Clasificación_

Sesión Presencial 1

Alcances de la unidad asignada

***Alcance de la sesión**:*

- *Conocer la regresión logística y sus fundamentos.*
- *Conocer y ser capaz de interpretar estadísticos de bondad de ajuste y coeficientes.*
- *Reconocer los supuestos en que tiene sustento teórico.*
- *Implementar un modelo de regresión con `statsmodels`.*
- *Implementar un modelo predictivo con `scikit-learn`.*
- *Conocer los conceptos de validación cruzada y medidas de desempeño.*

Activación de Conceptos

- En la unidad anterior aprendimos sobre los fundamentos básicos del machine learning y cómo implementar un modelo básico.
- ¡Pongamos a prueba nuestros conocimientos!
- Las preguntas van en subslides.

¿Cómo funciona el flujo de trabajo en Machine Learning?

1. Hacemos un algoritmo y posteriormente generamos predicciones

- Preprocesamos los datos y subdividimos, generamos un modelo, lo refactorizamos y comparamos métricas.
- Es lo mismo que el modelo econométrico.

¿Por qué preferimos trabajar con métricas de error por sobre métricas de explicación?

1. Porque las métricas de error se enfocan en la reducción de error en las predicciones.

- No importa con qué métrica se trabaja, el resultado es el mismo.
- Porque las métricas de explicación se enfocan en la optimización.

¿Cómo importamos `sklearn`?

1. No hay una forma única de importar `sklearn`, dado que necesitamos especificar los elementos a utilizar.

- `import scikit-learn as sklearn`.
- `from scikit-learn import sklearn`.

¿Qué podemos decir sobre el comportamiento de la curva de aprendizaje?

1. En la medida que la cantidad de atributos de una matríz aumenta, el desempeño del training set empeora.

- Los desempeños de training y testing convergen en la medida que el tamaño muestral aumenta.
- El desempeño de testing set tendrá una forma funcional cuadrática.

{desafío} latam_

Regresión Logeistica (Desde la Econometría)

{desafío}
latam_

Problema

- La regresión logística busca despejar sobre los determinantes de un fenómeno discreto, con dos categorías.

- Ejemplos:
    - Movimientos del mercado: *¿Bajará o subirá la bolsa?* $\rightsquigarrow Y_i \in \{0 : Baja, 1 : Sube\}$.
    - Clasificación de Spam: *¿Es este mail Spam o No?* $\rightsquigarrow Y_i \in \{0 : No, 1 : Sí\}$.
    - Optimización de Preferencias: *¿Es más probable votar o no?* $\rightsquigarrow Y_i \in \{0 : NoVota, 1 : Vota\}$

- Estos se pueden aproximar mediante una distribución binomial.

{desafío} latam_

Modelo de Probabilidad Lineal

- Regresión lineal donde asumimos que los parámetros estimados miden la probabilidad de ocurrencia de y_i.

$$\mathrm{y}_i = \beta_0 + \beta_1 \cdot \mathrm{X} + \varepsilon_i$$

Problemas:

1. $x_i\beta \mapsto (-\infty, \infty)$. Los parámetros no tienen límites.

- $\varepsilon_i \sim \frac{\pi^2}{3}$. El error no sigue una distribución normal.
- Los parámetros no son monotónicamente lineales.

{desafío} latam_

Regresión logística

- Mientras que la regresión lineal se optimiza mediante la reducción de los errores cuadráticos, la regresión logística se optimiza mediante el método de máxima verosimilitud.

- Para este caso, modelamos los parámetros como **lineales en los log-odds**.

$$\log\left(\frac{p(y)}{1-p(xy)}\right) = \beta_0 + \beta_1 \times \mathbb{X}$$

- El problema es que los log-odds no son muy intuitivos. Para ello podemos reescalarlos a la probabilidad mediante la función logística inversa:

$$\text{logit}^{-1}(x_i\beta) = \frac{\exp(x_i\beta)}{1+\exp(-x_i\beta)}$$

{desafío} latam_

Ejemplo: Pozos de Bangladesh (Gelman y Hill, 2007)

Modelo LPM

$$y_i = \beta_0 + \beta_1 \cdot \texttt{dist100} + \varepsilon_i$$

In [2]:
```
ols_fit = smf.ols('y ~ dist100', df).fit()
```

Modelo Logit

$$\log\left(\frac{p(y)}{1-p(y)}\right) = \beta_0 + \beta_1 \times \texttt{dist100}$$

In [3]:
```
logit_fit = smf.logit('y ~ dist100', df).fit()
```

```
Optimization terminated successfully.
         Current function value: 0.674874
         Iterations 4
```

Interpretación de bondad de ajuste

```
In [4]: print(logit_fit.summary2().tables[0])
```

```
                    0                 1                   2
3
0               Model:             Logit  Pseudo R-squared:        0.01
0
1  Dependent Variable:                 y               AIC:    4080.237
8
2                Date:  2018-10-30 17:00               BIC:    4092.263
9
3    No. Observations:              3020    Log-Likelihood:       -2038.
1
4            Df Model:                 1           LL-Null:       -2059.
0
5        Df Residuals:              3018       LLR p-value:  9.7978e-1
1
6           Converged:            1.0000             Scale:       1.000
0
7      No. Iterations:            4.0000
```

Interpretación de coeficientes

In [5]:
```
print('OLS\n',ols_fit.summary2().tables[1].loc[:, 'Coef.':'Std.Err.'],"\n")
print('Logit\n',logit_fit.summary2().tables[1].loc[:, 'Coef.':'Std.Err.'],"\n" )
```

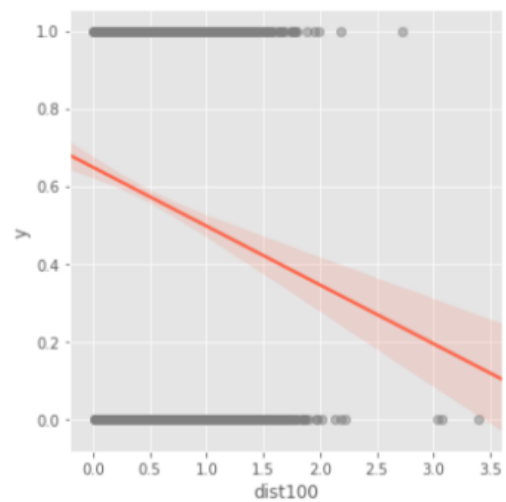
```
OLS
                Coef.  Std.Err.
Intercept  0.648407  0.014347
dist100   -0.151539  0.023225

Logit
                Coef.  Std.Err.
Intercept  0.605959  0.060310
dist100   -0.621882  0.097426
```

{desafío}
latam_

Interpolación $\mathbf{x_i}\boldsymbol{\beta}$ en LPM

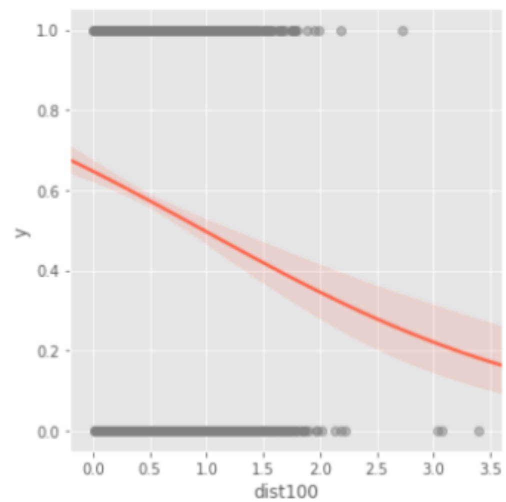
In [6]:
```
sns.lmplot('dist100', 'y', df, line_kws={'color':'tomato'}, scatter_kws={'color':'grey',
'alpha': .5});
```



{desafío}
latam_

Interpolación $\mathrm{logit}^{-1}(x_i\beta)$

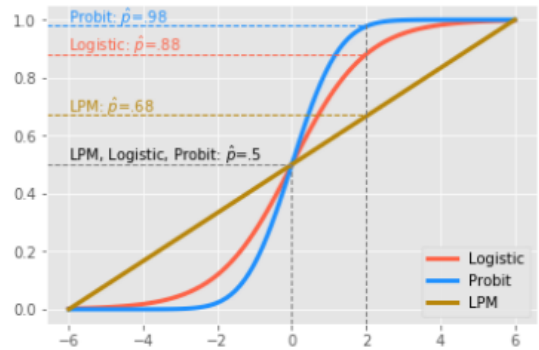
In [7]:
```
sns.lmplot('dist100', 'y', df, logistic=True, line_kws={'color':'tomato'}, scatter_kws={
'color':'grey', 'alpha': .5} );
```



{desafío}
latam_

Relación entre Logit y LPM

In [8]:
```
gfx.logit_probit_lpm()
```



{desafío}
latam_