Parcial Final
Curso: Introducción a HPC
Docente: John Jairo Corredor Franco.
Regresión Lineal
Fecha: 23 - Noviembre - 2022

Parcial Final

Chepe K[1],
[1] Ciencias de la Computación e Inteligencia Artificial, Universidad Sergio Arboleda, Bogotá DC, Colombia..

Resumen: *Esta práctica tiene como fin el poder replicar las técnicas aprendidas en clase para implementar regresión lineal, el desarrollo se llevará a cabo de manera manual desde Qt utilizando lenguaje de programación C++ y desde Google Collaboratory haciendo uso de módulos como Sklearn utilizando el lenguaje de programación Python*

Summary: *This practice is intended to replicate the techniques learned in class to implement linear regression, the development will be carried out manually from Qt using C++ programming language and from Google Collaboratory using modules such as Sklearn using the Python programming language.*

I. Introducción

El planteamiento de un conjunto de datos de prueba tomado de Kaggle que recoge los ingresos por ventas generadas con respecto a los costes de la publicidad en múltiples canales como la radio, la televisión y los periódicos, para llevar a cabo el experimento de aplicar regresión lineal sobre dos lenguajes de programación diferentes.

The approach of a test data set taken from Kaggle that collects sales revenue generated with respect to advertising costs across multiple channels such as radio, television and newspapers, to conduct the experiment of applying linear regression on two different programming languages.

II. Objetivos

- Modelar las predicciones basadas en la Regresión Lineal
  - Seleccionar un dataset.
  - Hacer una análitica de datos sobre el dataset seleccionado
  - Modelar usando la regresión lineal usando: Python, Scikit-Learn
  - Modelar usando la regresión lineal usando: C++
  - Comparar los modelos

III. Ejecución

Para el modelo de Python en Google Collab se utilizaron los módulos:

- Sk-learn: para la creación del modelo, la separación de los datos, las métricas de rendimiento
- Pandas: Para manejar el dataset y convertirlo a una variable manejable
- Numpy: Para realizar operaciones aritméticas.
- Matplotlib y Seaborn: Para graficar.

Se importó el dataset desde un repositorio de github y se trabajó con el módulo de pandas para poder trabajarlo de manera adecuada, se procedió a hacer un análisis de los datos y generar gráficas de matrices de dispersión y correlación con matplotlib y seaborn para ver qué tan relacionadas están unas columnas con otras, se halló el promedio y desviación estándar de todas las columnas para posterior su comparación y se procedió a dividir el conjunto de datos en datos de entrenamiento y datos de prueba con el módulo Sklearn para su uso, se aplicó regresión lineal y halló que cual fue su rendimiento con la métrica R2.

Para el modelo de C++ en Qt se hizo de manera algo manual apoyándose de herramientas como:

- Eigen: Para facilitar el uso de matrices en c++
- Módulos heredados de C como Vector, list.
- fstream: para manejar el dataset.

En este modelo se creó un proyecto en el cual se crearon diferentes clase como lo son:

- ClassExtraction: Donde se crearon funciones para abrir el dataset csv, para convertir el dataset a matrices trabajables desde eigen, sacar la normalización, el promedio, la desviación estándar y hacer la separación de datos simulando un traintestsplit de python.

UNIVERSIDAD SERGIO ARBOLEDA

Parcial Final

Curso: Introducción a HPC
Docente: John Jairo Corredor Franco.
Regresión Lineal
Fecha: 23 - Noviembre - 2022

- LinearRegression: Donde se crearon funciones como: de costo, gradiente descendiente y la métrica de rendimiento R2

IV. Comparativa

```
Metrica R2 train: 0.901771
```

Figura 1. Rendimiento modelo C++

```
Metrica de R2 Score:  0.8765948711365155
```

Figura 1. Rendimiento modelo Python

V. Análisis de los resultados

- Como se puede apreciar en los resultados de las métricas de rendimiento de cada uno de los modelos, el modelo de C++ es un poco más preciso que el de Python, pero aun así son resultados bastante buenos los dos muy similares.

VI. Conclusión

- Del dataset seleccionado se puede concluir que es un dataset muy simple pero muy práctico para casos en los que se tenga que aplicar regresión lineal ya que no presenta variaciones grandes,
- El rendimiento es demasiado bueno, al ser un dataset algo pequeño puede que no tenga los suficientes datos para determinar una regresión lineal óptima.
- No se encontraron muchos datos atípicos lo cual quiere decir que es un caso perfecto para la aplicación de modelos como la regresión lineal.
- No hubo ni un solo dato vacío ni errado, lo que quiere decir que es un dataset enfocado a la experimentación.
- Los valores hallados del promedio y la desviación estándar entre los dos modelos son muy cercanos, dando a entender que el modelo manual está bien hecho, y el desfase es probablemente por cómo maneja los tipos de datos el lenguaje utilizado en cuestión.
- La mayor correlación se presenta entre la variable TV y ventas esto debido a que el mercado de televisores es bastante estable y las ventas siempre siguen un promedio.
- Aunque el proceso para realizar regresión lineal fue mucho más sencillo en Python ya que tiene módulos específicos que se encargan de esta, se pierde un poco de rendimiento ya que el ajuste de datos está generado en un estándar y no puede ser retocado como si se puede en el modelo de C++.

VII. Referencias

- Minitab Blog Editor. (s. f.). Análisis de Regresión: ¿Cómo Puedo Interpretar el R-cuadrado y Evaluar la Bondad de Ajuste? https://blog.minitab.com/es/analisis-de-regresion-como-puedo-interpretar-el-r-cuadrado-y-evaluar-la-bondad-de-ajuste
- Eigen. (s. f.). https://eigen.tuxfamily.org/index.php?title=Main_Page
- GeeksforGeeks. (2022, 23 agosto). Python | Linear Regression using sklearn. https://www.geeksforgeeks.org/python-linear-regression-using-sklearn/
- González, J. D. M. (2020, 21 febrero). Estructura de un Programa. https://www.programarya.com/Cursos/C++/Estructura
- sklearn.impute.SimpleImputer. (s. f.). scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.impute.SimpleImputer.html