DAY-26

OCTOBER-14

On Insurance data, find the following:

Loading the data : insurance_data = pd.read_csv(r"C:\Users\INDU PRIYA\OneDrive\Attachments\Desktop\MLP\insurance.csv")

1. Each region wise total expenses
   insurance_data.groupby(by='region')['expenses'].sum().sort_values(ascending = False)
2. Gender wise average bmi and expenses
   insurance_data.groupby('sex')[['bmi','expenses']].mean().round(2).sort_values(by ='expenses',ascending = False)
3. Each region,each children and smoker class wise total expenses
   insurance_data.groupby(['region','children','smoker'])[['expenses']].sum().sort_values(by='expenses',ascending = False)
   To extract data using pivot table:
   import warnings
   warnings.filterwarnings('ignore')
   pd.pivot_table(data=insurance_data,values='expenses',index='region',columns=['smoker','children'],aggfunc='sum')

NUMPY:

- It is the fundamental library for numerical computing in python.
- It provides fast array operations,mathematical functions,linear algebra,statistics etc
- Best for 2D data like table and matrix.

1.import numpy as np

arr1 = np.array([1,2,3,4,5])

arr2 = np.array([4,5,6,7,8])

print(arr1+arr2)

O/P:

[5 7 9 11 13]

1. What are the roles and responsibilities of data analyst.

   A data analyst plays a full-cycle role from collecting raw data to delivering actionable insights. Their end-to-end responsibilities involve multiple technical and business-oriented phases.

   1. Data Collection and Integration
   - Gather data from diverse sources — databases, APIs, spreadsheets, social media, and cloud systems.
   - Verify dataset reliability, remove redundancies, and integrate disparate data formats into a usable structure.
   - Tools: SQL, Python (Pandas), APIs, ETL pipelines.

   2. Data Cleaning and Preprocessing
   - Handle missing or inconsistent values and format data for analytical readiness.
   - Detect outliers, standardize units, and ensure compliance with data quality standards.
   - Tools: Python (NumPy, Pandas), Excel Power Query.

   3. Data Storage and Database Management
   - Design and maintain logical data models and relational database schemas.
   - Optimize query efficiency and manage updates in database systems.
   - Tools: MySQL, PostgreSQL, MongoDB.

   4. Data Analysis and Pattern Recognition
   - Apply statistical, exploratory, and predictive methods to reveal trends.
   - Perform hypothesis testing, regression, clustering, and correlation analysis.
   - Tools: Python (scikit-learn, statsmodels), R, Excel.

   5. Data Visualization and Reporting
   - Translate analytical insights into visual stories via dashboards and reports.
   - Use interactive visual tools for stakeholder presentations.
   - Tools: Power BI, Tableau, Matplotlib, Seaborn.

   6. Business Interpretation and Communication
   - Collaborate with non-technical teams to interpret analytics in business terms.

- Develop actionable recommendations that directly impact KPIs and strategy.
- Skills: storytelling, stakeholder communication, presentation skills.

7. Performance Monitoring and Optimization

- Continuously track key metrics to detect emerging trends and anomalies.
- Automate monitoring dashboards for real-time data tracking.
- Tools: Power BI alerts, automated Python scripts.

2. Priority wise data type execution and execution time for each datatype.

Python's data type execution priority is based on computational efficiency mostly dependent on internal data structures (mutable vs immutable) and operation type (arithmetic, I/O, string, etc.).

Data Type	**Execution Priority (1=Fastest)**	**Average Execution Time (microseconds per 1M ops)**	**Reason for Speed**
Integer (int)	1	~0.035 µs	Implemented in C with direct CPU arithmetic
Float (float)	2	~0.045 µs	Uses IEEE 754 double precision arithmetic
Boolean (bool)	3	~0.050 µs	Special case of integers; minimal memory footprint
Tuple (tuple)	4	~0.060 µs	Immutable, thus faster hash computation
String (str)	5	~0.075 µs	Interned and immutable, optimized for lookup

List (list)	6	~0.100 µs	Mutable; resizing and memory allocation overhead
Dictionary (dict)	7	~0.120 µs	Hash-based; high lookup speed but expensive creation
Set (set)	8	~0.130 µs	Hash-based like dict; slower due to uniqueness enforcement

Execution Hierarchy Summary:
int < float < bool < tuple < str < list < dict < set