**Reporte- Taller 3 - Modelos de aprendizaje en python**

Analítica Computacional para la toma de Decisiones

IIND-4130

Código: 201915999 Código: 201714797

Nombre: Mateo Rodríguez Salazar Nombre: Diego Rojas Marinkelle

1. **Realice un análisis exploratorio y resuma (comentarios breves, preciso, enumerados) en su reporte:**
2. Comportamiento individual de cada característica y de la variable de respuesta.

Tabla 1. Comportamiento individual variables.

|  |  |
| --- | --- |
|  | |
| Codigo: data.head() | X1: Fecha  X2: Años  X3: Distancia  X4: Accesibilidad a tiendas  X5: Longitud  X6: Latitud  Y: Precio de la vivienda por área |

|  |  |
| --- | --- |
|  | |
| Codigo: data.describe() | X1: en promedio la última compra/venta de los inmuebles fueron en el 2013  X2: En la base de datos hay inmuebles totalmente nuevos hasta inmuebles de 45 años  X3: En promedio las personas tienen que caminar 1000 metros a la estación más cercana esto es alrededor de 30 minutos de caminata  X4: La mayoría de los inmuebles tienen una buena accesibilidad a servicios de óseo  X5,6: Los puntos se encuentran concentrados en una misma zona geográfica dado a la desviación estándar de la latitud y longitud  Y: La variabilidad entre precios es muy alto |

|  |  |
| --- | --- |
| Codigo: sns.pairplot(data,) | |
|  | X1: La mayor frecuencia es en los últimos meses del 2013  X2: La mayoría de inmuebles tienen 15 años  X3: No se encuentran distancias mayores a 2Km de las estaciones de Transmilenio  X4: Hay homogenidad en los datos de accesibilidad  X5,6: La zona de análisis tiene una densidad alta  Y: Se evidencia una distribución de Poisson |

1. Correlaciones entre características y con la variable de respuesta.

Tabla 2. Correlaciones

|  |  |
| --- | --- |
| **Código:** corr = data.corr()  sns.heatmap(corr, cmap="Blues", annot=True) | |
|  | Las variables descriptivas con una correlación significativa con la variable predictora (Y) son: X4: 0,57  X5: 0,55  X6: 0,52    Por otro lado, las correlaciones de las variables X1, X2 y X3 son considerablemente bajas. Esto, indica de manera preliminar que estas variables pueden no ser significativas para el modelo. |

1. **Exploración bivariada entre cada característica y la variable de respuesta**

Tabla 3. Exploración bivariada.entre cada característica y la variable de respuesta.

|  |  |
| --- | --- |
|  | |
| Codigo: sns.pairplot(data, x\_vars=['X1 transaction date','X2 house age','X3 distance to the nearest MRT station','X4 number of convenience stores','X5 latitude','X6 longitude'], y\_vars='Y house price of unit area', height=7, aspect=0.7, kind='reg') | Para todas las variables se evidencia una alta variabilidad en los datos, no se evidencia una tendencia clara en esta primera etapa de análisis. No obstante, la tendencia de ciertos conjuntos de datos tienen un menor impacto sobre la variable predictiva. Los conjuntos mencionados son “X1 transaction date” y “X2 house age”, las cuales tienen una relación de baja pendiente con la variable de costo de vivienda por área inversamente proporcional y directamente proporcional, respectivamente.  Por otra parte, las demás variables poseen impactos más pronunciados. Para el caso de las variables “X4 number of convenience stores”, “X5 latitude” y “X6 longitude” se establecen relaciones directamente proporcionales, mientras que la variable “X3 distance to the nearest MRT Station” presenta una relación inversamente proporcional. |

Tabla 4. Mapa de ubicación

|  |  |
| --- | --- |
| **Código:** import folium  import pandas as pd  first\_location = data.iloc[0]  # Crear el mapa base  m = folium.Map(location=[first\_location['X5 latitude'], first\_location['X6 longitude']], zoom\_start=12)  # Añadir un marcador para cada ubicación en el DataFrame  for idx, row in data.iterrows():  folium.Marker([row['X5 latitude'], row['X6 longitude']]).add\_to(m)  # Mostrar el mapa  m | |
|  | Los inmuebles se encuentran situados en el país Taiwán, específicamente en la ciudad de Taipéi y están altamente concentrados en el sur-este de la ciudad. |

1. **Cree un modelo lineal que permita predecir la variable de respuesta a partir de las características. En su reporte resuma y comente:**
2. Métricas del modelo usando datos de entrenamiento.

A partir del análisis del problema se determina que se busca estimar el costo de vivienda por unidad de área, por esta razón se segrega el conjunto de datos inicial en un conjunto “X” que almacena las variables descriptivas y un conjunto “y” que contiene la variable predictiva. Las variables se muestran a continuación:

X1: Fecha

X2: Años

X3: Distancia

X4: Accesibilidad a tiendas

X5: Longitud

X6: Latitud

Y: Precio de la vivienda por área

Posteriormente, los conjuntos “X” y “y” fueron segregados nuevamente para obtener datos de entrenamiento y prueba del modelo. Para esto se usó la función train\_test\_split de la librería scikit-learn, donde en los parámetros se indicó la semilla para asegurar la replicabilidad de los resultados y también el porcentaje de los datos que conformarían el conjunto de prueba. Esto se logro por medio del código dispuesto a continuación, como resultado se obtiene cuatro conjuntos correspondientes a las variables descriptivas y predictivas para ambas etapas: entrenamiento y prueba.

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, random\_state=1, test\_size=0.2)

Esto permitió la creación del objeto del modelo y el ajuste de parámetros utilizando los datos de entrenamiento, como se muestra a continuación:

linreg = LinearRegression()

linreg.fit(X\_train, y\_train)

Por medio de este conjunto de datos se obtuvo la siguiente ecuación:

Una vez se obtiene el ajuste del modelo se procede a realizar la predicción con los datos de prueba y posteriormente la evaluación de estos, haciendo uso de la librería numpy. Para la predicción se utilizó la función predict con el test de prueba como se muestra en la siguiente línea de código:

y\_pred = linreg.predict(X\_test)

Por otra parte, para la evaluación del modelo se obtuvieron las métricas Error Absoluto Medio (MAE), Error Cuadrático Medio (MSE) y la Raíz del error Cuadrático Medio (RMSE). A continuación, están dispuestos el código implementado y los correspondientes resultados para el modelo.

MAE = metrics.mean\_absolute\_error(y\_test, y\_pred)

MSE = metrics.mean\_squared\_error(y\_test, y\_pred)

RMSE = np.sqrt(MSE)

MAE: 5.343

MSE: 45.011

RMSE: 6.709

Para esta versión preliminar del modelo no es posible comparar los resultados, pero por medio de iteraciones y de análisis estadísticos se espera mejorar los resultados del modelo.

1. Métricas del modelo usando validación cruzada.

Para la obtención de métricas por medio de validación cruzada se utilizo la función cross\_val\_score de la librería scikit\_learn. En este proceso se definió un número de cinco divisiones (folds) para realizar la validación cruzada, donde se ajusta el modelo con cuatro de los cinco folds para hacer la validación con el fold restante. Este procedimiento se realiza para todas las posibles combinaciones. Por otra parte, se define la métrica de error cuadrático medio negativo para minimizar el error cuadrático medio. Esto se debe a que la librería scikit\_learn busca maximizar la métrica definida, pero en este caso buscamos minimizarla. Para esto se implementó el siguiente código que almacena el error cuadrático medio negativo en la variable scores, así mismo se muestran los resultados del error cuadrático medio positivo para cada una de las validaciones realizadas con las diferentes configuraciones de folds, sus respectivas raíces cuadradas (RMSE) y el promedio de los RMSE.

scores = cross\_val\_score(linreg, X, y, cv=5, scoring='neg\_mean\_squared\_error')

Tabla 5. Resultados métricas del modelo validación cruzada

|  |  |  |  |
| --- | --- | --- | --- |
| Configuración | Error Cuadrático Medio (MSE) | Raíz del error Cuadrático Medio (RMSE) | Promedio |
| 1 | 49,898 | 7,063 | 8,693 |
| 2 | 89,029 | 9,436 |
| 3 | 57,866 | 7,607 |
| 4 | 134,824 | 11,611 |
| 5 | 60,054 | 7,749 |

1. Evaluación del modelo y sus parámetros empleando pruebas estadísticas.

Para la evaluación del modelo con pruebas estadísticas fue necesario importar la librería statsmodels y se segregaron nuevamente los sets de entrenamiento y test como se desarrollo en el literal 3.a. Luego, se define un nuevo modelo de regresión, pero en esta ocasión haciendo uso de la librería statsmodels y por medio de los conjuntos de variables “X” y “y” de variables descriptivas y predictivas, respectivamente. Este modelo fue obtenido por medio del método de mínimos cuadrados ordinarios (OLS). El código y los resultados del modelo (figura 1) fueron dispuestos a continuación.

X\_train = sm.add\_constant(X\_train)

model = sm.OLS(y\_train, X\_train).fit()

Texto

Descripción generada automáticamente

Figura 1. Resultados modelo de regresión obtenido con OLS de la librería statsmodels con todas las variables predictivas.

Los principales puntos a destacar de los resultados obtenidos en este modelo son el R2 de 0.551 el cual indica que el ajuste del modelo respecto a los datos de entrenamiento es aceptable. No obstante, se espera mejorar el desempeño del modelo eliminando las variables descriptivas que no son significativas para el modelo. Para esto, se revisó la correlación y los análisis realizados en el punto dos de este informe junto con los resultados del modelo obtenido por medio de la regresión con OLS. Inicialmente se removieron las variables X1, X2 y X3 considerando que su correlación fue bastante baja, sin embargo, los resultados de ajuste bajaron a un R2 de 0.447.

Imagen que contiene Escala de tiempo

Descripción generada automáticamente

Figura 2. Resultados modelo de regresión obtenido con OLS de la librería statsmodels sin X1, X2 y X3.

Por esta razón se realizó un modelo eliminando únicamente la variable X3 que fue la que obtuvo una menor correlación con la variable predictiva, para este se obtuvieron los resultados ilustrados en la figura 3. A pesar de que el R2 incremento a 0.479 el ajuste continúa siendo menor que el modelo que el primer modelo con todas las variables descriptivas.

Texto

Descripción generada automáticamente con confianza media

Figura 3. Resultados modelo de regresión obtenido con OLS de la librería statsmodels sin X3.

Finalmente se realizó un último modelo removimiento en esta ocasión la única variable cuyo estadístico t y la P>|t| no permitían rechazar la hipótesis nula de que el coeficiente es significativo para el modelo, donde se obtuvo un R2 de 0.542 que se aproximó en mayor medida al R2 del modelo original pero continua teniendo un menor ajuste que este. Los resultados están contenidos en la figura 4.

Imagen que contiene Texto

Descripción generada automáticamente

Figura 4. Resultados modelo de regresión obtenido con OLS de la librería statsmodels sin X6.

Por último, se graficó la influencia y se obtuvieron los puntos de alta influencia con distancia de Cook y umbral 4/n. En la gráfica de influencia se puede observar que la gran mayoría de los puntos esta distribuido de manera similar con un studentized residual entre 2 y 4 y un leverage menor a 0.04, sin embargo, existen observaciones que pueden estar influyendo en el desempeño del modelo considerando que su apalancamiento es considerablemente mayor o lo que se refleja en la observación 271 que su tamaño y posición en el eje vertical indican que es la observación con mayor influencia en el modelo y a su vez es la que mayor studentized residual posee indicando mayor desviación respecto al resultado esperado.

Gráfico, Gráfico de dispersión, Gráfico de burbujas

Descripción generada automáticamente

Por ultimo se obtuvo el umbral de Cook y los datos que podrían considerarse outliers con alta influencia considerando esta métrica.

-Umbral con distancia de Cook: 0.0129

Texto

Descripción generada automáticamente

Figura 5. Datos que podrían ser outliers con alta influencia

1. Incluya todo el código de exploración y análisis como soporte.

Se adjunta el código en Bloque Neón y se incluye el link del repositorio:

Repositorio: https://github.com/DiegoRojasMarinkelle/Taller-3-ACTD