

**Maestría en Ciencia y Análisis de Datos- Universidad Mayor de San Andrés**  
**Machine learning y Deep learning**

---

NOMBRE: Lucy Gabriela Cuarita Ajno

INFORMACION DE CONTACTO:

Correo electrónico: [lcuarita@fcpn.edu.bo](mailto:lcuarita@fcpn.edu.bo)

Número de teléfono celular (opcional): 72577567

---

La práctica tiene un valor de 30 puntos. Utilice una semilla igual a **248**.

**PRACTICA 1**

**Ajuste los hiperparametros del modelo de machine learning prophet (MLP), compare los resultados y escoja el mejor modelo MLP:**

1. Haga una partición de datos de 87.5% en la muestra de entrenamiento para dejar como hold-out los datos de 2024 y entrenar el modelo con datos hasta 2023 (5 puntos)

**Solución.**

- *Partición de datos de 87.5% en la muestra de entrenamiento*

```
tamano_muestra_entrenamiento = 0.875 # 87.5%, hold-out en 2024  
train_size <- floor(tamano_muestra_entrenamiento * nrow(df))  
train <- df[1:train_size, ]  
test <- df[(train_size + 1):nrow(df), ]
```



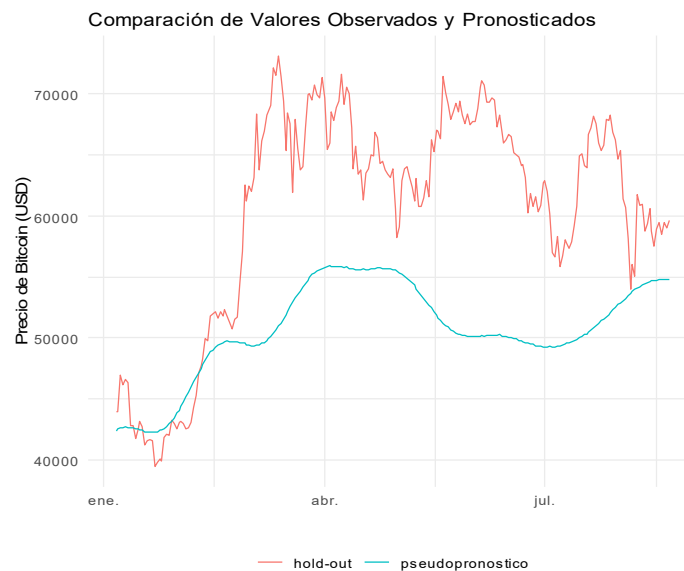
**Conclusión:** La partición de entrenamiento son los datos del bitcoin hasta el periodo 2023 y la partición de prueba considera los datos del periodo 2024.

- Entrenando los datos con el modelo Prophet, para luego predecir.

```
set.seed(248)
modelo_profeta <- prophet(train)

# ----- Predicciones con el modelo Prophet -----
predicciones <- make_future_dataframe(modelo_profeta, periods = nrow(test))
pseudopronostico <- predict(modelo_profeta, predicciones)
```

Los resultados se muestran gráficamente como se muestra:



**Conclusión:** La paseudo predicciones con el modelo prophet en el primer trimestre del 2024 siguen la tendencia de los datos de prueba, sin embargo apartir del mes de abril se observa una subestimación de los datos predichos con el modelo.

2. Entrene dos modelos prophet con los siguientes hiperparametros (5 puntos):

```
modelo_profeta_chp20 <- prophet(train, n.changepoints = 20)
modelo_profeta_chp05 <- prophet(train, n.changepoints = 5)
```

### Solución.

Entrenando a los dos modelos prophet solicitados y calculando las pseudo predicciones, se tiene el siguiente gráfico:

```
# Entrenar el modelo prophet
set.seed(248)
modelo_profeta_chp20 <- prophet(train, n.changepoints = 20 )
modelo_profeta_chp05 <- prophet(train, n.changepoints = 5 )

# ----- Predicciones con el modelo Prophet -----
```

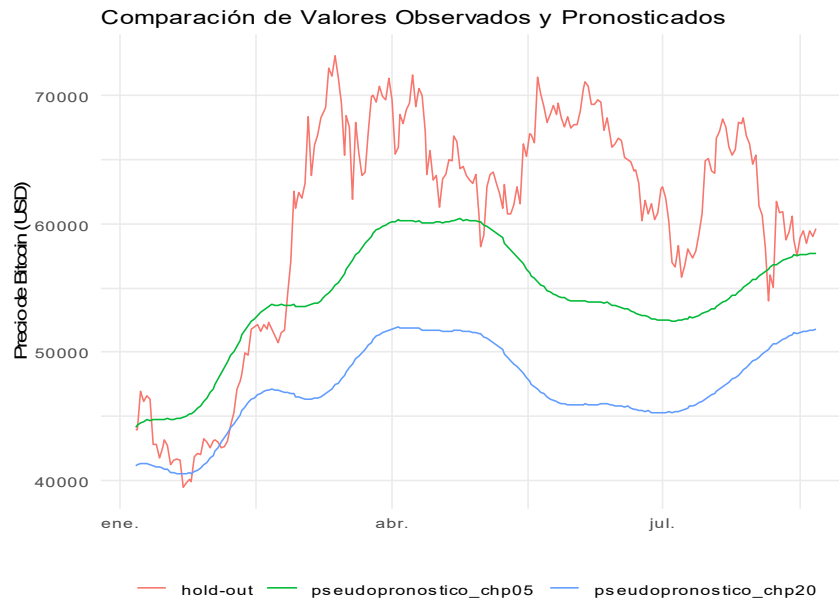
# Realizar predicciones

```
predicciones_chp20 <- make_future_dataframe(modelo_profeta_chp20, periods = nrow(test))
```

```
pseudopronostico_chp20 <- predict(modelo_profeta_chp20, predicciones_chp20)
```

```
predicciones_chp05 <- make_future_dataframe(modelo_profeta_chp05, periods = nrow(test))
```

```
pseudopronostico_chp05 <- predict(modelo_profeta_chp05, predicciones_chp05)
```



**Conclusión:** La incorporación de los puntos de cambio determina una mejora en las predicciones más propiamente en el caso cuando se utilizan 5 puntos de cambio (línea verde). Si bien en el primer trimestre las predicciones siguen la tendencia de los datos a partir del mes de abril se subestiman los mismos.

3. Compare los modelos MLP utilizando el error cuadrático medio (ECM) y el error absoluto medio (EAM) de las predicciones, ¿cuál modelo tiene el menor ECM y EAM? (5 puntos)

**Solución.** La tabla muestra los resultados de calcular el ECM y el MAE de los tres modelos prophet propuestos:

Modelo prophet	ECM	MAE
Sin puntos de cambio	137154353	9940.088
5 puntos de cambio	<b>76519893</b>	<b>7327.933</b>
20 puntos de cambio	223666172	13103.24

**Conclusión:** Los datos de la tabla muestran que el modelo prophet con 5 puntos de cambio es el mejor de las tres propuestas puesto que es el que tiene valores de ECM y MAE mínimos.

**Entrene bosques aleatorios y máquinas de soporte vectorial para predecir casos de enfermedades cardíacas. Utilice una partición de datos con 70% de datos en la muestra de entrenamiento.**

1. Entrene un modelo SVM con función de base radial (5 puntos):

**Solución:** Los resultados del entrenamiento se resumen en la siguiente matriz de confusión:

Test / pred_svm	0	1
0	43	4
1	9	34

**Conclusión.** La precisión de casos clasificados correctamente es:  $\frac{77}{90} \approx 0.856$

2. Entrene un modelo de bosques aleatorios con 100 árboles (5 puntos):

**Solución:** Los resultados del entrenamiento se resumen en la siguiente matriz de confusión:

Test / pred_rf	0	1
0	42	5
1	10	33

**Conclusión.** La precisión de casos clasificados correctamente es:  $\frac{75}{90} \approx 0.833$

3. Compare la exactitud (accuracy) de los modelos SVM y de bosques aleatorios, ¿cuál modelo tiene la mayor exactitud (accuracy)? (5 puntos)

**Solución:** Los datos de la tabla muestran la precisión calculada luego de aplicar support vector machine y random forest.

Algoritmo	Precisión
Support Vector Machine	0.856
Random Forest	<b>0.833</b>

**Conclusión.** Los resultados muestran que el Algoritmo Random Forest es ligeramente mejor que Support Vector Machine.