BTA Full-Time Data Scientist Assessment- Report

Q1. Machine Learning and EDA

Objective
Analyze the California Housing dataset, perform exploratory data analysis (EDA), and build a machine learning model to predict housing prices. Dataset: California Housing dataset (available in the sklearn.datasets module).

Code file

Some information regarding the dataset is referenced from Real world datasets and understood the feature attribution.

- MedInc median income in block group
- HouseAge median house age in block group
- AveRooms average number of rooms per household
- AveBedrms average number of bedrooms per household
- Population block group population
- AveOccup average number of household members
- Latitude block group latitude
- Longitude block group longitude

The target variable is the median house value for California districts, expressed in hundreds of thousands of dollars ($100,000). This dataset was derived from the 1990 U.S. census, using one row per census block group. A block group is the smallest geographical unit for which the U.S. Census Bureau publishes sample data (a block group typically has a population of 600 to 3,000 people). A household is a group of people residing within a home. Since the average number of rooms and bedrooms in this dataset are provided per household, these columns may take surprisingly large values for block groups with few households and many empty houses, such as vacation resorts.

Methodology and Findings
Loaded the California_housing dataset from sklearn datasets and converted it into pandas dataframe.There were no missing values. The descriptive statistics showed

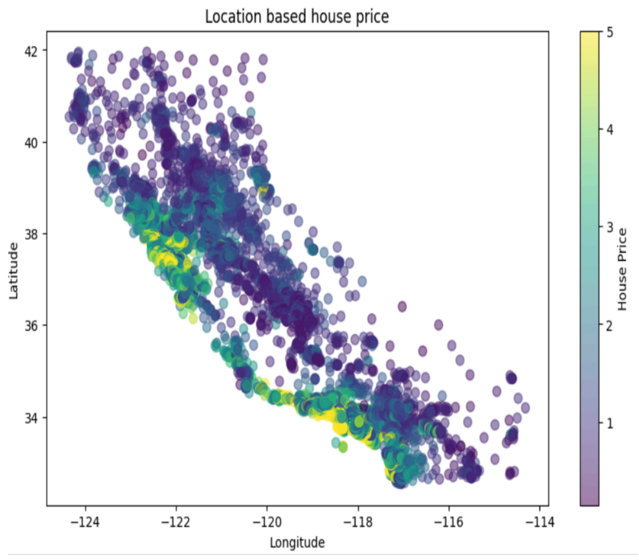
some possibilities of outliers. By using a hist plot, I have derived some conclusions regarding their distribution.,

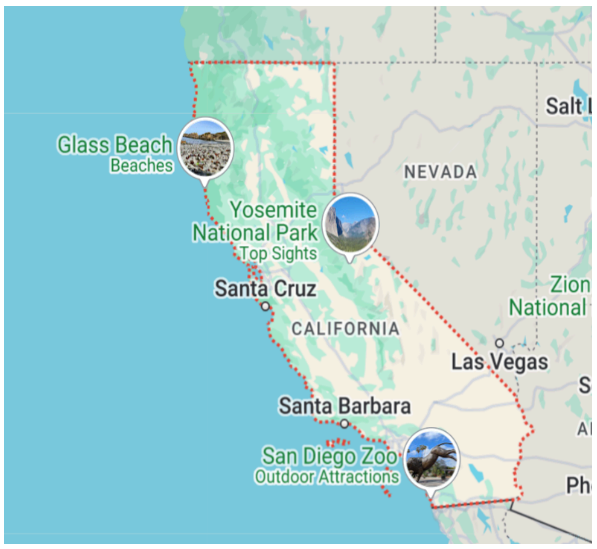
- MedInc is almost normally distributed but there are a few who have higher income.
- HouseAge has almost a uniform distribution.
- HousePrice has a long tail.

By using a boxplot I understood that AvgRooms, AvgBedrms, Population, AvgOccup have a higher maximum value, which might be an outlier. By using a pair plot and a correlation matrix, it's clearly understood that

- MedInc has a higher correlation with HousePrice, which means Higher income families normally have high price houses.
- Latitude and Longitude have a lower correlation, so there might be some special pattern related to geographical areas.
- MedInc and AvgRooms have higher correlation, which means higher income families are having bigger houses.

Then A scatterplot is used to plot latitude and longitude against the House price and there was was special pattern found as the majority of the higher priced houses are located in a certain pattern.





By analyzing the geographic map of california along with the location based scatter plot, It's visible that the high priced houses are majorly located along with the seaside.

For the Model building I have used linear regression and random forest regressor. The dataset was splitted as 20% test data and 80% training data For the evaluation matrix I have used mean absolute error, mean squared error and R square.

Evaluation matrix	Linear Regression	Random Forest
MAE	0.53	0.32
MSE	0.55	0.25
R square	0.57	0.80

Random forest is the best performing model since it has the lowest value of errors. Then I have tried hyper parameter tuning for random forest and found the best parameters. Then I have found the importance of every feature.

Q2. Case Study: Creating a Computer Vision-Based Solution to Track People at an Airport

Objective: Develop a computer vision solution to monitor and track people at an airport for security and operational efficiency.

Code File

Demo Video

Methodology and Findings

Since the video was from youtube the first step was to download the video from youtube in colab. For that special package needed to be installed in colab and I have gone with `yt-dlp`. Then I have imported the video from the youtube link and converted the video into image frames and as a result there are a total of 9458 frames. Since the low capacity of the system and huge time taken for processing the complete image frames, I have divided the images into test and train. Also for a sample purpose now I'm dealing with only the test set frames since its the smallest quality set. For object detection and tracking I have cloned the YOLOv5 repository, installed dependencies, and run detection on test frames. Since there were a lot of images to annotate and the main purpose was to create a sample system for object detection, I decided to go with pretrained models instead of annotating from scratch. By using the `detect.py` the pretrained model was applied over the frames to detect and draw bounding boxes. Then the annotated frames are combined together to form a video.to see the detection results. Then the Recall, Precision and mean average precision were calculated.

The model evaluation output provides the following metrics:

- **Precision**: 68.9% of the model's positive predictions are true positives.
- **Recall**: 63.4% of the actual positive instances are correctly identified by the model.
- **mAP50**: An average precision of 70.9% at IoU threshold 0.5, indicating how well the model is performing at a common IoU threshold.
- **mAP50-95**: An average precision of 47.6% across a range of IoU thresholds, which provides a more comprehensive view of model performance across different levels of overlap.
- The files in the `runs/val/exp` directory provide insights into various aspects of the model's performance. These files provide a detailed view of model performance, including visualizations of precision, recall, and F1 scores, as well as confusion matrices and example predictions.