



Pricing & Portfolio Management para Seguros No Vida con {PYTHON}

{Temario: //}



1: //

Introducción y Fundamentos de Python para Actuarios

1.

Objetivos:

Nivelar conocimientos de Python, configurar el entorno y entender el flujo de trabajo para análisis actuarial.

2.

Teoría:

- Bienvenida, objetivos del curso y panorama general.
- El rol de Python en el pricing y portfolio management moderno. Ventajas sobre herramientas tradicionales (Excel, etc.).
- Repaso rápido de conceptos actuariales clave (Prima Pura, Prima de Riesgo, Prima Comercial, Siniestralidad, Frecuencia, Severidad, Exposición).

3.

Práctica (Python):

- Instalación y configuración del entorno (Anaconda, Jupyter Notebooks/Labo VS Code).
- Introducción a Python: Tipos de datos básicos, estructuras de control (loops, condicionales), funciones.
- Pandas: Introducción a Series y DataFrames. Carga de datos (CSV, Excel), inspección básica (head, info, describe), selección y filtrado de datos.
- NumPy: Operaciones básicas con arrays, importantes para cálculos numéricos.

4.

Material:

Datasets de ejemplo simplificados (pólizas, siniestros).

2: //

Tratamiento y Visualización Exploratoria de Datos (EDA)

1.

Objetivos:

Aprender a limpiar, transformar y visualizar datos de seguros para obtener insights iniciales.

2.

Teoría:

- Importancia de la calidad de los datos en el modelado actuarial.
- Tipos de datos comunes en seguros no vida (categóricos, numéricos, fechas).
- Técnicas de limpieza: Manejo de valores faltantes (imputación simple/avanzada), detección y tratamiento de outliers.
- Principios de visualización efectiva. Qué gráficos usar para qué tipo de datos/preguntas.

3.

Práctica (Python):

- **Polars:** Limpieza de datos (dropna, fillna), transformación de variables (creación de nuevas features, binning, encoding categórico - OneHotEncoder). Agregaciones (groupby). Merging/Joining datasets.
- **Matplotlib & Seaborn:** Creación de gráficos esenciales:
 - Univariados: Histogramas, Boxplots, Density Plots (para entender distribuciones de severidad, edad, etc.).
 - Bivariados: Scatter plots (relación entre variables), Bar plots (comparación de categorías), Heatmaps (correlaciones).
- **Aplicación a datos de seguros:** Visualizar patrones de frecuencia/severidad por CCAA, tipo de vehículo, edad del asegurado, etc.

4.

Material:

Dataset más complejo con datos "sucios" para limpiar y explorar..



3:// Modelado de Primas I - Modelos Lineales Generalizados (GLM) para Frecuencia

1. Objetivos:

Entender y aplicar GLMs para modelar la frecuencia de siniestros.

2. Teoría:

- Fundamentos de GLMs: Familia Exponencial de distribuciones, Función de enlace (Link Function), Componente Aleatorio y Sistemático.
- Distribución Poisson para conteos (frecuencia). Supuestos y limitaciones.
- Concepto de Exposición (Offset) en modelos de frecuencia.
- Interpretación de coeficientes (escala logarítmica y original). Significancia estadística.
- Selección de variables (AIC, BIC, p-values - con cautela). Diagnósticos básicos del modelo (residuos).

3. Práctica (Python):

- Statsmodels: Introducción a la librería para modelado estadístico.
- Formulación de modelos usando la sintaxis de R-style formulas.
- Ajuste de un GLM Poisson (`sm.GLM` con `family=sm.families.Poisson()`).
- Análisis de la salida del modelo (`summary()`).
- Predicción de frecuencias para nuevas observaciones o segmentos.
- Validación cruzada simple (concepto y aplicación básica).

4. Material:

Dataset preparado para modelado de frecuencia.



4:// Modelado de Primas II - GLM para Severidad y Prima Pura

1. Objetivos:

Modelar la severidad de siniestros usando GLMs y combinar ambos modelos para calcular la prima pura.

2. Teoría:

- Desafíos del modelado de severidad: Asimetría positiva, presencia de ceros (si se modela sobre pólizas), outliers de alto impacto.
- Distribución Gamma para severidad. Varianza, función de enlace (Log). Supuestos.
- Otras distribuciones posibles (Log-Normal, Tweedie - mención).
- Modelado de la Prima Pura: Prima Pura = Frecuencia Esperada * Severidad Esperada.
- Segmentación y Tarificación: Cómo los coeficientes del modelo se traducen en factores de tarifa.

3. Práctica (Python):

- Ajuste de un GLM Gamma (sm.GLM con `family=sm.families.Gamma(link=sm.families.links.log())`).
- Análisis e interpretación de resultados del modelo de severidad.
- Combinación de predicciones de frecuencia y severidad para obtener la prima pura por póliza/segmento.
- Comparación de primas puras modeladas vs. observadas por segmento.
- Visualización de factores de tarifa derivados del modelo.

4. Material:

Dataset preparado para modelado de severidad (solo siniestros > 0).



5:// Modelado de Primas III - Técnicas Avanzadas (Machine Learning) I

1. Objetivos:

Introducir modelos de Machine Learning como alternativa o complemento a los GLMs en pricing.

2. Teoría:

- Limitaciones de los GLMs (linealidad, interacciones manuales).
- Introducción a modelos de ML: Árboles de Decisión, Random Forests, Gradient Boosting Machines (GBM) como LightGBM o XGBoost.
- Ventajas (captura de no linealidades e interacciones) y Desventajas (interpretabilidad, "caja negra").
- Conceptos clave: Overfitting, Regularización, Validación Cruzada (más robusta).
- Métricas de evaluación para modelos de regresión (RMSE, MAE) y clasificación/conteo (Deviance, AUC - si se reformula).

3. Práctica (Python):

- Scikit-learn: Introducción a la librería estándar de ML en Python.
- Preparación de datos para modelos de ML (encoding, scaling si es necesario).
- Ajuste de un modelo de Árbol de Decisión (DecisionTreeRegressor). Visualización del árbol.
- Ajuste de un Random Forest (RandomForestRegressor). Entendiendo sus hiperparámetros básicos.
- Uso de train_test_split y cross_val_score para evaluación.
- Comparación inicial de desempeño predictivo vs GLM.

4. Material:

Mismo dataset de pricing, preparado para Scikit-learn.



6:// Modelado de Primas IV - Técnicas Avanzadas (ML) II e Interpretabilidad

1. Objetivos:

Profundizar en GBMs y abordar el desafío de la interpretabilidad en modelos de ML.

2. Teoría:

- Gradient Boosting Machines (GBM): Intuición detrás del boosting. Ventajas en rendimiento predictivo.
- Importancia de la Interpretabilidad en Seguros: Explicar tarifas a reguladores y clientes, entender el comportamiento del modelo.
- Técnicas de Interpretabilidad:
 - Feature Importance (global).
 - Partial Dependence Plots (PDP) y Individual Conditional Expectation (ICE) plots (para entender el efecto marginal de una variable).
 - SHAP (SHapley Additive exPlanations): Explicaciones locales y globales consistentes (muy innovador y potente).

3. Práctica (Python):

- Ajuste de un modelo GBM (e.g., LGBMRegressor de LightGBM). Tuning básico de hiperparámetros.
- Cálculo y visualización de Feature Importance.
- Implementación y visualización de PDP/ICE plots.
- SHAP: Introducción a la librería shap. Cálculo y visualización de SHAP values (summary plot, dependence plots, force plots para explicaciones individuales).
- Comparación final de modelos (GLM vs ML) en términos de predicción e interpretabilidad.

4. Material:

Librerías lightgbm y shap.



7:// Portfolio Management I - Monitorización y KPIs

1. Objetivos:

Utilizar Python para calcular, visualizar y monitorear los KPIs clave de una cartera de seguros no vida.

2. Teoría:

- Introducción al Portfolio Management: Objetivos (Rentabilidad, Crecimiento, Volatilidad). El ciclo actuarial.
- KPIs Fundamentales:
 - ▶ Primas (Emitida, Devengada).
 - ▶ Siniestralidad (Pagada, Incurrida - caso cerrado vs IBNR simplificado).
 - ▶ Loss Ratio (Simple, Calendar Year, Accident Year).
 - ▶ Frecuencia y Severidad Media (monitoreo).
 - ▶ Combined Ratio (simplificado).
 - ▶ Ratio de Renovación, Tasa de Caída (Churn).
- Importancia de la segmentación en el monitoreo (por producto, canal, región, añada/generación).
- Triángulos de desarrollo de siniestros (Run-off triangles) - Concepto básico y visualización.

3. Práctica (Python):

- Polars:
 - ▶ Cálculo de KPIs a partir de datos transaccionales (pólizas, siniestros, recibos). Agregaciones temporales (mensual, trimestral, anual). Uso de `pivot_table`.
 - ▶ Cálculo de Primas Devengadas (método pro-rata simplificado).
 - ▶ Visualización de tendencias de KPIs a lo largo del tiempo (Line plots).
 - ▶ Creación de tablas resumen de KPIs por segmento.
 - ▶ Visualización de triángulos de desarrollo (pagados/incurridos) usando `matplotlib` o `seaborn` (heatmap). (Mencionar librerías como `chainladder` - python como recurso avanzado).

4. Material:

Dataset transaccional simulado a lo largo de varios periodos.



8:// Portfolio Management II - Análisis de Rentabilidad, Impacto y Síntesis

1. Objetivos:

Realizar análisis más profundos de la cartera, simular impactos de acciones de pricing y consolidar el aprendizaje.

2. Teoría:

- Análisis de Rentabilidad por Segmento: Identificación de nichos rentables y problemáticos. Análisis de causas raíz (¿frecuencia?, ¿severidad?, ¿tarifa inadecuada?).
- Análisis de Impacto (What-if): Simulación del efecto de cambios de tarifa en la prima media, volumen de primas y siniestralidad esperada del portafolio.
- Optimización de Precios (Introducción): Concepto de elasticidad de la demanda, trade-off entre precio y volumen/retención. (Enfoque conceptual, la implementación es compleja).
- El Feedback Loop: Cómo los insights del Portfolio Management informan las decisiones futuras de Pricing y Suscripción.
- Creación de Dashboards (Concepto): Cómo estructurar los análisis para reportes dinámicos (mencionar herramientas como Dash/Streamlit o Power BI/Tableau alimentados por Python).

3. Práctica (Python):

- Aplicación de modelos de pricing (GLM o ML) a la cartera actual para comparar prima técnica vs prima cobrada por segmento.
- Identificación de segmentos con mayor/menor rentabilidad técnica. Visualizaciones (waterfall charts conceptuales).
- Simulación de un cambio de tarifas (+X% general o por segmento) y recálculo de GWP y Loss Ratio esperado.
- Discusión de un caso de estudio integrador: Desde el análisis de datos hasta la propuesta de acciones de pricing/portfolio y su posible impacto.
- Resumen del curso, próximos pasos, recursos adicionales.

4. Material:

Consolidación de los datasets y modelos de sesiones anteriores.