DISHANK KALRA | 9313601825 | https://dishankkalra23.github.io/ | contact.dishankk@gmail.com | https://www.linkedin.com/in/dishankkalra/ | https://github.com/dishankkalra23

EDUCATION

NIIT University Aug 2018 – present

B.Tech, Computer Science and Engineering **8.02 (till semester- 6)**

SKILLS

Mathematics and Statistics*: Descriptive Statistics, Probability, Binomial Distribution, Bayes Rule, Sampling distributions, Central Limit Theorem, Confidence Interval, Hypothesis Testing, A/B Test, Regression*

Programming*: Python (pandas, NumPy), SQL, R, Java*

Visualization*: Tableau, Excel Chart, Matplotlib, seaborn*

Database*: MySQL, PostgreSQL, Microsoft SQL Server*

Effective Communication, Project management, Problem-solving, Research, Teamwork

PROJECTS

Medical Appointment No-Show**,** Data Wrangling, EDA, Data Visualization **May 2021 – June 2021**

- ***Problem Statement* -** Many patients book the appointment with the doctor and then failed to attend scheduled appointments. The average No-Show is **20%** leading to **lower clinical efficiency** and loss of **20 million** every year to the Brazilian economy.
- ***Objective* -** To investigate the reason why some patients do not show up to their scheduled appointments.
- Data was gathered from **kaggle's** Medical Appointment No Show dataset and loaded in **google colaboratory** for analysis.
- Dataset has more than **100K** records/rows.
- In **data-wrangling** major time was devoted to **assessing and cleaning data**. Data was **dirty** and **messy** with issues in its content.
- Cleaning **invalid data** like float datatype for PatientID and AppointmentID, negative values in age column which is impossible.
- Removing **irrelevant data** like Appointment Time which was **00:00:00** (HH:MM:SS) in all the rows, some records have appointment day before the scheduled day.
- Transforming **messy data** like ScheduledDay and AppointmentDay having **multiple variables** in date-time format (dd-mmm-yyyy HH:MM:SS) in a single column. They were separated into different columns such that there is **one variable per column.**
- Renaming column name in **snake case** to access the column using period with data frame like df.column_name
- Summarizing features and finding **descriptive statistics** like a **five-number summary** for the age column.
- Handling outliers in age column using **68–95–99.7 rule**.
- Undertaken **exploratory data analysis** (EDA) to find the important feature responsible for the no-show.
- To support our analysis used libraries like **matplotlib** and **seaborn** to make **clean**, **uncluttered design** with **easy-to-interpret** data visualization.
- Both **categorical** and **quantitative** variables were used for visualization.
- **Important features** to predict no-shows are age, hypertension, diabetes, neighborhood, and scholarship.