PAP II - Modelos de predición en empresas y gobierno mediante aprendizaje estadístico.

Juan Francisco Muñoz Elguezábal
Alumno Ingeniería Financiera
IF149833@iteso.mx

Rodrigo Ledesma Elorriaga
Alumno Ingeniería Financiera
IF683439@iteso.mx

Zurisadai Velazquez Manzanero
Alumno Ingeniería Financiera
IF677633@iteso.mx

Este proyecto integrador es desarrollado como documento del Proyecto de Aplicación Profesional, actividad donde se incluye la realización del servicio social y la opción de titulación para programas de licenciatura en la universidad ITESO (Instituto Tecnológico y de Estudios Superiores de Occidente). Realizado para alumnos de la licenciatura Ingeniería Financiera de la generación 2011-2015.

Índice general

Resumen

Capítulo 1

Introducción

En este capítulo...

Sección 1.1 **Objetivos**
Sección 1.2 **Justificación**
Sección 1.3 **Antecedentes**
Sección 1.4 **Contexto**
Sección 1.5 **Enunciado**

1.1. Objetivos

- **Principal:**

  Aplicar metodologías de machine learning a problemas de diversas industrias productivas y de servicios, con la finalidad de construcción de modelos para la clasificación y/o pronóstico de variables de particular importancia como las ventas, incumplimiento de pagos a crédito, etc.

- **Secundarios (de conocimiento):**

  - Utilizar metodología *Boosted trees*.
  - Utilizar metodología *Random Forest*.
  - Utilizar metodología *Natural Language Processing*.

- **Secundarios (de herramientas):**

  - Utilizar eficientemente R and Python Statistical Programming.
  - Desarrollar todo código con Git Version Control.

Regresar a: Capítulo 1 (Introducción)

1.2. Justificación

La ciencia de datos en la actualidad representa una herramienta de nueva generación. Tanto para los que buscan soluciones a problemas cotidianos mediante el uso y procesamiento adecuado de datos, como para los que buscan precisamente convertirse en científicos de datos. La programación estadística es una herramienta que hace posible efectuar estudios y ser un profesionista en la ciencia de datos, para realizar programación estadística ciertamente existe un gran número de herramientas disponibles, en partícular lenguajes de programación, en este PAP y documento mostramos el uso de *R* y *Python*.

Regresar a: Capítulo 1 (Introducción)

1.3. Antecedentes

1.4. Contexto

1.5. Enunciado

Regresar a: Capítulo 1 (Introducción)

Capítulo 2

Desarrollo

En este capítulo...

Sección 2.1 **Sustento teórico y metodológico**
Sección 2.2 **Planeación y seguimiento**
Sección 2.3 **Resultados**
Sección 2.4 **Reflexiones de aprendizajes**

2.1. Sustento teórico y metodológico

Se utiliza aprendizaje computacional o *Machine Learning* como metodología para procesar datos y encontrar patrones de comportamiento. Hacer uso de recursos computacionales en esta época y desde la perspectiva de ingeniería financiera es lo que hace característico a este Proyecto de Aplicación Profesional. En concreto se utilizaron técnicas de aprendizaje supervisado. Todas de métodos *tradicionales* o los que **No** son *heurísticos*.

2.2. Planeación y seguimiento

Se inició con el método de árboles de decisión para tener una metodología básica previa a las demás, a pesar de que esta es ampliamente utilizada es también considerada como la inicial y susceptible de mejoras y optimización. Posteriormente se desarrolla e implementa el método conocido como *Boosted Trees* o *árboles aumentados o estimulados*.

2.3. Resultados

Regresar a: Capítulo 2 (Desarrollo)

2.4. Reflexiones de aprendizajes

2.4.1. Implicaciones éticas

2.4.2. Aportaciones sociales

Regresar a: Capítulo 2 (Desarrollo)

Capítulo 3

Herramientas computacionales

En este capítulo...

3.1. Linux

3.2. Sistema de control de versiones Git

3.3. Programación estadística en R

3.4. Programación estadística en Python

Regresar a: Capítulo 3 (Herramientas computacionales)

Capítulo 4

Boosted Trees - Ponpare

En este capítulo...

4.1. Descripción del problema

Regresar a: Capítulo 4 (Boosted Trees - Ponpare)

4.2. Exploración los datos

Regresar a: Capítulo 4 (Boosted Trees - Ponpare)

4.3. Construcción de modelo

Regresar a: Capítulo 4 (Boosted Trees - Ponpare)

4.4. Desempeño de modelo

Regresar a: Capítulo 4 (Boosted Trees - Ponpare)

Capítulo 5

Random Forest - Rossman

En este capítulo...

5.1. Descripción del problema

Objetivo: Prononsticar las ventas de 1115 farmacias de la cadena Rossman localizadas por todo Alemania. Utilizando información y datos provenientes de puntos de venta, promociones y datos de competidores.

Rossmann opera más de 3.000 farmacias en 7 países europeos. Los Gerentes de las tiendas Rossmann tienen la tarea de predecir sus ventas diarias para las próximas seis semanas de operación. Naturalmente se puede pensar que las ventas en tienda son influenciadas por muchos factores, incluyendo las promociones, la competencia, la dinámica social como periodos escolares, los días festivos estatales e incluso las estaciones del año.

Regresar a: Capítulo 5 (Random Forest - Rossman)

5.2. Exploración los datos

5.2.1. Básica inicial

En los datos de *Entrenamiento* se tiene un número de tiendas de *1,115* de las cuales se recabaron datos desde el *2013-01-01* al *2015-07-31*. Así mismo al realizar la exploración se ha encontrado que el mayor número de ventas registradas en una tienda fue de *41,551*, esto en la tienda con ID: *909*. Por el contrario, la tienda que menos ventas registró fue la correspondiente al ID: *652* debido a que se registran sólo *46*.

5.2.2. Ajuste de datos

La información de fechas en las columnas *Date* y *Store* se encuentran almacenadas como un objeto tipo *Factor*, el siguiente codigo es para convertirlas a individuales, tanto para el conjunto de entrenamiento, *train*, como para el de prueba *test*. También se hace una transformación logarítmica a la cifra de ventas, con la finalidad de reducir el efecto de los datos atípicos que de manera no formal se encontraron en la exploración inicial.

```
ChapRFtrain[,Date:= as.Date(Date)]
ChapRFtest[ ,Date:= as.Date(Date)]

ChapRFtrain[,month:= as.integer(format(Date, "%m"))]
ChapRFtrain[,year:=  as.integer(format(Date, "%y"))]
ChapRFtrain[,Store:= as.factor(as.numeric(Store))]

ChapRFtest[,month:= as.integer(format(Date, "%m"))]
ChapRFtest[,year:=  as.integer(format(Date, "%y"))]
ChapRFtest[,Store:= as.factor(as.numeric(Store))]

ChapRFtrain[,logSales:=log1p(Sales)]
```

5.2.3. Uso de libreria H2O

Transcripción de documentación oficial: *R scripting functionality for H2O, the open source math engine for big data that computes parallel distributed machine learning algorithms such as generalized linear models, gradient boosting machines, random forests, and neural networks (deep learning) within various cluster environments*

Lo que significa que se utilizará para efectuar *Parallel Distributed Machine Learning Algorithms* o Algoritmos de Aprendizaje Computacional y Distribución Paralela.

5.2.4. Inicializar Cluster

```
h2o.init(nthreads=-1, max_mem_size='6G')
trainHex <- as.h2o(ChapRFtrain)
```

5.2.5. Establecer variables y entrenar modelo

```
features <- colnames(ChapRFtrain)[!(colnames(ChapRFtrain)
%in% c("Id","Date","Sales","logSales","Customers"))]

rfHex <- h2o.randomForest(x=features,y="logSales",ntrees=100,max_depth=30,
nbins_cats = 1115,training_frame=trainHex)
summary(rfHex)
```

5.2.6. Datos de prueba

```
testHex <- as.h2o(ChapRFtest)
predictions <- as.data.frame(h2o.predict(rfHex,testHex))
pred <- expm1(predictions[,1])
summary(pred)
```

5.2.7. Resultado Final

```
submission <- data.frame(Id=test$Id, Sales=pred)
```

Regresar a: Capítulo 5 (Random Forest - Rossman)

5.3. Construcción de modelo

Regresar a: Capítulo 5 (Random Forest - Rossman)

5.4. Desempeño de modelo

Regresar a: Capítulo 5 (Random Forest - Rossman)

Capítulo 6

Natural Language Processing - Word2Vec

En este capítulo...

6.1. Descripción del problema

Regresar a: Capítulo 6 Natural Language Processing - Word2Vec

6.2. Exploración los datos

- Cuenta en Twitter.

```
username = '@iffranciscome'
```

- Registro de APP.

```
https://bitbucket.org/quant-ai/twittermining
```

- Generar llaves.

```
consumer_key    = '4TJeb1mqGw22VmxWd8w7gf3Tk'
consumer_secret = 'uvwY6vtdfxzbsbUOS14sHQMfZgKXOcRyAi7041XbdEkZWVKXhc'
access_token    = '3288299311-sk9rkLFG0bXoeZbVbV4mJN7lmLCuUqweMhkO7BV'
access_token_secret = 'FhGj0ab2j7b7UN5LLjgeZ3ZBeTyCrkt0T6dGe6IeYAoVj'
```

Cargar Librerías

```
import tweepy as tp # API para Twitter https://github.com/tweepy/tweepy
import numpy  as np #
import pandas as pd # Manejo de datos
import json   as js # Manejo de datos tipo JSON
import nltk
import re

from nltk.tokenize import word_tokenize   # Importar funcion tokenizar.
from nltk.corpus import stopwords         # Importar lista "stop word".
from bs4 import BeautifulSoup             # Extraer datos de archivos HTML y XML.
from sklearn.feature_extraction.text import CountVectorizer
```

Autentificacion y Usuarios

```
auth = tp.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api  = tp.API(auth)

Periodicos = {'Nombre' : ['Wall Street Journal','Bloomberg Markets','CNBC'],
              'Cuenta' : ['@WSJmarkets','@markets','@CNBC']}

Profesion  = {'Nombre' : ['Guillermo Barba','Jim Cramer','Tom Keene'],
              'Cuenta' : ['@memobarba','@jimcramer','@tomkeene']}

Bolsas     = {'Nombre' : ['Dow Jones','BMV','CME Group'],
              'Cuenta' : ['@DowJones','@GrupoBMV','@CMEGroup']}

DFPer = pd.DataFrame(Periodicos)
DFPro = pd.DataFrame(Profesion)
DFBol = pd.DataFrame(Bolsas)
```

Peticiones de información

```
DFPer['Seguidores'] = 0
DFPer['Seguidores'][0] = len(api.followers_ids(DFPer['Cuenta'][0]))
DFPer['Seguidores'][1] = len(api.followers_ids(DFPer['Cuenta'][1]))
DFPer['Seguidores'][2] = len(api.followers_ids(DFPer['Cuenta'][2]))

DFPro['Seguidores'] = 0
DFPro['Seguidores'][0] = len(api.followers_ids(DFPro['Cuenta'][0]))
DFPro['Seguidores'][1] = len(api.followers_ids(DFPro['Cuenta'][1]))
DFPro['Seguidores'][2] = len(api.followers_ids(DFPro['Cuenta'][2]))

DFBol['Seguidores'] = 0
DFBol['Seguidores'][0] = len(api.followers_ids(DFBol['Cuenta'][0]))
DFBol['Seguidores'][1] = len(api.followers_ids(DFBol['Cuenta'][1]))
DFBol['Seguidores'][2] = len(api.followers_ids(DFBol['Cuenta'][2]))

guarda = api.user_timeline(DFPer['Cuenta'][0])[1].created_at

Tweets0 =  [t.text for t in api.user_timeline(DFPer['Cuenta'][0])]
Tweets1 =  [t.text for t in api.user_timeline(DFPer['Cuenta'][1])]
Tweets2 =  [t.text for t in api.user_timeline(DFPer['Cuenta'][2])]

Tweets3 =  [t.text for t in api.user_timeline(DFPro['Cuenta'][0])]
Tweets4 =  [t.text for t in api.user_timeline(DFPro['Cuenta'][1])]
Tweets5 =  [t.text for t in api.user_timeline(DFPro['Cuenta'][2])]

Tweets6 =  [t.text for t in api.user_timeline(DFBol['Cuenta'][0])]
Tweets7 =  [t.text for t in api.user_timeline(DFBol['Cuenta'][1])]
Tweets8 =  [t.text for t in api.user_timeline(DFBol['Cuenta'][2])]
```

Construcción de vocabulario

```
review_text = BeautifulSoup(Tweets0[1]).get_text()
letters_only = re.sub("[^a-zA-Z]", " ", review_text)
words = letters_only.lower().split()
stops = set(stopwords.words("english"))
meaningful_words = [w for w in words if not w in stops]
```

Regresar a: Capítulo 6 Natural Language Processing - Word2Vec

6.3. Construcción de modelo

Regresar a: Capítulo 6 Natural Language Processing - Word2Vec

6.4. Desempeño de modelo

Regresar a: Capítulo 6 Natural Language Processing - Word2Vec

Capítulo 7

Algorithmic Trading - Oanda

En este capítulo...

7.1. Descripción del problema

Regresar a: Capítulo 7 Algorithmic Trading - Oanda

7.2. Exploración los datos

Regresar a: Capítulo 7 Algorithmic Trading - Oanda

7.3. Construcción de modelo

Regresar a: Capítulo 7 Algorithmic Trading - Oanda

7.4. Desempeño de modelo

Regresar a: Capítulo 7 Algorithmic Trading - Oanda

Capítulo 8

Anexos y Bibliografía

En este capítulo...

Sección 7.1 **Descripción del problema**
Sección 7.2 **Exploración los datos**

8.1. Anexos

API oficial de twitter para conectividad directa.

- Pagina Oficial Desarrolladores
- Documentacion general
- Documentacion REST APIs
- Consola oficial para pruebas

Regresar a: Sección 8.1 Anexos

8.2. Bibliografía

Regresar a: Sección 8.1 Anexos