



# MODELO PREDICTIVO PROYECCIÓN DE VENTAS

Modelos Predictivos

Prof. Juan Marcos Castillo, PhD

Est. Miguel Valzania



# Mi motivación personal

La pasión por descubrir patrones ocultos en datos relacionados con ventas para transformar o entregar los mismos a las unidades negocio y gerenciales para facilitar decisiones estratégicas de valor.



## Curiosidad analítica

Búsqueda constante de respuestas en los datos



## Impacto real

Aplicar machine learning a problemas cotidianos



## Crecimiento profesional

Desarrollar habilidades avanzadas de análisis

# La búsqueda de los datos perfectos

## El desafío

Encontrar dataset completo del ciclo de venta

## La exploración

Revisión de múltiples fuentes hasta encontrar en Kaggle

## El hallazgo

5.000 transacciones con 24 variables

## La conexión

Datos similares a mi trabajo diario



# Lecciones del análisis descriptivo

78%

Artículos de oficina

Del total de ventas

\$1M+

Copiadoras

Solo dos productos

2

Meses pico

Junio y julio cada año

# Mi proceso analítico paso a paso

## Análisis descriptivo

- Exploración inicial
- Visualización por categoría
- Análisis de correlaciones
- Patrones estacionales

```
# Creamos columnas temporales
df['periodo'] = df['Order Date'].dt.year
df['mes'] = df['Order Date'].dt.month
df['periodo_mes'] = df['Order Date'].dt.strftime('%Y-%m')

# Análisis de ventas por categoría
categoria_ventas = df.groupby('Product Category')['Total'].sum().reset_index()
categoria_ventas = categoria_ventas.sort_values('Total', ascending=False)
print("\nCategorías de productos más vendidos:")
print(categoria_ventas)

# Visualización
plt.figure(figsize=(10, 6))
ax = sns.barplot(x='Total', y='Product Category', data=categoria_ventas)
plt.title('Categorías de Productos Más Vendidos')
```

1

2

```
# Preprocesamiento
df = pd.get_dummies(df, columns=['Ship day', 'Order day', 'Customer Type',
                                'Account Manager', 'Order Priority',
                                'Product Category', 'Product Container', 'Ship Mode'])

# Normalización de variables numéricas
scaler = StandardScaler()
variables_numericas = ['Cost Price', 'Retail Price', 'Profit Margin', 'Order Quantity',
                       'Sub Total', 'Discount', 'Total Discount', 'Order Total',
                       'Shipping Cost', 'Total']
df[variables_numericas] = scaler.fit_transform(df[variables_numericas])

# Entrenamiento del modelo
X = df.drop('Total', axis=1)
y = df['Total']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Evaluación
y_pred = model.predict(X_test)
r2 = r2_score(y_test, y_pred)
mape = mape_porc(y_test, y_pred)
print(f'R-cuadrado: {r2:.4f}')
print(f'Error Porcentual Absoluto Medio (MAPE): {mape:.2f}%')
```

## Análisis predictivo

- Limpieza de datos
- Selección de características
- Random Forest Regressor
- Evaluación con métricas
- Series temporales



# Descubrimientos de la Investigación





# Si pudiera empezar de nuevo...



## Variables externas

Indicadores económicos y tendencias



## Segmentación fina

Análisis por categoría de producto



## Modelos

Especializados



## Más historia

Datos para ciclos más largos



# Lo que descubrí de mí mismo

## Mi curiosidad es mi brújula

Preguntas con respuestas por descubrir

## Conexión datos-decisiones

Busco impacto, no solo números

## Paciencia analítica

Proceso metódico revela insights profundos

## Traductor de complejidad

Algoritmos a recomendaciones prácticas



# Thank You!

