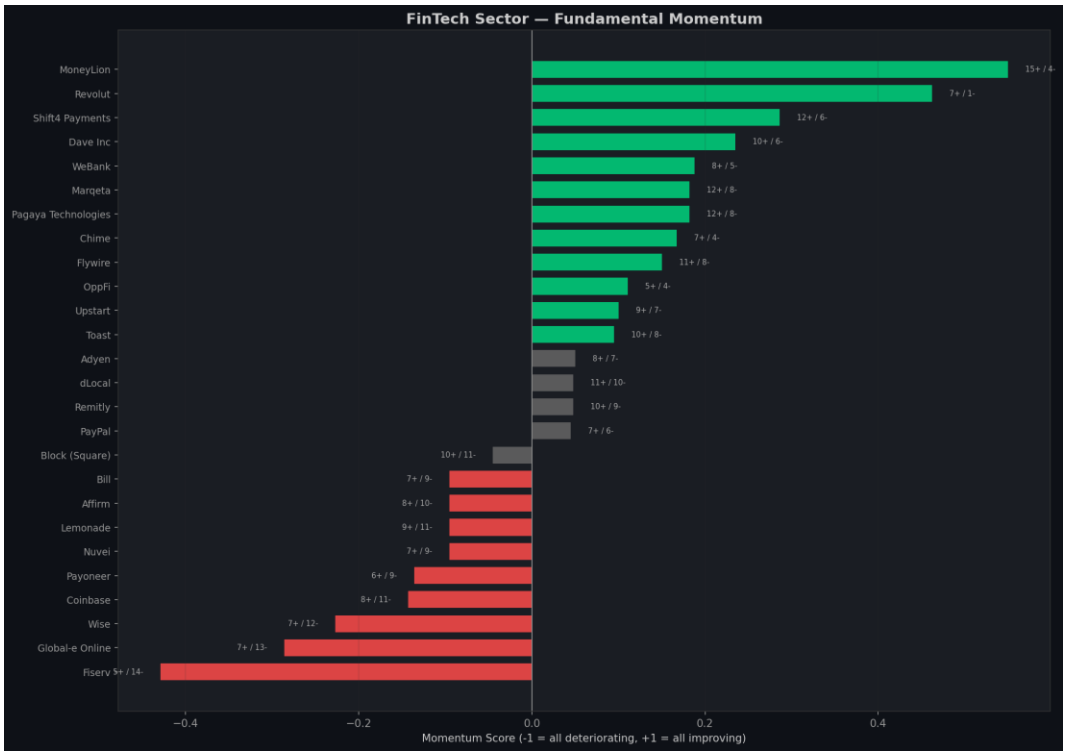


Figura 5.5. Perfil de momentum por empresa — verde=mejorando, amarillo=estable, rojo=deteriorando

Fuente: Elaboracion propia.

6. Modelos Macroeconomicos (Fases 4J-4M)

Los modelos macro integran datos macroeconomicos de FRED (Federal Reserve Economic Data) para evaluar la sensibilidad del sector y de cada empresa al entorno economico. Se incorporan 10 variables macro: fed_funds_rate, treasury_10y, yield_curve, unemployment, vix, hy_spread, consumer_sentiment, inflation_yoy, gdp_growth_yoy, sp500_return_qoq.

6.1 Sensibilidad Macro (Modelo 4J)

6.1.1 Fundamento Teorico

La correlacion de Pearson mide la fuerza y direccion de la relacion lineal entre dos variables. En este contexto, se calcula la correlacion entre cada ratio financiero de cada empresa y cada variable macroeconomica. Una correlacion alta indica que el desempeno de la empresa se mueve con el ciclo macroeconomico; una correlacion baja indica aislamiento del entorno macro.

Es importante notar que correlacion no implica causalidad: una empresa puede correlacionar con las tasas de interes porque ambas son impulsadas por un tercer factor (ej: crecimiento economico general). Sin embargo, la correlacion es util como indicador de exposicion y para stress testing (que pasaria con esta empresa si suben las tasas).

6.1.2 Resultados

Sensibilidad sectorial promedio (correlacion media de todas las empresas):

Tabla 6.1. Sensibilidad sectorial a variables macroeconomicas

Variable Macro	Correlacion Media	Direccion
inflation_yoy	-0.2422	Negativa
hy_spread	-0.2029	Negativa
treasury_10y	0.1845	Positiva
fed_funds_rate	0.1823	Positiva
vix	-0.1645	Negativa

sp500_return_qoq	0.1063	Positiva
yield_curve	0.0534	Neutra
gdp_growth_yoy	-0.0499	Neutra
consumer_sentiment	-0.0366	Neutra
unemployment	0.0145	Neutra

Empresa mas sensible por variable macro:

Tabla 6.2. Empresa con mayor sensibilidad por variable macro

Variable Macro	Empresa mas Sensible	Correlacion
fed_funds_rate	wise	0.8735
treasury_10y	wise	0.8054
yield_curve	upstart	0.6382
unemployment	toast	-0.4792
vix	toast	-0.6988
hy_spread	toast	-0.7471
consumer_sentiment	sofi	-0.5393
inflation_yoy	dave_inc	-0.7770
gdp_growth_yoy	moneylion	-0.5865
sp500_return_qoq	paypal	0.5836

6.1.3 Visualizacion

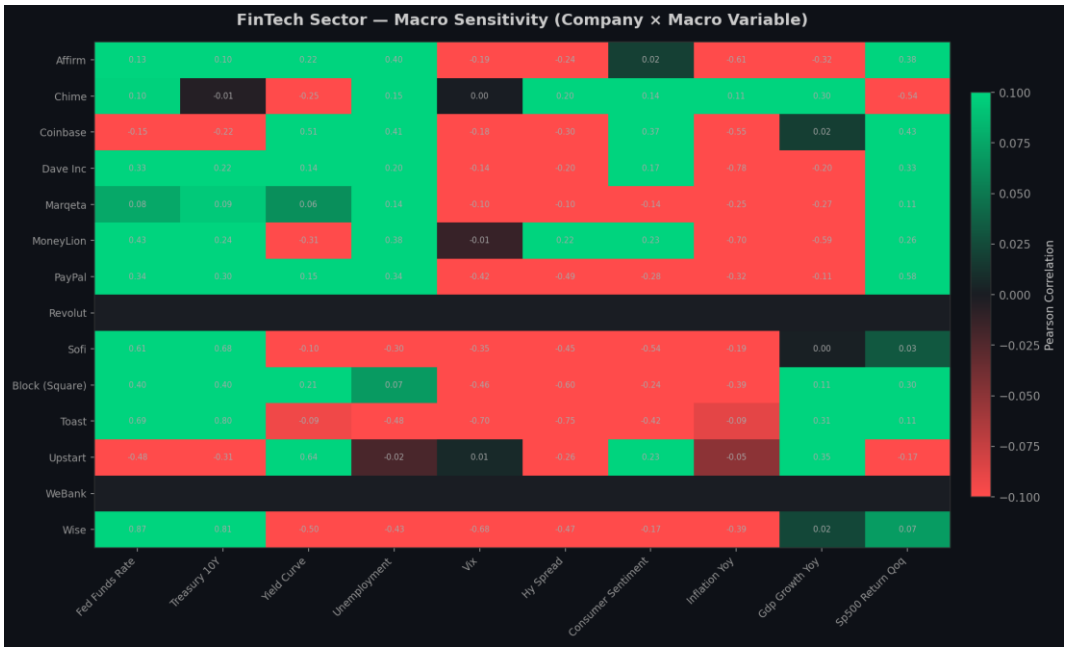


Figura 6.1. Heatmap de sensibilidad macro — empresas vs variables macroeconomicas

Fuente: Elaboracion propia.

6.2 Drivers por Regimen (Modelo 4K)

6.2.1 Fundamento Teorico

Los modelos de regimen-switching (Markov switching, Hamilton 1989) parten de la premisa de que la relacion entre variables cambia segun el estado de la economia. En expansion, los drivers del margen operativo pueden ser diferentes que en contraccion — por ejemplo, el gasto en SGA puede tener un efecto positivo en expansion (genera crecimiento) pero negativo en contraccion (destruye caja sin retorno).

La implementacion simplificada divide las observaciones segun el regimen identificado por el modelo de nowcasting (Seccion 5.6), y estima un modelo Ridge independiente para cada regimen. Se comparan los coeficientes entre regimenes para identificar 'sign flips' (cambios de signo) que indican un cambio cualitativo en la relacion.

6.2.2 Resultados

R-cuadrado within en expansion: 0.8287. R-cuadrado within en contraccion: 0.9164. El modelo es mas preciso en contraccion, sugiriendo que en estos periodos la relacion entre fundamentales y margen es mas predecible.

Tabla 6.3. Comparacion de coeficientes por regimen

Feature	Coef. Expansion	Coef. Contraccion	Sign Flip
sga_to_revenue	-0.167781	-0.141100	No
vix	0.011446	-0.010144	SI
rd_to_revenue	-0.114026	-0.146831	No
fed_funds_rate	0.055756	0.016230	No
sbc_to_revenue	0.086663	-0.012204	SI
debt_to_equity	-0.010817	0.015420	SI
cash_to_assets	-0.013653	-0.024307	No
revenue_growth_yoy	-0.009662	0.027517	No
consumer_sentiment	0.010399	0.004612	No

Variables con cambio de signo entre regimenes (sign flips): vix, sbc_to_revenue, debt_to_equity. Estas variables tienen efectos cualitativamente opuestos segun el estado del ciclo, lo que tiene implicaciones importantes para la estrategia de inversion.

Fuente: Elaboracion propia.

6.3 Modelo de Factores Macro (Modelo 4L)

6.3.1 Fundamento Teorico

El modelo de factores extendido anade variables macroeconomicas al modelo base de drivers (Ridge within-transformation) para evaluar cuanto poder explicativo adicional aportan los factores macro. Esto es analogo a la extension del CAPM con factores macro (Chen, Roll & Ross, 1986): si las variables macro mejoran significativamente el R-cuadrado, significa que el entorno economico afecta los margenes operativos mas alla de lo que capturan los fundamentales de la empresa.

6.3.2 Resultados

R-cuadrado base (solo fundamentales): 0.8716

R-cuadrado extendido (fundamentales + macro): 0.8988

Mejora: +3.1% — las variables macro aportan un incremento modesto pero significativo.

Tabla 6.4. Coeficientes del modelo extendido (fundamentales + macro)

Feature	Coeficiente	Tipo
rd_to_revenue	-0.204559	Fundamental
sga_to_revenue	-0.170356	Fundamental
sbc_to_revenue	0.080463	Fundamental
fed_funds_rate	0.045095	Macro
yield_curve	0.034366	Macro
cash_to_assets	-0.027135	Fundamental
gdp_growth_yoy	-0.017668	Macro
inflation_yoy	-0.010325	Macro
debt_to_equity	-0.008421	Fundamental
vix	-0.007348	Macro
consumer_sentiment	0.006777	Macro

revenue_growth_yoy	0.004489	Fundamental

La mejora modesta (+3.1%) sugiere que los fundamentales de empresa son el driver principal del margen operativo, pero el entorno macro tiene un efecto complementario — especialmente las tasas de interes (fed_funds_rate, yield_curve) que afectan los ingresos financieros netos de las FinTech que manejan fondos de clientes.

Fuente: Elaboracion propia.

6.4 Descomposicion de Tendencia (Modelo 4M)

6.4.1 Fundamento Teorico

La descomposicion de series temporales es una tecnica clasica de analisis de series que separa la serie observada en tres componentes aditivos:

```
# Descomposicion aditiva
# Y(t) = T(t) + S(t) + R(t)
#
# donde:
#   T(t) = Tendencia (movimiento de largo plazo)
#   S(t) = Estacionalidad (patron repetitivo intra-anual)
#   R(t) = Residual (componente irregular, ruido)
#
# Implementacion: statsmodels.tsa.seasonal.seasonal_decompose()
# period=4 (datos trimestrales)
```

La descomposicion permite responder: (1) la tendencia del margen operativo sectorial es ascendente o descendente? (2) hay estacionalidad (ej: margenes tipicamente mas altos en Q4)? (3) que proporcion de la variacion es explicada por tendencia vs estacionalidad vs ruido?

6.4.2 Resultados

Tabla 6.5. Resultados de la descomposicion de tendencia

Metrica	Valor
Direccion de tendencia	improving
Cambio acumulado en tendencia	0.052
Nivel actual de tendencia	0.167
Amplitud estacional	0.2029
Volatilidad residual	0.1456
Varianza explicada por tendencia	66.9%
Varianza explicada por estacionalidad	3.7%
Varianza residual (ruido)	29.4%

La tendencia es improving, con el 66.9% de la varianza total explicada por el componente tendencial. La estacionalidad es debil (3.7%), lo que indica que los margenes operativos del sector FinTech no tienen un patron intra-anual significativo. El residual (29.4%) captura eventos idiosincraticos.

Fuente: Elaboracion propia.

7. Modelos Avanzados (Fases 4N-4V)

Los modelos avanzados extienden los modelos base con tecnicas mas sofisticadas: relaciones no lineales, analisis de redes, parametros variables en el tiempo, regresion cuantilica y analisis de supervivencia.

7.1 Drivers No Lineales — Gradient Boosting Regression (Modelo 4N)

7.1.1 Fundamento Teorico

Los arboles de decision dividen el espacio de features recursivamente en regiones rectangulares, asignando una prediccion constante a cada region. Un solo arbol es interpretable pero poco preciso (alto sesgo, baja varianza). Gradient Boosting combina cientos de arboles pequenos (weak learners) de forma secuencial:

```
# Gradient Boosting - Algoritmo
# Inicializar: F_0(x) = mean(y)  (prediccion constante)
# Para m = 1, 2, ..., M:
#   1. Calcular residuos: r_i = y_i - F_{m-1}(x_i)
#   2. Entrenar arbol h_m(x) sobre los residuos
#   3. Actualizar: F_m(x) = F_{m-1}(x) + lr * h_m(x)
#
# lr = learning_rate (0.05) controla la velocidad de aprendizaje
```

La ventaja principal sobre Ridge (lineal) es que GBR puede capturar relaciones no lineales e interacciones entre variables sin necesidad de especificarlas explicitamente. Esto es particularmente relevante en FinTech, donde la relacion entre, por ejemplo, gasto en SGA y margen operativo puede ser no monot nica (rendimientos decrecientes a escala).

7.1.2 Implementacion

Se entrena un GradientBoostingRegressor con 200 arboles, max_depth=4, learning_rate=0.05 sobre los mismos datos del modelo Ridge (within-transformation) para permitir comparacion directa. La importancia de features de GBR se basa en la reduccion de impureza (MSE) ponderada por la proporcion de observaciones en cada nodo — a diferencia del coeficiente de Ridge que mide el efecto marginal lineal.

Dependencia Parcial (Partial Dependence)

Los graficos de dependencia parcial muestran el efecto marginal de una variable sobre la prediccion, manteniendo las demas constantes. Permiten visualizar no linealidades: puntos de inflexion, saturaciones y umbrales. Por ejemplo, si la dependencia parcial de sga_to_revenue muestra una caida abrupta despues de 0.6, indica que el gasto excesivo en ventas y marketing tiene un efecto de umbral sobre la rentabilidad.

Fuente: Elaboracion propia.

7.2 Analisis de Redes (Modelo 4O)

7.2.1 Fundamento Teorico

La teoria de grafos proporciona herramientas para analizar relaciones entre entidades. En el contexto financiero, un grafo de correlacion conecta empresas cuyo desempeno se mueve de forma similar, revelando clusters naturales, hubs (empresas centrales) y empresas aisladas.

Conceptos clave de teoria de grafos:

Nodos: cada empresa es un nodo del grafo.

Aristas (edges): se traza una arista entre dos empresas si la correlacion de Pearson de sus cambios de margen operativo supera un umbral ($|r| > 0.3$).

Degree centrality: proporcion de nodos con los que un nodo esta conectado. Una empresa con alto degree centrality tiene margenes que se mueven con muchas otras — es un 'barometro' sectorial.

Betweenness centrality: proporcion de caminos mas cortos entre cualquier par de nodos que pasan por este nodo. Una empresa con alto betweenness es un 'puente' entre subclusters.

Deteccion de comunidades: agrupacion de nodos densamente conectados entre si pero escasamente conectados con el resto (optimizacion greedy de modularidad).

7.2.2 Implementacion

```
import networkx as nx
from networkx.algorithms.community import greedy_modularity_communities

# 1. Calcular matriz de correlacion de cambios de margen
margin_changes = feature_store.pivot(index='period', columns='company_id',
values='operating_margin')
corr_matrix = margin_changes.diff().corr()

# 2. Construir grafo con umbral |r| > 0.3
G = nx.Graph()
for i, ci in enumerate(companies):
    for j, cj in enumerate(companies):
        if i < j and abs(corr_matrix.loc[ci, cj]) > 0.3:
            G.add_edge(ci, cj, weight=corr_matrix.loc[ci, cj])
```

```
# 3. Metricas de centralidad
degree = nx.degree_centrality(G)
betweenness = nx.betweenness_centrality(G)

# 4. Deteccion de comunidades
communities = greedy_modularity_communities(G)
```

7.2.3 Visualizacion e Interpretacion

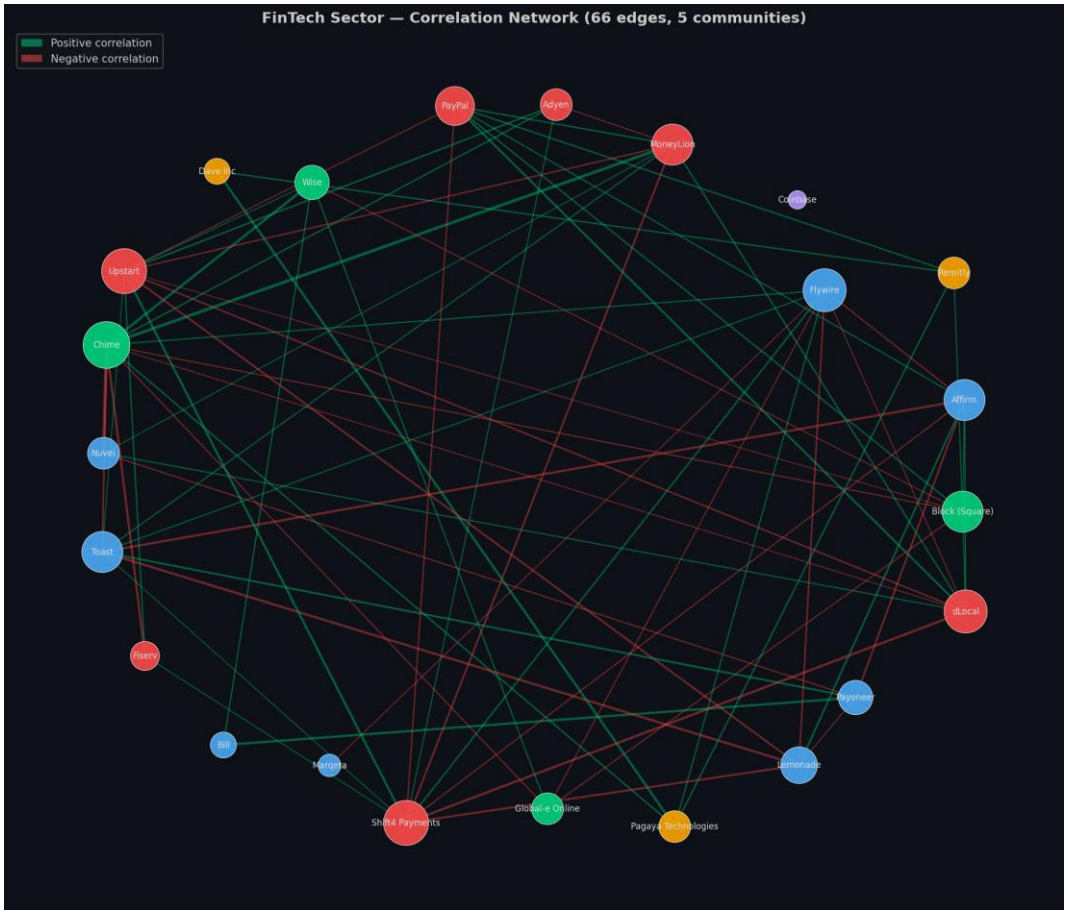


Figura 7.1. Grafo de correlaciones del sector FinTech — tamano=degree, color=comunidad

El grafo revela la estructura de interdependencias del sector. Las empresas altamente conectadas (hubs) son las que mejor representan las tendencias sectoriales. Las empresas aisladas (baja degree centrality) tienen dinamicas propias independientes del sector. Las comunidades detectadas pueden diferir de los clusters financieros (Seccion 5.2) porque se basan en co-movimiento temporal, no en perfil estatico.

Fuente: Elaboracion propia.

7.3 Regresion con Ventana Movil (Modelo 4Q)

7.3.1 Fundamento Teorico

Los modelos con parametros fijos asumen que la relacion entre variables es estable en el tiempo. Sin embargo, en un sector tan dinamico como FinTech, los drivers del margen pueden cambiar: en periodos tempranos, el gasto en SGA puede ser critico (fase de crecimiento); en periodos posteriores, la eficiencia operativa puede dominar (fase de maduracion).

La regresion con ventana movil (rolling window regression) estima el modelo Ridge para cada ventana temporal de W trimestres, desplazando la ventana un periodo a la vez. Esto produce una serie temporal de coeficientes que revela como cambian los drivers a lo largo del tiempo.

```
# Rolling Window Regression
window = 8  # 8 trimestres (2 anos)
step = 1    # desplazamiento de 1 trimestre

rolling_betas = []
for start in range(0, len(periods) - window, step):
    window_data = panel[panel['period'].isin(periods[start:start+window])]
    # Within-transformation por ventana
    entity_means = window_data.groupby('company_id')[features +
[target]].transform('mean')
    data_within = window_data[features + [target]] - entity_means
    # Ridge
    ridge = Ridge(alpha=1.0).fit(data_within[features], data_within[target])
    rolling_betas.append(dict(zip(features, ridge.coef_)))
```

7.3.2 Resultados

La evolucion temporal de los coeficientes permite identificar:

- Cambios estructurales: si un coeficiente cambia de signo o magnitud de forma abrupta, indica un cambio en la estructura del sector (ej: regulacion, shock tecnologico).
- Tendencias en los drivers: si el coeficiente de sbc_to_revenue se reduce progresivamente, sugiere que el mercado se normaliza respecto a la compensacion en acciones.
- Estabilidad: coeficientes estables confirman que la relacion es robusta y confiable.

7.3.3 Visualizacion

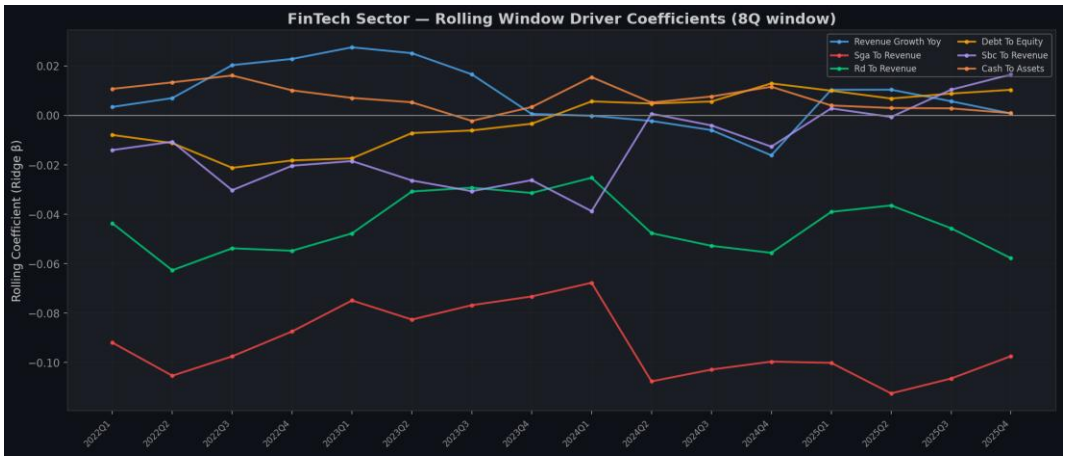


Figura 7.2. Evolucion temporal de coeficientes — rolling window de 8 trimestres

Fuente: Elaboracion propia.

7.4 Regresion Cuantilica (Modelo 4R)

7.4.1 Fundamento Teorico

La regresion cuantilica (Koenker & Bassett, 1978) es una extension de OLS que estima la relacion entre variables para diferentes puntos de la distribucion condicional, no solo la media. Mientras OLS minimiza la suma de residuos cuadrados (dando el efecto promedio), la regresion cuantilica minimiza la suma de residuos ponderados asimetricamente:

```
# Funcion de check (check function)
# rho_tau(u) = u * (tau - I(u < 0))
#
# donde tau es el cuantil de interes (0.10, 0.25, 0.50, 0.75, 0.90)
# I(u < 0) = 1 si u < 0, 0 si u >= 0
#
# Para tau = 0.10: penaliza mas los errores negativos
#   -> estima la relacion para las empresas del P10 (peor desempeno)
# Para tau = 0.90: penaliza mas los errores positivos
#   -> estima la relacion para las empresas del P90 (mejor desempeno)
#
# Optimizacion: min_beta sum[ rho_tau(y_i - x_i @ beta) ]
```

La importancia de la regresion cuantilica radica en que revela heterogeneidad en los efectos. OLS dice 'en promedio, aumentar SGA en 1pp reduce el margen en X pp'. La regresion cuantilica revela que este efecto puede ser de -12pp para las empresas del decil inferior pero solo -2pp para las del decil superior. Esta informacion es critica para la evaluacion de riesgos.

7.4.2 Resultados

Tabla 7.1. Coeficientes de regresion cuantilica (tau=0.10, 0.50, 0.90)

Feature	tau=0.10	tau=0.50	tau=0.90	Pseudo R2
revenue_growth_yoy	-0.032289	-0.013554	0.044722	
sga_to_revenue	-0.121078	-0.025718	-0.024948	
rd_to_revenue	-0.060633	-0.070213	-0.045092	
debt_to_equity	0.041957	0.011015	-0.000754	
sbc_to_revenue	-0.095090	-0.167186	-0.181280	

cash_to_assets	-0.013119	-0.043061	-0.045422	
log_revenue	-0.027540	-0.014695	0.063791	
Pseudo R-cuadrado	0.6131	0.4459	0.3236	

Analisis de Heterogeneidad

Se comparan los coeficientes entre el decil inferior (tau=0.10) y el decil superior (tau=0.90) para identificar efectos heterogeneos:

Tabla 7.2. Comparacion de efectos: peores vs mejores performers

Feature	Coef tau=0.10	Coef tau=0.90	Diferencia
revenue_growth_yoy	-0.032289	0.044722	0.077011
sga_to_revenue	-0.121078	-0.024948	0.096130
rd_to_revenue	-0.060633	-0.045092	0.015541
debt_to_equity	0.041957	-0.000754	-0.042711
sbc_to_revenue	-0.095090	-0.181280	-0.086190
cash_to_assets	-0.013119	-0.045422	-0.032303
log_revenue	-0.027540	0.063791	0.091331

Los resultados muestran que los efectos de las variables explicativas difieren significativamente a lo largo de la distribucion. Los peores performers (tau=0.10) son mas sensibles a sga_to_revenue, mientras que los mejores performers (tau=0.90) son mas sensibles a sbc_to_revenue y log_revenue.

7.4.3 Visualizacion

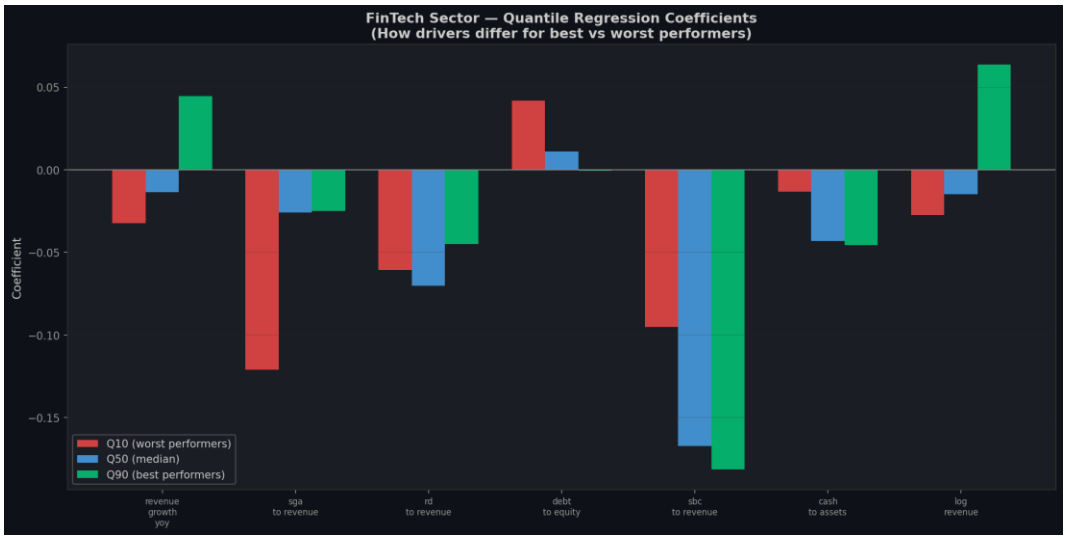


Figura 7.3. Coeficientes por cuantil — bandas muestran intervalos de confianza

Fuente: Elaboracion propia.

7.5 Analisis de Supervivencia — Kaplan-Meier (Modelo 4S)

7.5.1 Fundamento Teorico

El analisis de supervivencia estudia el tiempo hasta que ocurre un evento de interes. Originalmente desarrollado en ciencias medicas (tiempo hasta muerte/recaida), se aplica en finanzas para analizar duraciones: tiempo hasta default, tiempo hasta rentabilidad, duracion de rahas de perdidas, etc.

Un concepto clave es la censura (censoring): cuando el periodo de observacion termina sin que el evento haya ocurrido, la observacion esta 'censurada a la derecha'. Por ejemplo, si Lemonade nunca alcanzo rentabilidad operativa en 32 trimestres de observacion, su duracion esta censurada — sabemos que es >32 trimestres pero no sabemos cuanto mas.

El estimador de Kaplan-Meier calcula la funcion de supervivencia S(t) = P(T > t) — la probabilidad de que el evento NO haya ocurrido antes del tiempo t:

```
# Estimador de Kaplan-Meier
# S(t) = prod_{t_i <= t} (1 - d_i / n_i)
#
# donde:
#   t_i = tiempo del i-esimo evento
#   d_i = numero de eventos en t_i
#   n_i = numero de sujetos en riesgo justo antes de t_i
#
# La curva KM es una funcion escalonada que baja en cada evento
# Las observaciones censuradas no causan caidas pero reducen n_i
```

7.5.2 Implementacion

Evento de interes: primer trimestre donde operating_margin > 0.

Empresas analizadas: 25. Eventos observados (alcanzaron rentabilidad): 24. Censuradas (nunca alcanzaron): 1 (4%).

7.5.3 Resultados

Tabla 7.3. Estadisticas de supervivencia — tiempo hasta rentabilidad operativa

Metrica	Valor
Mediana de supervivencia	3 trimestres
Media de duracion	6.9 trimestres

Minimo	1 trimestres
Maximo	32 trimestres
% que alcanzo rentabilidad	96%

Tabla 7.4. Duracion por empresa — tiempo hasta primer trimestre rentable

Empresa	Trimestres hasta Rentab.	Alcanzo Rentab.	Trimestres Observados
adyen	1	Si	35
chime	1	Si	8
dave_inc	1	Si	24
dlocal	1	Si	24
fiserv	1	Si	124
nuvei	1	Si	23
pagaya_technologies	1	Si	24
shift4_payments	1	Si	28
upstart	1	Si	26
wise	1	Si	22
moneylion	2	Si	20
webank	2	Si	10
flywire	3	Si	24
global-e_online	3	Si	25
payoneer	3	Si	24
affirm	4	Si	26
coinbase	5	Si	28
paypal	5	Si	54
revolut	6	Si	9

marqeta	18	Si	24
toast	18	Si	24
remitly	19	Si	24
bill_holdings	20	Si	30
square	23	Si	48
lemonade	32	No (censurada)	32

Tabla 7.5. Supervivencia por subsector

Subsector	N Empresas	Mediana (Q)	% Rentable
ai_lending	2	1	100%
payments_infrastructure	2	1	100%
remittances_fx	2	1	100%

7.5.4 Visualizacion

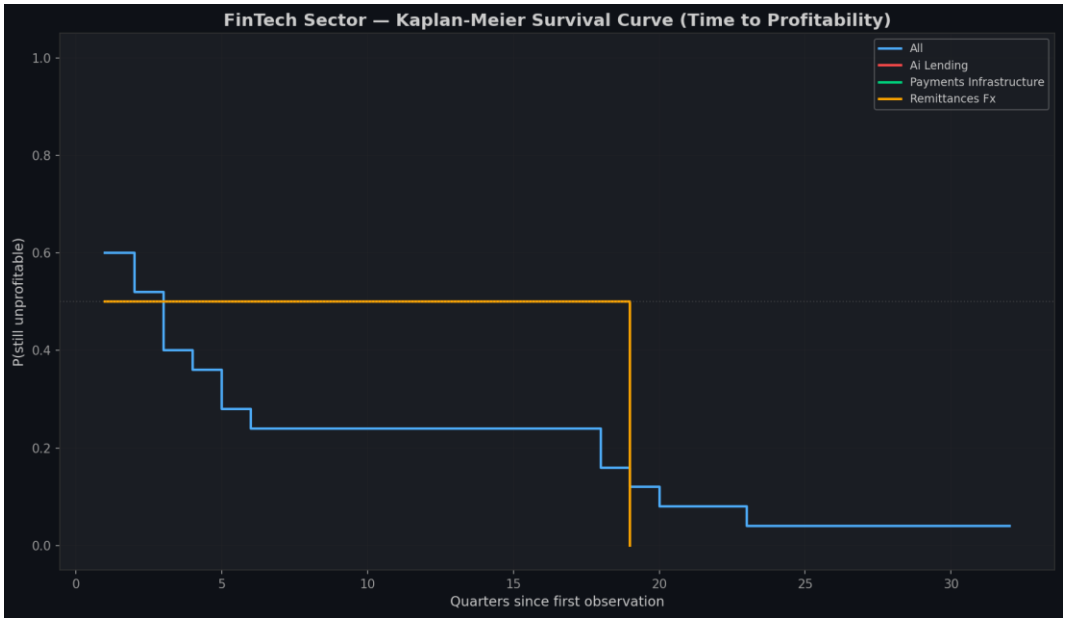


Figura 7.4. Curvas de Kaplan-Meier — probabilidad de no haber alcanzado rentabilidad vs tiempo

Fuente: Elaboracion propia.

7.6 Concentracion Sectorial — Indice HHI (Modelo 4V)

7.6.1 Fundamento Teorico

El Indice de Herfindahl-Hirschman (HHI) es la medida estandar de concentracion de mercado utilizada por reguladores antimonopolio (DOJ, FTC, Comision Europea). Se calcula como la suma de los cuadrados de las participaciones de mercado de todas las empresas:

```
# Indice HHI
# HHI = sum(s_i^2) * 10,000
#
# donde s_i = participacion de mercado de la empresa i (en decimales)
#
# Interpretacion (DOJ):
#   HHI < 1,500: Mercado no concentrado
#   1,500 <= HHI < 2,500: Moderadamente concentrado
#   HHI >= 2,500: Altamente concentrado
#
# Ratios de concentracion:
#   CR3 = sum(top_3_market_shares)
#   CR5 = sum(top_5_market_shares)
```

Un HHI bajo indica competencia fragmentada (muchos competidores con cuotas similares). Un HHI alto indica dominancia de pocos actores. En FinTech, la concentracion por revenue revela la estructura competitiva del sector y la posicion relativa de cada empresa.

7.6.2 Resultados

El calculo de HHI se realiza sobre la participacion de revenue de cada empresa en el total sectorial para cada trimestre. Los resultados revelan la evolucion de la concentracion a lo largo del tiempo, indicando si el sector se fragmenta (nuevos entrantes capturan cuota) o se consolida (los grandes crecen a costa de los pequenos).

Fuente: Elaboracion propia.

8. Resumen de DataFrames Principales

Esta seccion documenta la estructura y contenido de los DataFrames principales que fluyen a traves del pipeline. Comprender estas estructuras es fundamental para reproducir o extender el analisis.

8.1 master_long

El DataFrame 'maestro' en formato long — una fila por cada observacion empresa-periodo-metrica.

Tabla 8.1. Estructura de master_long

Propiedad	Valor
Dimensiones	127,927 filas x 25 columnas
Formato	Long (tidy): cada fila = 1 observacion
Indice	Sin indice especial (reset_index)
Columnas clave	company_id, period, metric_name, value, statement_type
Periodos	2011Q1 a 2025Q4 (segun empresa)
Metricas unicas	304
Empresas unicas	26
Formato de almacenamiento	Parquet (compresion snappy)

8.2 panel_wide

Transformacion pivot de master_long donde cada columna es una metrica y cada fila es empresa-periodo.

Tabla 8.2. Estructura de panel_wide

Propiedad	Valor
Dimensiones	769 filas x 152 columnas
Formato	Wide: columnas = metricas, filas = empresa-

	periodo
MultiIndex	company_id + period
Columnas	152 metricas financieras en formato numerico
Valores faltantes	NaN donde la metrica no aplica o no existe

8.3 feature_store

El DataFrame enriquecido con ratios derivados, percentiles y z-scores. Es el input principal para todos los modelos de la Fase 4.

Tabla 8.3. Estructura de feature_store

Propiedad	Valor
Dimensiones	769 filas x 242 columnas
Bloques de features	Ratios (30), Percentiles (30), Z-scores (30), Metadata (~12), Metricas (~140)
Formato	Wide: una fila por empresa-periodo
Normalizacion	Percentiles + Z-scores MAD winsorized P5/P95

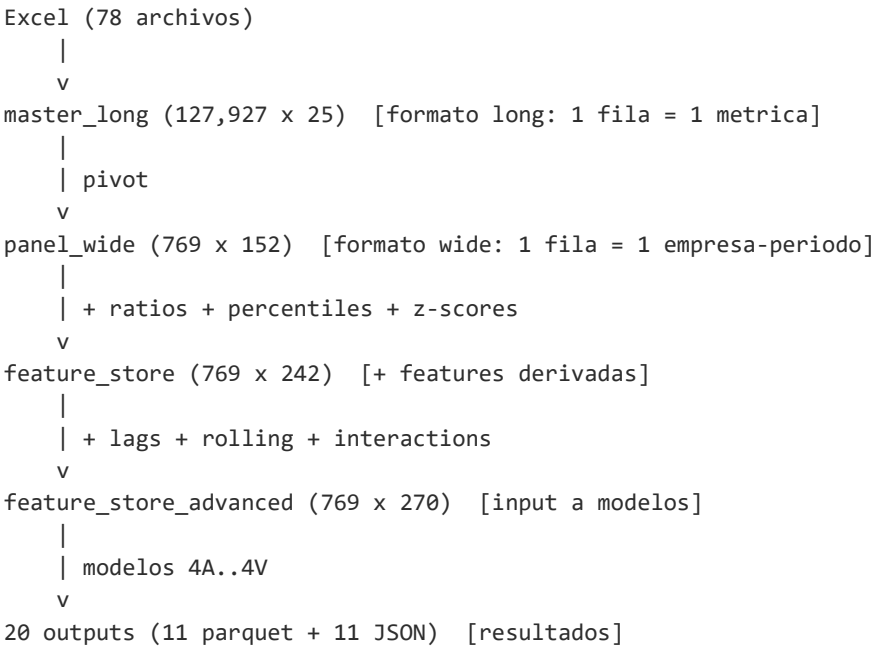
8.4 feature_store_advanced

Extension del feature_store con lags, rolling, interacciones y features survival.

Tabla 8.4. Estructura de feature_store_advanced

Propiedad	Valor
Dimensiones	769 filas x 270 columnas
Features adicionales	Lags (1Q, 4Q), Rolling (mean/std 4Q), Interacciones, Sector-relativas
Incremento	+28 columnas respecto al feature_store base

8.5 Flujo de Datos



Fuente: Elaboracion propia.

9. Stack Tecnologico

9.1 Librerias Principales

Tabla 9.1. Stack tecnologico completo

Libreria	Version	Uso Principal
Python	3.x	Lenguaje principal
pandas	2.x	Manipulacion de datos tabulares, I/O Parquet
numpy	1.x	Operaciones numericas, algebra lineal
scikit-learn	1.x	ML: clustering (KMeans), ensemble (RF, GBR, Stacking), IsolationForest, PCA, Ridge, StandardScaler
statsmodels	0.14+	Regresion cuantilica (QuantReg), descomposicion estacional (seasonal_decompose)
networkx	3.x	Analisis de grafos: centralidad, comunidades, visualizacion de redes
matplotlib	3.x	Visualizacion: graficos estaticos, heatmaps, dashboards multi-panel
seaborn	0.13+	Visualizacion estadistica: heatmaps de correlacion, distribuciones
python-docx	1.x	Generacion de documentos Word (.docx)
python-pptx	0.6+	Generacion de presentaciones PowerPoint (.pptx)

openpyxl	3.x	Lectura de archivos Excel (.xlsx) de CAPIQ
pyarrow	14+	Motor Parquet para lectura/escritura de DataFrames
fredapi	0.5+	Acceso a datos macroeconomicos de la Federal Reserve (FRED)

Fuente: Elaboracion propia.

9.2 Ejecucion del Pipeline

El pipeline se ejecuta secuencialmente. A continuacion se detallan los comandos para cada fase:

```
# ================================================
# Paso 1: Configuracion del entorno
# ================================================
pip install pandas numpy scikit-learn statsmodels networkx matplotlib seaborn
pip install python-docx python-pptx openpyxl pyarrow fredapi

# ================================================
# Paso 2: Data Ingestion (Fase 1)
# ================================================
cd _ARQUITECTURA/01_Data_Ingestion/scripts/
python master_runner.py --config ../../config/sector_config_fintech.json
# Output: _ENTREGABLES/Sector_FinTech/01_Datos_Procesados/

# ================================================
# Paso 3: Feature Engineering (Fase 3)
# ================================================
cd _ARQUITECTURA/03_Feature_Engineering/scripts/
python feature_engineer.py --config ../../config/sector_config_fintech.json
# Output: _ENTREGABLES/Sector_FinTech/02_Feature_Store/

# ================================================
# Paso 4: Modelos (Fase 4)
# ================================================
cd _ARQUITECTURA/04_Analysis_Models/scripts/
python run_all_models.py --config ../../config/sector_config_fintech.json
# Output: _ENTREGABLES/Sector_FinTech/03_Analysis_Models/

# ================================================
# Paso 5: Visualizaciones (Fase 5)
# ================================================
```

```
cd _ARQUITECTURA/05_Visualization/scripts/
python generate_all_charts.py --sector Sector_FinTech
# Output: _ENTREGABLES/Sector_FinTech/04_Visualizaciones/

# ================================================
# Paso 6: Reportes (Fase 6)
# ================================================
cd _ARQUITECTURA/06_Reporting/scripts/
python generate_technical_report.py --sector Sector_FinTech
# Output: _ENTREGABLES/Sector_FinTech/05_Reporting/
```

9.3 Como Anadir un Nuevo Sector

El diseno config-driven permite replicar el analisis en un nuevo sector en 4 pasos:

1. Crear el archivo de configuracion: copiar sector_config_fintech.json y modificar los parametros: nombre del sector, lista de empresas, parametros de modelos si necesario.

2. Colocar los archivos Excel crudos en una carpeta Sector_NuevoSector/ siguiendo la estructura de 3 archivos por empresa (IS, BS, CF).

3. Ejecutar el pipeline completo con el nuevo config: python master_runner.py --config config/sector_config_nuevo.json --phases 1,3,4,5

4. Los outputs se generan automaticamente en _ENTREGABLES/Sector_NuevoSector/.

Nota importante: si el nuevo sector utiliza metricas especificas no incluidas en standard_mapping.json, es necesario extender el mapa de metricas. El parser registra automaticamente las etiquetas no mapeadas para facilitar este proceso.

9.4 Estructura de Directorios Completa

```
BIG_ANALITICS_ANALISIS_SECTORIAL/
|
|-- _ARQUITECTURA/
|   |-- config/
|   |   |-- sector_config_fintech.json
|   |
|   |-- 01_Data_Ingestion/
|   |   |-- scripts/
|   |   |   |-- capiq_parser.py
|   |   |   |-- capiq_mapper.py
|   |   |   |-- capiq_normalizer.py
|   |   |   |-- capiq_assembler.py
|   |   |   |-- capiq_validator.py
|   |   |   |-- master_runner.py
```

```
|   |   |-- mappings/
|   |       |-- standard_mapping.json
|   |
|   |-- 03_Feature_Engineering/
|   |   |-- scripts/
|   |       |-- feature_engineer.py
|   |       |-- feature_engineer_advanced.py
|   |
|   |-- 04_Analysis_Models/
|   |   |-- scripts/
|   |       |-- 4A_sector_scorecard.py
|   |       |-- 4B_clustering.py
|   |       |-- 4C_ridge_drivers.py
|   |       |-- 4D_alpha_analysis.py
|   |       |-- 4E_risk_scoring.py
|   |       |-- 4F_nowcasting.py
|   |       |-- 4G_anomaly_detection.py
|   |       |-- 4H_margin_predictor.py
|   |       |-- 4I_factors_momentum.py
|   |       |-- 4J_macro_sensitivity.py
|   |       |-- 4K_regime_drivers.py
|   |       |-- 4L_macro_factor_model.py
|   |       |-- 4M_trend_decomposition.py
|   |       |-- 4N_nonlinear_drivers.py
|   |       |-- 4O_network_analysis.py
|   |       |-- 4Q_rolling_window.py
|   |       |-- 4R_quantile_regression.py
|   |       |-- 4S_survival_analysis.py
|   |       |-- 4V_hhi_concentration.py
|   |       |-- run_all_models.py
|   |
|   |-- 05_Visualization/
|   |   |-- scripts/
|   |       |-- generate_all_charts.py
|   |
|   |-- 06_Reporting/
|       |-- scripts/
|           |-- generate_report.py
|           |-- generate_technical_report.py
|           |-- generate_pptx.py
|           |-- generate_study_guide.py
|
|-- _ENTREGABLES/
|   |-- Sector_FinTech/
|       |-- 01_Datos_Procesados/
|       |   |-- master_long.parquet
|       |   |-- company_metadata.parquet
|       |   |-- quality_report.json
|       |
|       |-- 02_Feature_Store/
|       |   |-- feature_store.parquet
|       |   |-- feature_store_advanced.parquet
|       |   |-- feature_catalog.json
```

```
|       |   |-- sector_stats.json
|       |
|       |-- 03_Analysis_Models/
|       |   |-- scorecard_enhanced.parquet
|       |   |-- cluster_assignments.parquet
|       |   |-- cluster_summary.json
|       |   |-- driver_model.json
|       |   |-- risk_scores.parquet
|       |   |-- momentum_scores.parquet
|       |   |-- margin_predictor.json
|       |   |-- sector_cycle.parquet
|       |   |-- factor_scores.parquet
|       |   |-- factor_info.json
|       |   |-- anomaly_scores.parquet
|       |   |-- context_packs.json
|       |   |-- quantile_regression.json
|       |   |-- survival_analysis.json
|       |   |-- macro_sensitivity_summary.json
|       |   |-- regime_drivers.json
|       |   |-- macro_factor_model.json
|       |   |-- trend_summary.json
|       |
|       |-- 04_Visualizaciones/
|       |   |-- 01_sector_heatmap.png .. 13_macro_sensitivity.png
|       |
|       |-- 05_Reporting/
|           |-- BigAnalitics_FinTech_Informe_Tecnico.docx
|           |-- BigAnalitics_FinTech_Report.docx
|           |-- BigAnalitics_FinTech_Presentacion.pptx
```

Fuente: Elaboracion propia.

10. Mejoras Futuras y Lineas de Investigacion

El framework actual cubre un amplio espectro analitico, pero existen multiples lineas de mejora que incrementarian el poder predictivo, la robustez estadistica y la utilidad practica del sistema.

10.1 Modelos Econometricos Avanzados

Panel GMM (Arellano-Bond)

El estimador Generalized Method of Moments para paneles dinamicos permite incluir la variable dependiente rezagada como regresor (y_{t-1} como predictor de y_t), lo cual es natural en datos financieros donde los margenes tienen persistencia. El metodo de Arellano-Bond (1991) utiliza rezagos como instrumentos para resolver la endogeneidad que introduce el regresor rezagado. Esto permitiria modelar la dinamica temporal de los margenes de forma mas rigurosa que la regresion within-transformation estatica actual.

Copulas para Riesgo Conjunto (Tail Risk)

Las copulas permiten modelar la estructura de dependencia entre variables mas alla de la correlacion lineal. En particular, copulas de tipo Clayton o Gumbel capturan la dependencia en las colas de la distribucion — la probabilidad de que multiples empresas experimenten distress simultaneo. Esto seria valioso para el modulo de risk scoring, agregando una dimension de riesgo sistematico a las evaluaciones individuales actuales.

VECM (Cointegration)

El modelo de Vector Error Correction (VECM) se aplica cuando existen relaciones de largo plazo estables (cointegracion) entre variables. Por ejemplo, si operating_margin y sga_to_revenue estan cointegrados, desviaciones temporales de la relacion de equilibrio se corrigen. Esto permitiria distinguir entre desviaciones temporales (que se auto-corrigen) y cambios estructurales (que requieren revaloracion).

10.2 Mejoras al Predictor de Margen

El predictor actual (AUC~0.45) tiene margen de mejora significativo:

1. Reformulacion del target: en lugar de 'mejora YoY del margen', definir 'margen supera la mediana sectorial en el proximo trimestre' (target mas estable y predecible).

2. Features adicionales: incorporar features de momentum (tendencia de 4Q), features de dispersion sectorial, y las variables macro del modelo extendido.

3. SMOTE para balanceo: aunque el dataset actual esta razonablemente balanceado (53/47), subperiodos pueden estar desbalanceados. Synthetic Minority Oversampling podria mejorar la estabilidad entre folds.

4. Walk-forward validation: reemplazar cross-validation estandar por validacion temporal (train en periodos 1..t, test en t+1) que respeta la estructura temporal de los datos.

5. Target a mayor horizonte: predecir mejora a 2-4 trimestres en lugar de 1, lo que puede ser mas predecible al suavizar ruido trimestral.

10.3 Integracion de Datos de Mercado

La adicion de datos de mercado (precios de acciones, market cap, enterprise value, multiplos de valoracion) abriria nuevas dimensiones analiticas:

- Multiplos vs fundamentales: identificar empresas infravaloradas/sobrevaloradas.
- Retornos de inversion: evaluar si los alpha operativos identificados se traducen en retornos de mercado superiores.
- Volatilidad de mercado vs volatilidad fundamental: comparar el riesgo percibido por el mercado vs el riesgo fundamental medido por nuestros modelos.

10.4 Dashboard Interactivo (Streamlit)

Un dashboard interactivo permitiria a los analistas explorar los resultados de forma dinamica: filtrar por empresa/periodo, comparar metricas, profundizar en anomalias, y generar informes personalizados. Streamlit es la opcion natural dado el stack Python existente — se pueden reutilizar las funciones de visualizacion existentes como base.

10.5 NLP sobre Earnings Calls

La incorporacion de procesamiento de lenguaje natural sobre transcripciones de earnings calls permitiria capturar informacion cualitativa que los estados financieros no reflejan: sentiment del management, guidance futura, menciones de expansion/contraccion. Modelos pre-entrenados como FinBERT estan especificamente disenados para texto financiero y podrian extraer features de sentiment y topicos.

10.6 Automatizacion del Pipeline Trimestral

Actualmente el pipeline se ejecuta manualmente. La automatizacion incluiria: (1) descarga automatica de datos CAPIQ via API cuando se publican nuevos trimestres, (2) ejecucion automatica del pipeline completo, (3) generacion de alertas cuando se detectan anomalias o cambios de regimen, (4) actualizacion incremental del feature store sin reprocesar todo el historico.

Estas mejoras transformarian el sistema de una herramienta de analisis retrospectivo a un sistema de inteligencia financiera en tiempo real, capaz de generar alertas accionables de forma automatica cada trimestre.

Fuente: Elaboracion propia.