Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2115 – Programación como Herramienta para la Ingeniería (II/2022)

Laboratorio 2

1. Aspectos generales

- **Objetivo:** evaluar individualmente el aprendizaje sobre análisis y visualización de datos en Python, a través de la construcción de una serie de tareas asociadas las ventas de tiendas de *retail*.
- **Lugar de entrega:** jueves 6 de octubre a las 23:59 hrs. en repositorio privado.
- **Formato de entrega:** el archivo Python Notebook (**L2.ipynb**) con la solución del laboratorio. El archivo debe estar ubicado en la carpeta **L2**. Es requerimiento de formato el utilizar múltiples celdas de texto y código para la construcción de la solución. Laboratorios que no cumplan el formato de entrega tendrán un descuento de 0,2 décimas de la nota final.
- **Entregas atrasadas:** El descuento por atraso se realizará de acuerdo a lo definido en el programa del curso. Si su laboratorio es entregado fuera de plazo, tiene hasta el **viernes 7 de octubre a las 11:59 AM** para responder el formulario de **entregas fuera de plazo** disponible en el Syllabus.
- **Issues:** Las discusiones en las *issues* del Syllabus que sean relevantes para el desarrollo del laboratorio, serán destacadas y se considerarán como parte de este enunciado. Así mismo, el uso de librerías externas que solucionen aspectos fundamental del laboratorio no podrán ser utilizadas. Solo se podrán utilizar las que han sido aprobadas en las *issues*, previa consulta de los estudiantes.
- **Laboratorios con errores de sintaxis y/o que generen excepciones en todas las ejecuciones serán calificados con nota 1.0.**

2. Introducción

En este laboratorio utilizará dos conjuntos de datos que recopilan información de diferentes tiendas de *retail* y registro de las ventas que han hecho por día. Su objetivo es utilizar estos datos para hacer distintos tipos de predicciones: clasificación de tipo de tienda y regresión para predecir las ventas de una tienda.

2.1. Archivo `store_data.csv`

En el archivo tiene la información general de las tiendas y cuenta con las siguientes columnas:

- **Store**: identificador único de la tienda.
- **StoreType**: tipo de tienda. Toman los valores `a`, `b`, `c` y `d`.
- **Assortment**: describe la variedad de productos que tiene la tienda, a = básico, b = general y c = extendido.
- **CompetitionDistance**: distancia en metros de la tienda que es competencia más cercana.
- **Promo2**: indica con un 1 si la tienda está participando de una promoción y 0 si no.

2.2. Archivo `sales_data.csv`

En el archivo tiene los registros por día de las ventas y clientes que han visitado las tiendas.

- **Store**: identificador único de la tienda.
- **Sales**: cantidad total de ventas en un día.
- **Customers**: cantidad total de clientes en un día.
- **Date**: fecha del registro.

En base a los campos recién descritos y utilizando las librerías presentadas en clases, deberá cumplir una serie de misiones relacionadas con el análisis de datos en Python.

3. Misiones

3.1. Carga y exploración de los datos (3 ptos)

Cargue los datos contenidos en el archivo `store.csv` y en sales.csv en un `DataFrame`, obtenga los tipos de datos de cada columna de cada archivo y consulte algunos estadísticos generales con los métodos revisados en clases. A continuación, presente visualizaciones relevantes para las columnas, agregando un párrafo de comentarios para cada una, analizando los estadísticos observados y caracterizando la distribución.

Por último, identifique los elementos nulos y outliers. Decida qué hacer con estos elementos y no olvide justificar su respuesta.

3.2. Generar `Dataframe` para la predicción (3 ptos)

Utilizando la información de ambas tablas, haga una combinación de ellas para tener los datos unificados en un solo `DataFrame`. Luego, aplique una operación de agregación por cada tienda que indique el promedio de ventas y clientes. Para el resto de las columnas, debe determinar qué operación de agregación usar. Justifique su respuesta. Para cerrar, genere dos `DataFrame`, uno para entrenamiento y otro de prueba. La división debe ser de 80 a 20.

3.3. Clasificación: predecir `StoreType` (3 ptos)

En esta sección sólo debe usar las siguientes columnas como atributos para predecir el tipo de tienda: `CompetitionDistance`, `Promo2`, `Sales`, `Customers`.

Para comenzar, haga una reducción a 2 dimensiones utilizando PCA para poder graficar graficar los datos de entrenamiento en un *Scatter Plot* y además, el color de cada punto debe corresponder al tipo de tienda. Hint: busque en Google cómo mapear colores a una variable categórica en un *Scatter Plot*.

Luego, debe entrenar 3 modelos de clasificación y evalúe la *Accuracy* de cada uno. Genere las predicciones de los datos de prueba y evalúe la *Accuracy* nuevamente más la matriz de confusión para cada modelo. Discuta el comportamiento de entrenamiento y prueba de cada modelo; determine cuál tiene mejor rendimiento.

3.4. Regresión: predecir `Sales` (3 ptos)

En esta sección sólo debe usar las siguientes columnas como atributos para predecir el tipo de tienda: `CompetitionDistance`, `Promo2`, `Customers`, `StoreType`, `Assortment`.

Notará que `StoreType` es una variable categórica, o sea, no toma valores numéricos. Su primer objetivo es explorar la documentación de *Scikit Learn* para encotrar un método para trabajar con variables categóricas en modelos predictivos de esta librería. Explique su elección.

Luego, debe entrenar 3 modelos de regresión y evalúe el error cuadrático medio (MSE) de cada uno. Genere las predicciones de los datos de prueba y evalúe su MSE nuevamente. Discuta el comportamiento de entrenamiento y prueba de cada modelo; determine cuál tiene mejor rendimiento.

3.5. Bonus (0,5 décimas)

Para los modelos de regresión, determine la relevancia en la predicción de cada una de las características usadas. Explique cómo lo hizo y su respuesta.

Corrección

Para la corrección de este laboratorio, se revisarán los procedimientos desarrollados para responder las diferentes misiones propuestas y la estructura de como utiliza los módulos *pandas*, *matplotlib*, *numpy* y/o *sklearn* en ellos. Dado lo abierto de las misiones, se espera que las respuestas incluyan análisis y visualizaciones que permitan justificar las decisiones tomadas.

Política de Integridad Académica

Los alumnos de la Escuela de Ingeniería deben mantener un comportamiento acorde al Código de Honor de la Universidad:

"Como miembro de la comunidad de la Pontificia Universidad Católica de Chile me comprometo a respetar los principios y normativas que la rigen. Asimismo, prometo actuar con rectitud y honestidad en las relaciones con los demás integrantes de la comunidad y en la realización de todo trabajo, particularmente en aquellas actividades vinculadas a la docencia, el aprendizaje y la creación, difusión y transferencia del conocimiento. Además, velaré por la integridad de las personas y cuidaré los bienes de la Universidad."

En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un procedimiento sumario. Ejemplos de actos deshonestos son la copia, el uso de material o equipos no permitidos en las evaluaciones, el plagio, o la falsificación de identidad, entre otros. Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica en relación a copia y plagio: Todo trabajo presentado por un alumno (grupo) para los efectos de la evaluación de un curso debe ser hecho individualmente por el alumno (grupo), sin apoyo en material de terceros. Si un alumno (grupo) copia un trabajo, se le calificará con nota 1.0 en dicha evaluación y dependiendo de la gravedad de sus acciones podrá tener un 1.0 en todo ese ítem de evaluaciones o un 1.1 en el curso. Además, los antecedentes serán enviados a la Dirección de Docencia de la Escuela de Ingeniería para evaluar posteriores sanciones en conjunto con la Universidad, las que pueden incluir un procedimiento sumario. Por "copia" o "plagio" se entiende incluir en el trabajo presentado como propio, partes desarrolladas por otra persona. Está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, siempre y cuando se incluya la cita correspondiente.