

León Felipe Serna Montoya



**UNIVERSIDAD
DE ANTIOQUIA**
1 8 0 3

Fundamentos de deep learning

Informe final

Energy forecasting

Docente: Raúl Ramos Pollan

Contenido

1.	Contexto de la aplicación	2
2.	Objetivo.....	3
3.	Dataset	3
	Disponibilidad y acceso a los datos.....	3
4.	Estructura de los notebooks	4
5.	Descripción de la solución.....	4
6.	Resultados.....	6
7.	Referencias.....	8

1. Contexto de la aplicación

El pronóstico de demanda energética es el proceso por el cual es la cantidad de energía que consume un área geográfica y eléctricamente conectada en un período de tiempo dado. El pronóstico de demanda energética es importante para la estabilidad del sistema eléctrico, ya que, la frecuencia depende del balance entre la generación y la demanda, razón por la cual los centros de despacho de energía monitorean la frecuencia 24/7. En caso de grandes desbalances se podría producir un apagón del sistema (*blackout*).

En Colombia, XM es la empresa encargada del pronóstico de la demanda [1]. Actualmente, XM posee modelos para dicha estimación, sin embargo, los modelos deben ser actualizados debido a la alta penetración de generación distribuida que es dada por fuentes de alta volatilidad, como la generación solar fotovoltaica y eólica.

Concluyendo, este proyecto pretende construir un modelo que permita estimar la demanda basado en históricos de demanda.

La siguiente imagen ilustra una curva de demanda.



Figure 1 curva de demanda ejemplo [2].

2. Objetivo

Queremos predecir la demanda futura con base en los datos históricos de los registros de demanda. Es decir, los rezagos de la demanda histórica permiten estimar la demanda futura, desde un periodo $t-i$ para un periodo posterior t .

$$y_t = \text{demanda [MW]}$$

$$y_{t-i} = \text{demanda [MW]}$$

El dataset contiene la demanda de cada hora y la fecha en la cual es realizada cada medida, esta información podría ayudar a mejorar el desempeño de la estimación del modelo a implementar.

El dataset contiene variables exógenas, como temperaturas y de efecto calendario, estas variables podrían ser consideradas para mejorar el pronóstico de demanda energética.

3. Dataset

La información proviene de la etapa de clasificación de la GEFCom2017. Esta serie de datos cuenta con la información real publicada por el ISO New England (ISONE), donde se presentan datos de demanda, temperatura, y algunos precios de energía, para las ocho zonas que abarca ISONE. El conjunto de datos incluye 15 años de información, desde el año 2003 al 2017. Para el análisis son tomados cerca de dos años y medio de datos de demanda de tiempo real [MW] del estado ISONE, además de la dry-bulb temperature [F°] y la dewpoint temperature [F°].

Fuente: [Energy forecasting](#) [3].

Tamaño en disco: 19.1 MB

Cantidad de datos: 43824

Disponibilidad y acceso a los datos.

Los datos se encuentran almacenados en mi github personal y para acceder a ellos he usado los siguientes comandos, los cuales permiten acceder al contenido del repositorio, los comandos y estructura son análogos a lo usado en el curso.

https://github.com/leonidasf300/dl22_2/tree/main/content/local/data

```
!wget -nc --no-cache -O init.py -  
q https://raw.githubusercontent.com/leonidasf300/dl22_2/main/content/i  
nit.py  
import init; init.init(force_download=False);
```

4. Estructura de los notebooks

La estructura de los notebooks entregados es la siguiente:

- **02 LS Preprocesado.ipynb:** este notebook contiene el procesamiento de la información base que contiene como insumo un archivo de MS Excel (.xlsx) y entrega como producto un archivo CSV para ser procesado posteriormente en los modelos implementados. Dentro de la información procesada son codificados de forma booleana los días de la semana, clasificando si hay días festivos y si tenemos festivos cercanos. Adicionalmente, información estadística como lo son medias y desviaciones estándar.
- **03 LS ArquitecturaLineaBase.ipynb:** este notebook recibe como insumo el archivo procesado en el notebook mencionado anteriormente, en el cual son evaluados varios modelos y analizados los resultados.

5. Descripción de la solución

El objetivo de este proyecto es pronosticar la demanda de energía, la primera parte es la exploración de los datos, la Figure 2 ilustra un diagrama de cajas y bigotes, mostrando los valores típicos de demanda en cada periodo. Los periodos mostrados en el eje horizontal corresponden un rango horario, de este modo el periodo cero va desde las 0:00 hasta las 0:59 y así sucesivamente por cada periodo. Adicionalmente, las variables: festivo, dlf y dfnl son graficadas, que muestran la cantidad de días festivos (festivo), los días festivos lunes (dlf) y los días festivos que no son lunes (dfnl). El eje vertical corresponde a la demanda, note que los valores son normalizados para evitar el desbordamiento numérico. En el análisis de las curvas de demanda los días feriados afectan la demanda de energía eléctrica, razón por la cual son hallados los días feriados y los días cercanos a ellos. Este proyecto es complementario al proyecto realizado por el ingeniero Aarón Casadiegos, quien está enfocado en la construcción de un modelo estadístico de pronóstico de demanda, esta parte del proyecto es enfocada a la implementación de redes neuronales profundas, como resultado hay código que comparten ambas partes del proyecto.

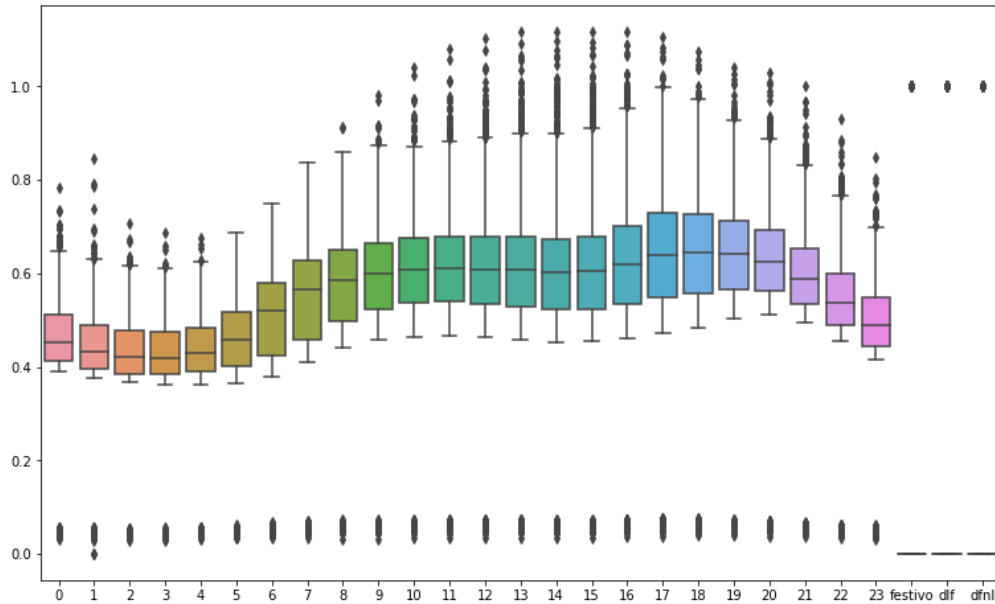


Figure 2 diagrama de cajas y bigotes curva de demanda.

La Figure 3 ilustra el conjunto de datos de prueba, es decir, son los datos reales de la demanda, el objetivo es pronosticar la demanda con valores cercanos a esta curva. Los datos graficados son de demanda versus un ID de secuencia.

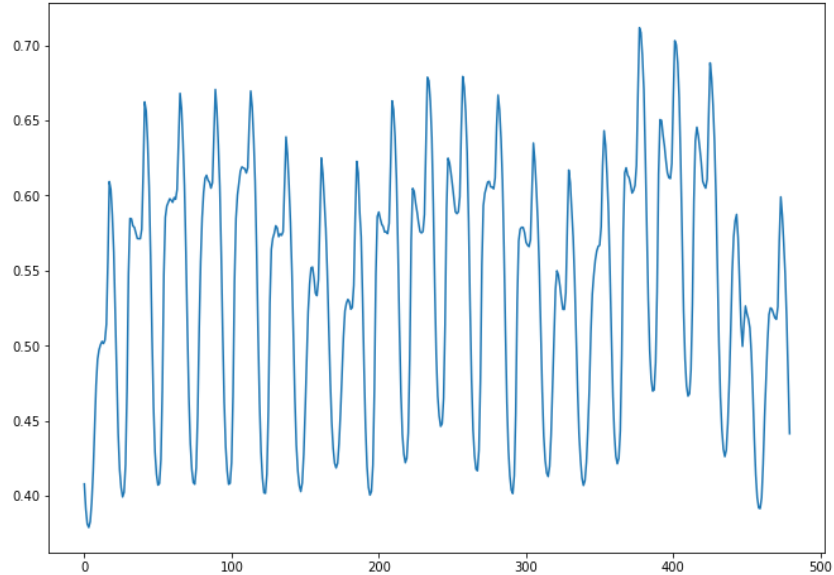


Figure 3 conjunto de datos de prueba.

Los datos de entrenamiento y prueba son abordados desde el enfoque tradicional, como es mencionado en el notebook 5.0 Intro time series.ipynb, conservando los primeros valores para entrenamiento y los últimos para prueba, en el caso particular son considerados alrededor de dos años y medio de entrenamiento, conservando los últimos 20 días para pruebas. Los dataframe:

df_data_e y df_data_p, contienen la información para entrenamiento y pruebas respectivamente. Note que algunos modelos trabajan con rezagos y análisis de pronóstico de demanda, razón por la cual son reservados los primeros valores para ser los rezagos del test de entrenamiento, por ende en el código es definido un offset de partida, considerando los valores previos.

```
FORECAST_DATE_INIT = datetime.datetime(2015,1,1)
TRAINING_DATE_END = FORECAST_DATE_INIT + datetime.timedelta(days=565+31-15)
HNTE = 22*24
NDHT = 15*24
zonal = df0.copy()

zonal = zonal.rename_axis('fecha').reset_index() # Se resetea el indice para convertir
a diccionario mas facil

colum=zonal.columns[1:]
df_data= zonal.iloc[HNTE:].reset_index(drop=True)
df_data_e = df_data.loc[:len(df_data)-NDHT-
1] #El menos 1 es debido a que comienza en cero
df_data_p = df_data.loc[len(df_data)-NDHT:]
```

6. Resultados

La línea base es evaluada con un modelo clásico lassoCV, el cual es parametrizado como se muestra en la siguiente imagen.

```
from sklearn.linear_model import LassoCV, Lasso
from sklearn.metrics import mean_absolute_percentage_error, mean_squared_error

lasso_reg = LassoCV(cv=10, random_state=20221112, alphas=np.arange(0.5,10,0.5))
lasso_reg.fit(X=X_train.values, y=y_train.values)

pron_realista_lasso_reg = {} # pronotico realista
pron_realista_lasso = {} # pronotico realista
demanda=df_data_e['Dem'].to_dict()

    # for t,i in enumerate(range(len(demanda),len(demanda)+NDHT)):
X_test_c = X_test.copy()
for i in X_test_c.index:
    d=lasso_reg.predict(X_test_c.loc[i].to_numpy().reshape(1,-1))[0]
    #print(d,i)
```

```

pron_realista_lasso_reg[i]=d
for r in resagos:
    if i+r > len(demanda)+NDHT:
        break
    else:
        X_test_c.loc[i+r,'r_'+str(r)]=d # pone el nuevo resultado de la demanda pr
onosticada en el rezago que le corresponde para cada hora y poder hacer el pronostico
con este.

```

El resultado del modelo LassoCV es mostrado en la Figure 4 en el cual es mostrada la y de prueba (Real values) y los datos predichos por el modelo (LassoCV).

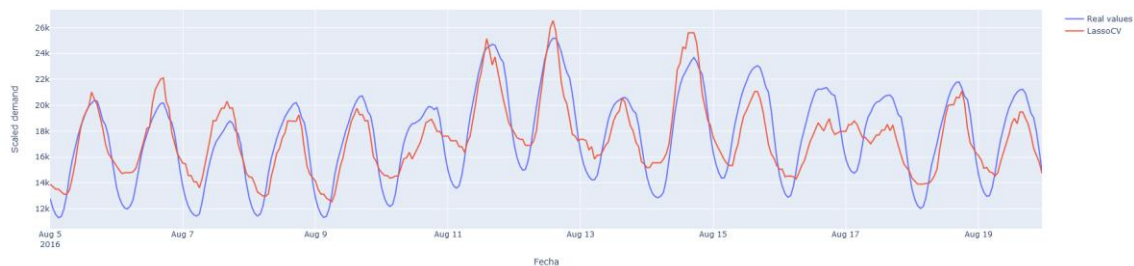


Figure 4 resultado modelo LassoCV.

La Figure 5 muestra el resultado del pronóstico de la red neuronal para el modelo de la red neuronal, note que los valores fueron transformados a sus valores originales por medio del scaler. La red neuronal empleada es una red de dos capas, una con 5 neuronas con activación relu y la capa de salida con una neurona de salida, sin activación por ser un regresión. La red es parametrizada y los datos formateados para dos pasos hacia atrás.

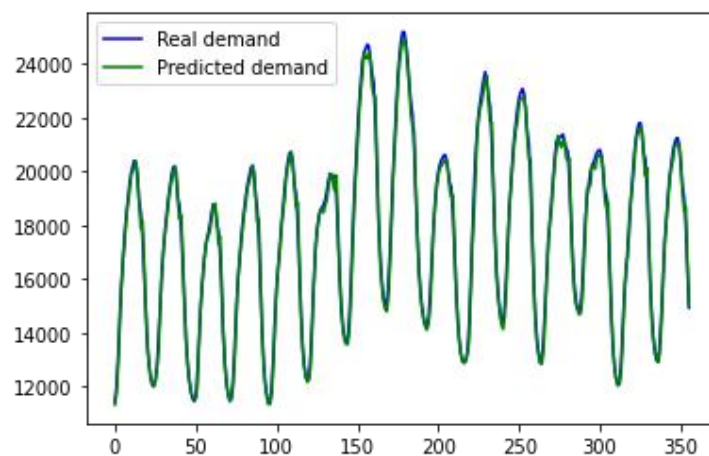


Figure 5 resultado red neuronal.

En términos generales podemos usar métricas de regresión para comparar los resultados del pronóstico. La métrica de desempeño más usada en la literatura científica y la industria es Error Porcentual Absoluto Medio (MAPE). A continuación, la siguiente tabla presenta el resumen de las métricas de desempeño.

Modelo	MAPE	RMSE
LassoCV	9.2	1752.78
NN	1.43	377.35

La métrica de desempeño MAPE y RMSE nos indican un mejor resultado para la red neuronal, cabe mencionar que el modelo Lasso está sometido a predecir un modelo sobre los datos pronosticados, mientras que en la red neuronal fue usada la función `DataPreparationRNN.split_sequence()` para dar formato a los datos.

7. Referencias

- [1] "Pronóstico de demanda," *Portal XM*. <https://www.xm.com.co>Portal XM (accessed Dec. 11, 2022).
- [2] "Proyección de la demanda eléctrica: clasificación, variables y métodos empleados," *dheybicervan*, Apr. 19, 2017. <https://dheybicervan.com/proyeccion-de-la-demanda-electrica/> (accessed Dec. 11, 2022).
- [3] "GEFCom2017: Hierarchical Probabilistic Load Forecasting." <http://blog.drhongtao.com/2016/10/gefcom2017-hierarchical-probabilistic-load-forecasting.html> (accessed Dec. 11, 2022).