# Regresión Logística con Python

Por: Angel Francisco Hernández Gámez

# Introducción

La regresión logística es una técnica estadística y de aprendizaje automático ampliamente utilizada para resolver problemas de clasificación. A diferencia de la regresión lineal, cuyo objetivo es predecir valores continuos, la regresión logística está diseñada para predecir la probabilidad de ocurrencia de un evento categórico. Esta técnica es particularmente útil cuando se busca clasificar observaciones en dos o más categorías discretas, tales como "sí/no", "aprobado/reprobado", o, como en este caso, "Windows/Mac/Linux".

La regresión logística modela la probabilidad de que una observación pertenezca a una clase particular utilizando una función logística o sigmoide, que transforma una combinación lineal de variables independientes en un valor entre 0 y 1:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Donde:

$- z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$ es la combinación lineal de las variables explicativas,
$- \sigma(z)$ representa la probabilidad de que una observación pertenezca a una clase positiva.

En el caso multiclase, se utiliza la función softmax:

$$P(y = k \mid x) = \frac{e^{\beta_k^T x}}{\sum_{j=1}^{k} e^{\beta_j^T x}}$$

donde $k$ es el número de clases posibles y $\beta_k$es el vector de coeficientes para la clase $k$.

Este modelo es valorado por su simplicidad, interpretabilidad y eficiencia computacional, especialmente en contextos donde las relaciones entre variables predictoras y categorías son lineales.

# Metodología

1. Importación de bibliotecas y carga del conjunto de datos.
2. Exploración y visualización de los datos.
3. Definición de variables predictoras (X) y objetivo (y).
4. Entrenamiento del modelo de regresión logística multinomial.
5. Evaluación del modelo con métricas estándar.
6. Validación cruzada.
7. Predicción con datos nuevos.

Fragmento de código:

```
    from sklearn.linear_model import LogisticRegression
    model = LogisticRegression(multi_class='multinomial',
solver='lbfgs')
    model.fit(X_train, y_train)

    y_pred = model.predict(X_test)

    from sklearn.metrics import confusion_matrix,
classification_report
    print(confusion_matrix(y_test, y_pred))
    print(classification_report(y_test, y_pred))

    from sklearn.model_selection import cross_val_score
    scores = cross_val_score(model, X, y, cv=5)
```

# Resultados

- Precisión en datos completos: 77%
- Precisión media (validación cruzada): 74%
- Precisión en conjunto de validación: 85%

Métricas por clase:
- Precision: 0.86
- Recall: 0.84
- F1-score: 0.84

El modelo fue capaz de predecir correctamente el sistema operativo de un usuario ficticio, mostrando aplicabilidad práctica.

# Conclusión

La regresión logística demostró ser efectiva para predecir la clase de sistema operativo utilizada por los usuarios según su comportamiento en el sitio web.

El modelo alcanzó resultados satisfactorios incluso con un conjunto de datos reducido. Además, la validación cruzada ayudó a garantizar su capacidad de generalización. Se sugiere, para trabajos futuros, explorar otras técnicas más complejas, aplicar regularización y ajustar hiperparámetros para mejorar la precisión del modelo.