Sporting Future: A Comprehensive Analysis of Athletic Facilities

Group-9

Dhirubhai Ambani Institute of Information and Communication Technology
(DA-IICT)
Gandhinagar

Group 9

Keerrtivardhan Goyal
ID:
202103007
Course:
BTech(MnC)

Yash Mashru
ID:
202103045
Course:
BTech(MnC)

Sanchit Satija
ID:
202103054
Course:
BTech(MnC)

Content

Outline

- Introduction and Defining the problem
- Shape and how big is data
- Data Cleaning
- Visualization
- Feature Engineering
- Model Fitting and Evaluation
- Conclusion

Problem Statement

When designing sports facilities, many factors are needed to be considered for optimal resource allocation and layout planning. This leads to the situation where various dimensions of the facility, i.e., length, width, area, precinct, material, etc. are within consideration when deciding the budget and land allocation. This gives a compelling reason to obtain the optimal predictive models of the dimensions of the facility based on historical data on sports facilities.

Packages Required

Throughout our journey of data cleaning and modelling, we will be using many different **python** libraries for our dataset cleaning and visualization purposes. The major ones among them are :

- NumPy Library
- Pandas Library
- PyPlot from MatPlotLib Library
- MissingNo Library
- SeaBorn Library
- SKlearn library

Apart from this, Most of the Visualizations has been done with the tool **Microsoft PowerBI**.

Dataset Description - Data Role

The dataset 'Athletic_Facilities' describes the distribution of various sports facilities around different districts with 6897 rows of records over 44 columns. It distinguishes the facilities based on factors such as

- The sport the facility is made for - BASKETBALL, VOLLEYBALL, LL_SOFTBALL, TENNIS, etc
- Its material - SURFACE_TYPE
- The department and jurisdiction it belongs to - BOROUGH, COMMUNITYBOARD, COUNCILDISTRICT, DEPARTMENT, GISPROPNUM, ZIPCODE
- Its various dimensions and measurements - PRECINCT, STArea, STLength
- Identification details - PRIMARY_SPORT, SYSTEM, FIELD_NUMBER
- Accessibility - ACCESSIBILITY, FEATURESTATUS, FIELD_LIGHTED, MAINTENANCEAGREEMENT

Dataset Description - Data Type

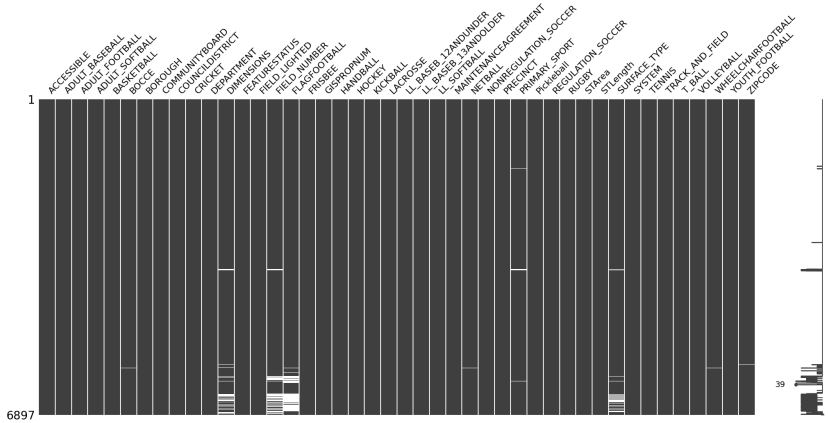
- **Boolean** - ACCESSIBLE, ADULT_BASEBALL, ADULT_FOOTBALL, ADULT_SOFTBALL, BASKETBALL, CRICKET, FRISBEE, FIELD_LIGHTED, HANDBALL, HOCKEY, KICKBALL, LACROSSE, LL_BASEB_12ANDUNDER, LL_BASEB_13ANDOLDER, LL_SOFTBALL, MAINTENANCEAGREEMENT, FLAGFOOTBALL, Pickleball, NONREGULATION_SOCCER, REGULATION_SOCCER, RUGBY, TENNIS, TRACK_AND_FIELD, T_BALL, VOLLEYBALL, YOUTH_FOOTBALL
- **Integer** - COMMUNITYBOARD, COUNCILDISTRICT, PRECINCT
- **Float** - STArea, STLength, ZIPCODE
- **Object** - BOCCE, BOROUGH, DEPARTMENT, DIMENSIONS, FEATURESTATUS, FIELD_NUMBER, GISPROPNUM, NETBALL, PRIMARY_SPORT, SURFACE_TYPE, SYSTEM, WHEELCHAIRFOOTBALL

Data Cleaning - Detecting the Null Values

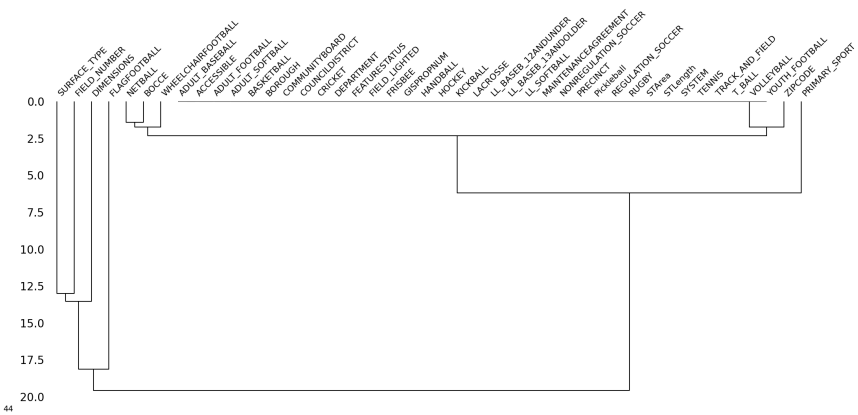
- So let's first see the missing values present in each of the columns of our dataset.
- We can do this by *df.isnull().sum()*.So the number of missing values in our dataset is as follows :
  - FLAG FOOTBALL: 546
  - FIELD NUMBER: 418
  - DIMENSIONS: 274
  - SURFACE TYPE: 325
  - PRIMARY SPORT: 38
  - WHEELCHAIRFOOTBALL: 6
  - NETBALL: 5
  - BOCCE: 5
  - ZIPCODE: 3

  while the rest of the columns don't have null values.
- Now we visualize these null values by missingno. package and then try to detect and remove the outliers by IQR method.

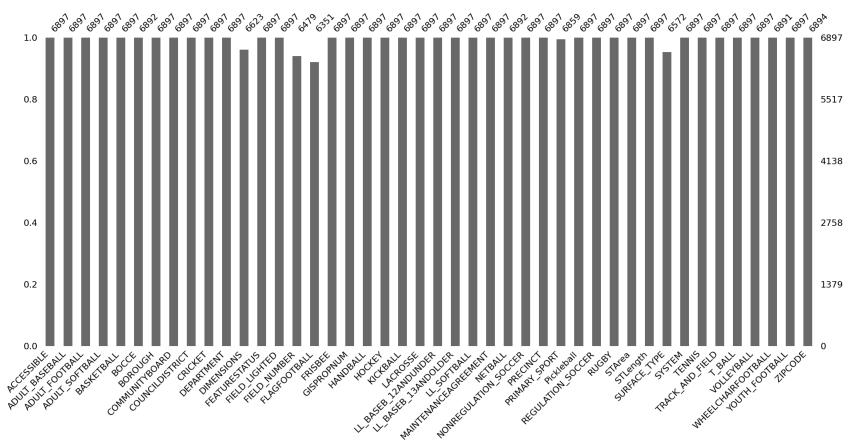
Data Cleaning - Visualizing by MissingNo. Package



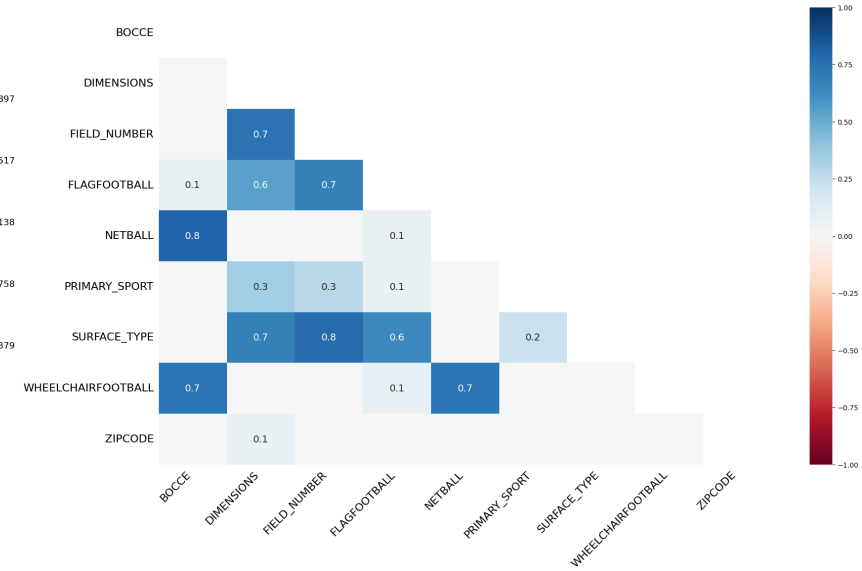
(a) Matrix Plot



(b) Dendrogram Plot

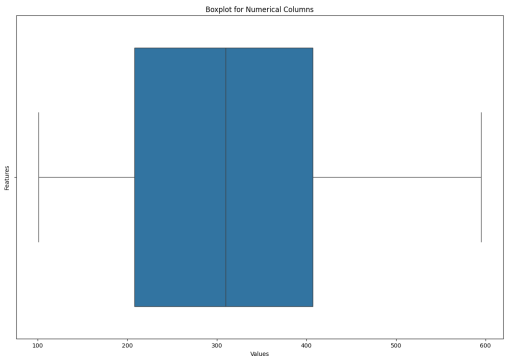


(c) Bar Plot

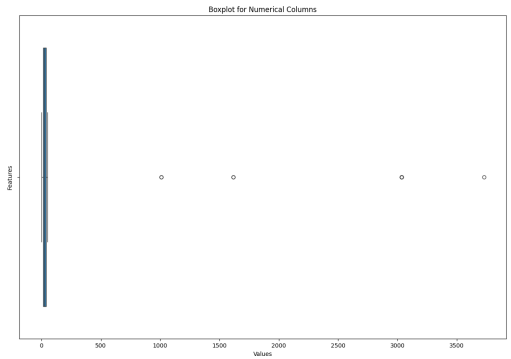


(d) Heatmap

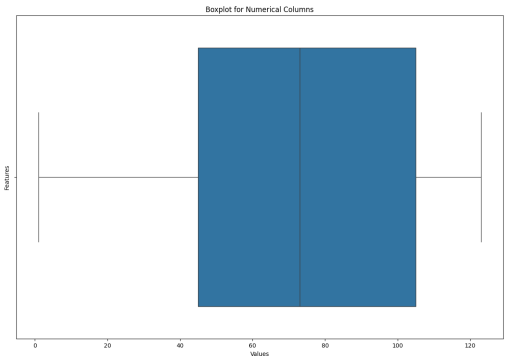
Data Cleaning - BoxPlots before removing outliers



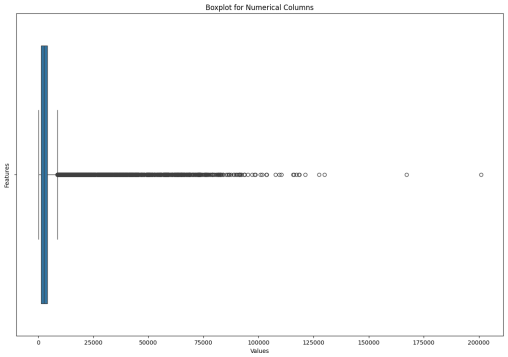
(a) CommunityBoard



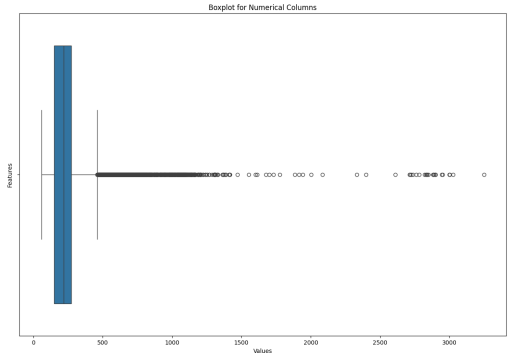
(b) CouncilDistrict



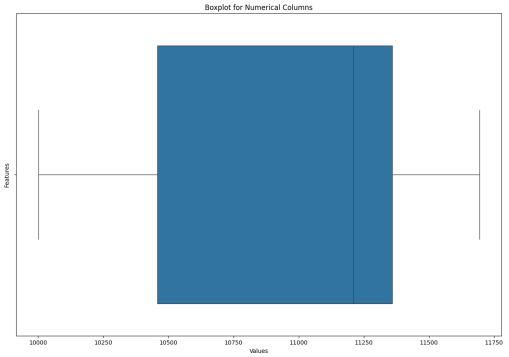
(c) PRECINCT



(d) STArea



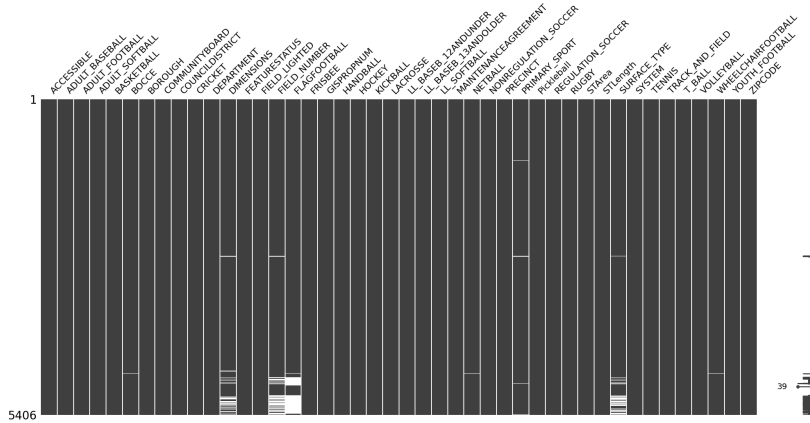
(e) STLength



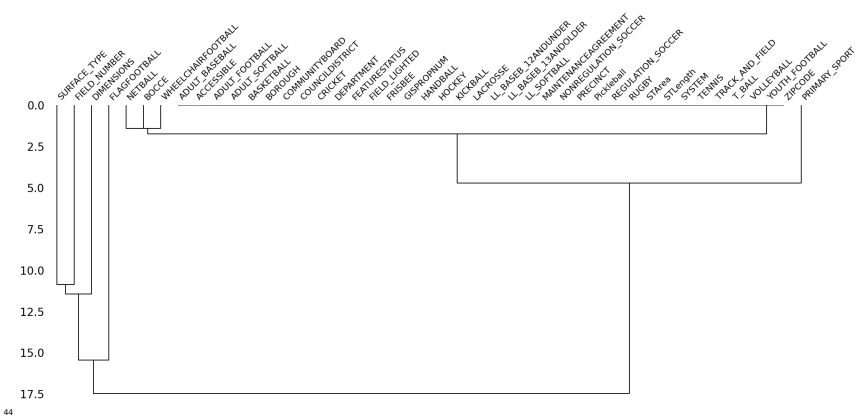
(f) ZIPCODE

Figure 2: BoxPlots before removing outliers

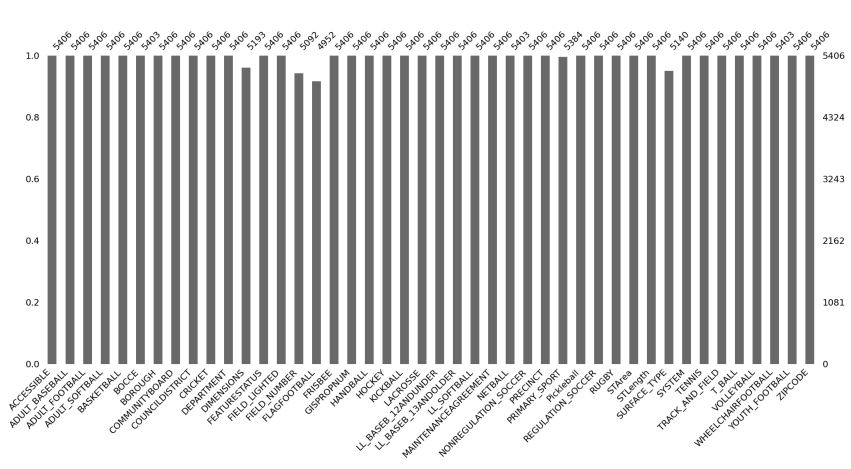
Data Cleaning - Visualizing by MissingNo. Package after removing outliers



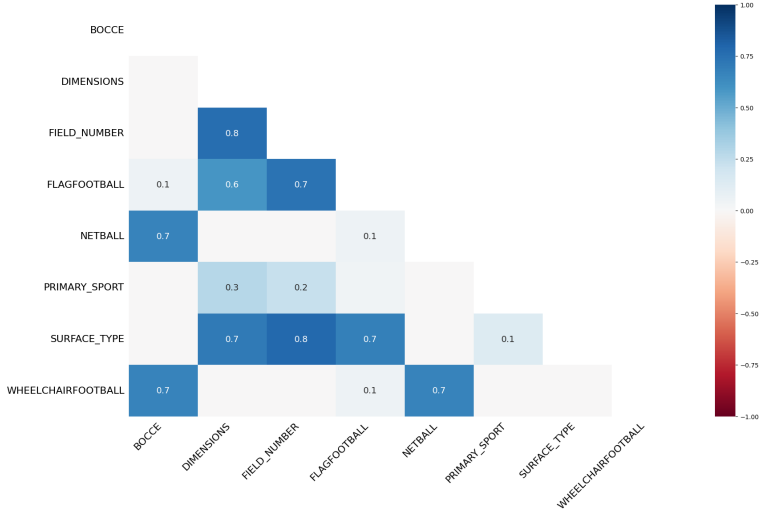
(a) Matrix Plot



(b) Dendrogram Plot

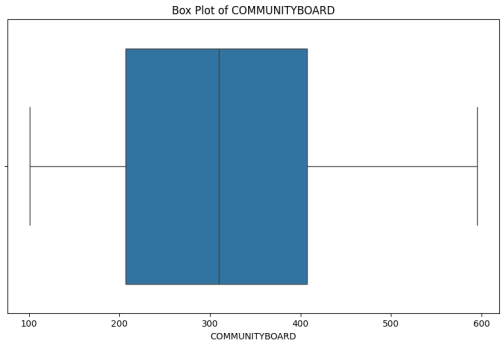


(c) Bar Plot

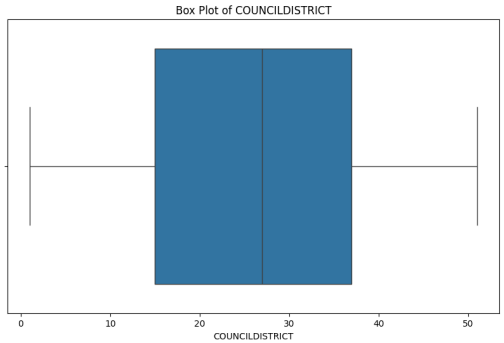


(d) Heatmap

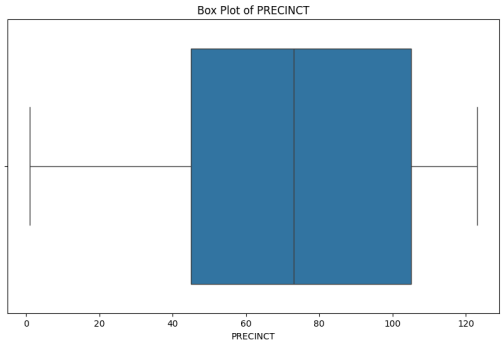
Data Cleaning - BoxPlots after removing outliers



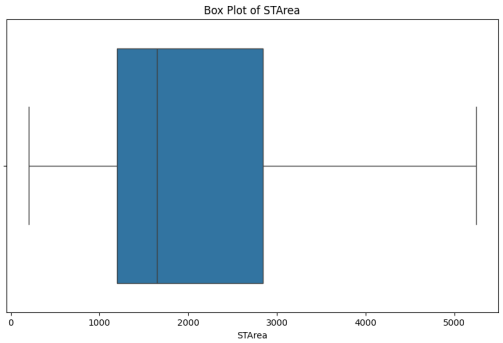
(a) CommunityBoard



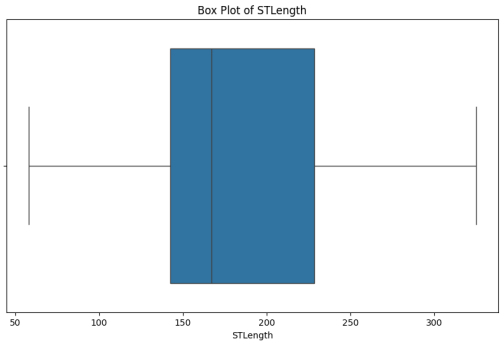
(b) CouncilDistrict



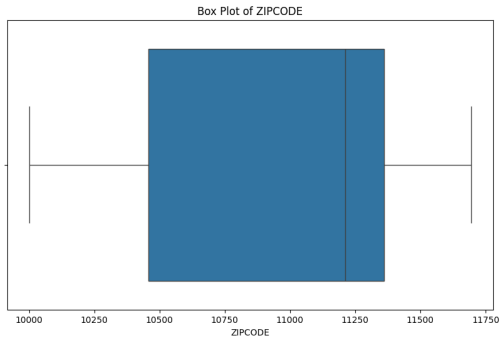
(c) PRECINCT



(d) STArea



(e) STLength



(f) ZIPCODE

Figure 4: BoxPlots after removing outliers

Data Cleaning - Detecting Type Of Missingness

- Now after removing the outliers and visualizing them, we will detect which type of missingness is there in our data , which means whether it is MCAR, MAR or MNAR type.
- So we will first analyze the Little MCAR test.We have attached the code and output of the same :

```
from scipy.stats import chi2

def little_mcar_test(data, alpha=0.05):
    data = pd.DataFrame(data)
    data.columns = ['x' + str(i) for i in range(data.shape[1])]
    data['missing'] = np.sum(data.isnull(), axis=1)
    n = data.shape[0]
    k = data.shape[1] - 1
    df = k * (k - 1) / 2
    chi2_crit = chi2.ppf(1 - alpha, df)
    chi2_val = ((n - 1 - (k - 1) / 2) ** 2) / (k - 1) / ((n - k) * np.mean(data['missing']))
    p_val = 1 - chi2.cdf(chi2_val, df)
    if chi2_val > chi2_crit:
        print(
            'Reject null hypothesis: Data is not MCAR (p-value={:.4f}, chi-square={:.4f})'.format(p_val, chi2_val)
        )
    else:
        print(
            'Do not reject null hypothesis: Data is MCAR (p-value={:.4f}, chi-square={:.4f})'.format(p_val, chi2_val)
        )
little_mcar_test(df,0.05)
```

```
Do not reject null hypothesis: Data is MCAR (p-value=1.0000, chi-square=531.7155)
```

Figure 5: Performing Little MCAR test to check whether the missingness is of type MCAR or not.We can see from the code that the null hypothesis is not rejected .So missingness is of MCAR type.

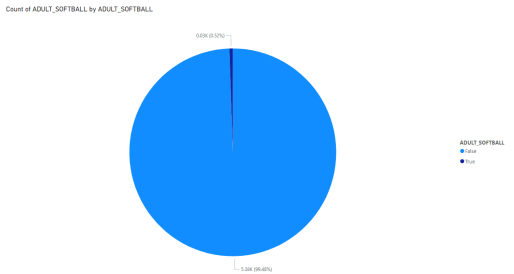
Data Cleaning - Imputation of the Null Values

- As missingness present in our data is of MCAR type, we can do Mean,Median or Mode imputation.
- So we do mean imputation if there is normal or Gaussian Distribution otherwise for skewed data we use median imputation.
- For numerical columns, the column named ZIPCODE only had null values and when we analyzed *df.skew()* function , its skewness was coming to be -0.578887.So it is slightly left skewed
- So we have done **mean imputation** for the **numerical column** (as the one column which has null values is slightly skewed) .
- For **categorical columns**, we imputed them with **mode**.

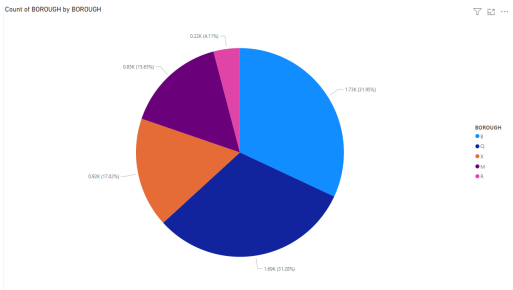
Data Visualization

- Data Visualization is the practice of representing data through visual elements like graphs, charts, maps, etc. to make understanding of the data more intuitive and insightful.
- It can help present data in a form that makes it possible for parties not involved in the processing of data or those lacking relevant knowledge to understand it without confusion.
- We have done many visualizations in **PowerBI** also.
- Now we will perform visualizations on the cleaned data which means after removing null values and outliers.

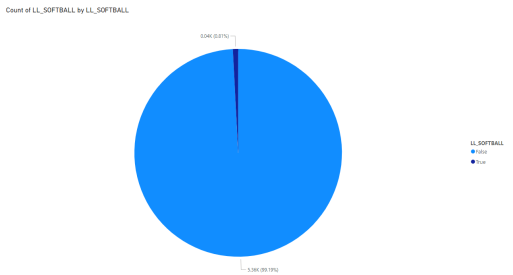
Data Visualization - Pie Plots after removing outliers and null values



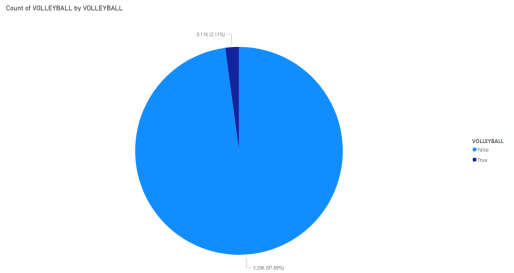
(a) ADULT_VOLLEYBALL



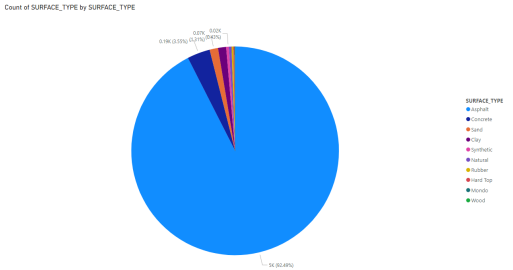
(b) BOROUGH



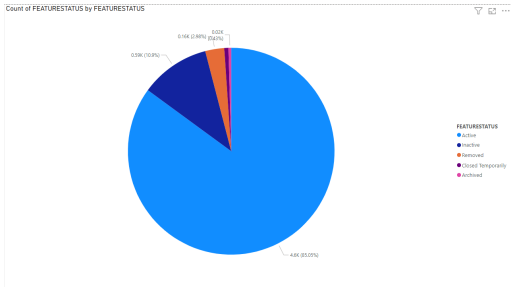
(c) LL_SOFTBALL



(d) VOLLEYBALL



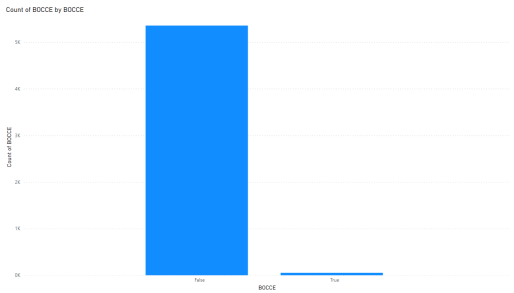
(e) SURFACE_TYPE



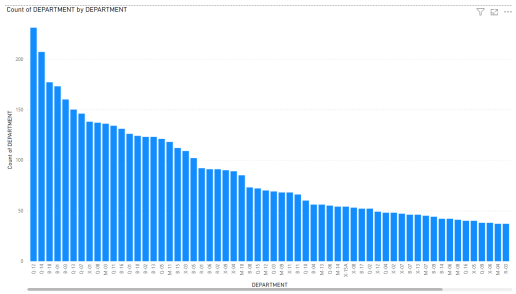
(f) FEATURESTATUS

Figure 6: Pie Plots after removing outliers and null values

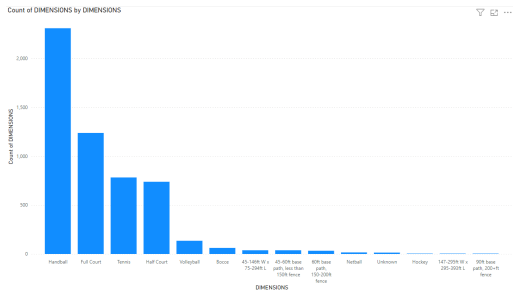
Data Visualization - Bar Plots after removing outliers and null values



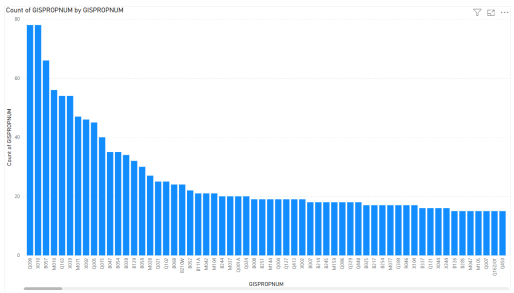
(a) BOCCE



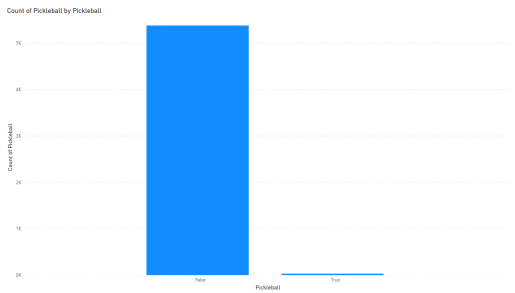
(b) DEPARTMENT



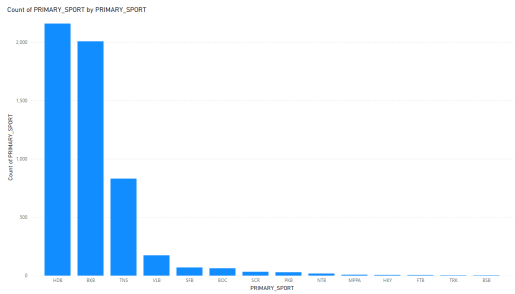
(c) DIMENSIONS



(d) GISPROPNUM



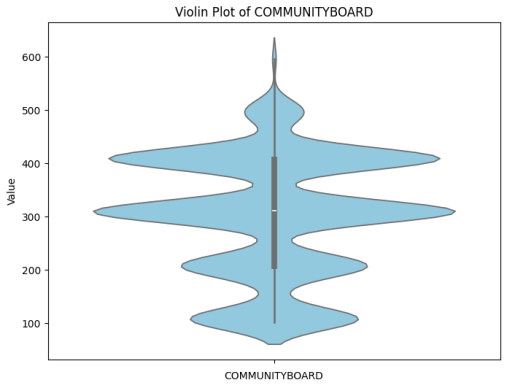
(e) Picleball



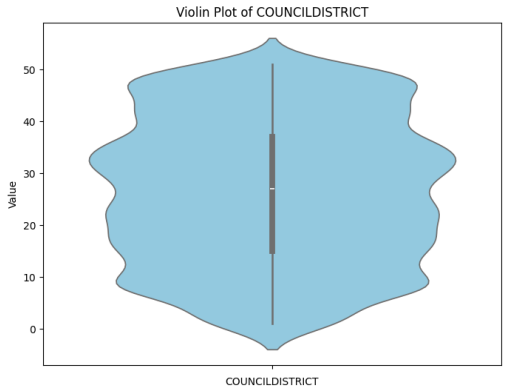
(f) PRIMARY_SPORT

Figure 7: Bar Plots after removing outliers and null values

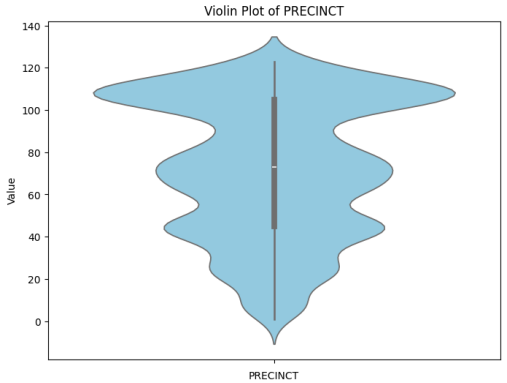
Data Visualization - Violin Plots after removing outliers and null values



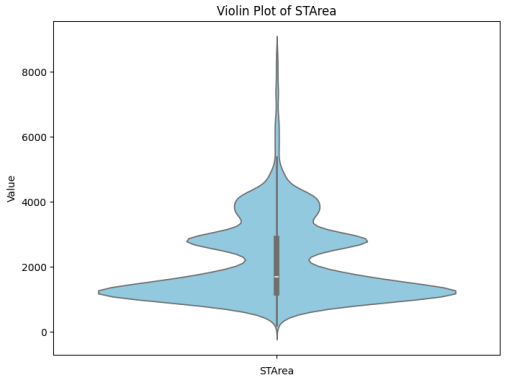
(a) COMMUNITYBOARD



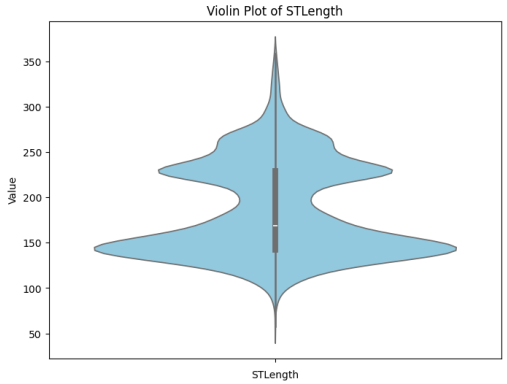
(b) COUNCILDISTRICT



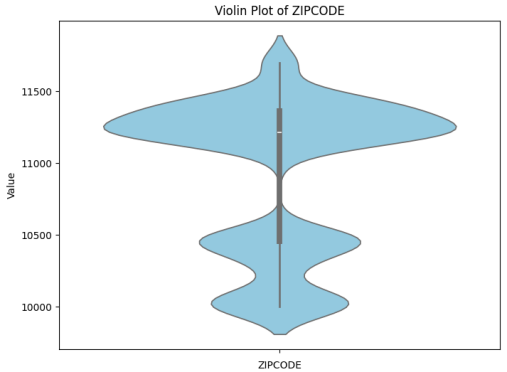
(c) PRECINCT



(d) STArea



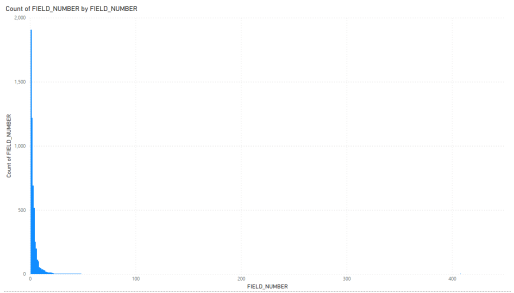
(e) STLength



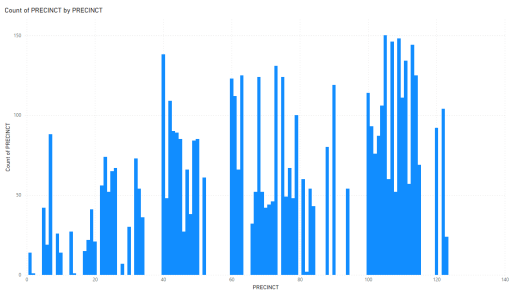
(f) ZIPCODE

Figure 8: Violin Plots after removing outliers and null values

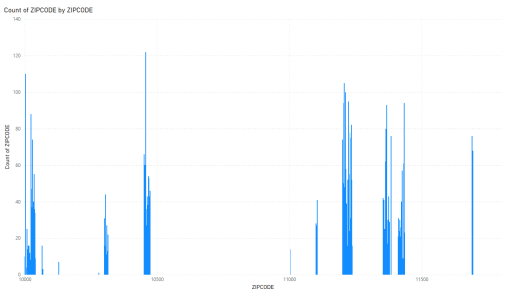
Data Visualization - Histograms after removing outliers and null values



(a) FIELD_NUMBER



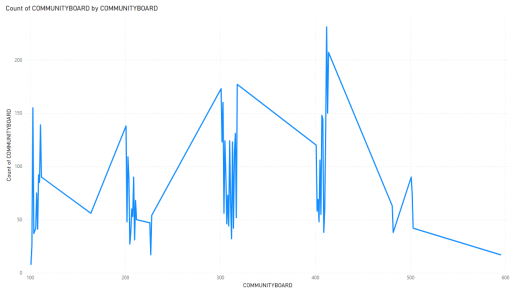
(b) PRECINCT



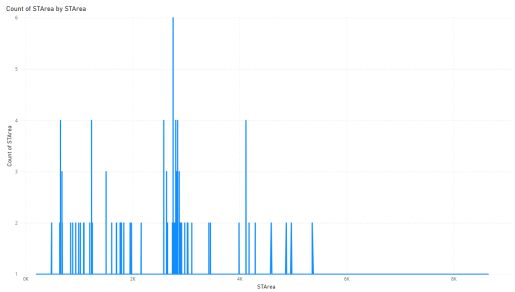
(c) ZIPCODE

Figure 9: Histograms after removing outliers and null values

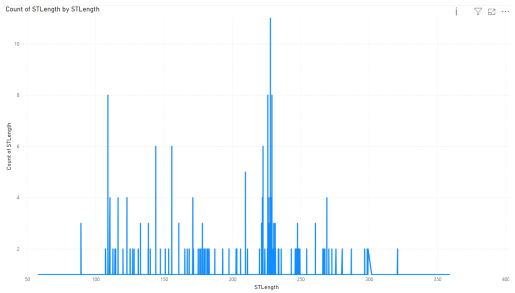
Data Visualization - Line Plots after removing outliers and null values



(a) COMMUNITYBOARD



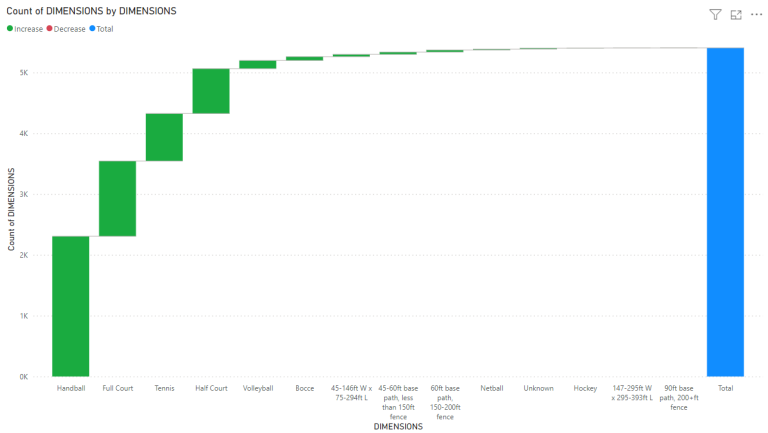
(b) STArea



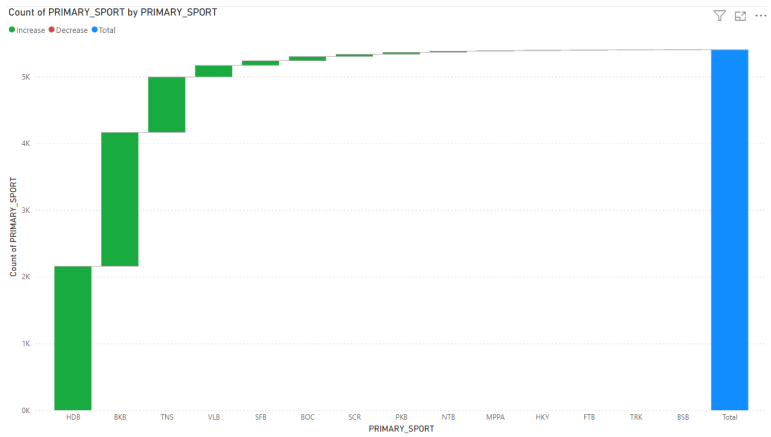
(c) STLength

Figure 10: Line Plots after removing outliers and null values

Data Visualization - Waterfall Plots after removing outliers and null values



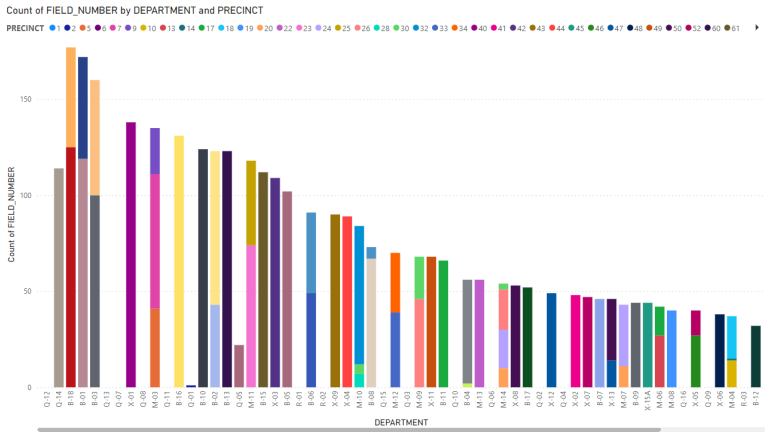
(a) DIMENSIONS



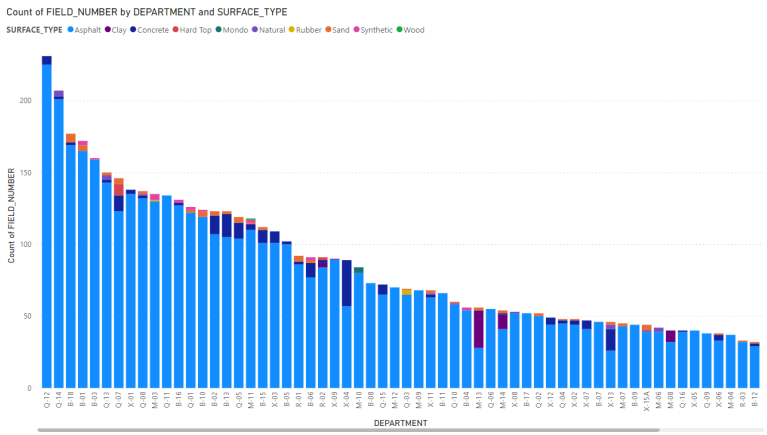
(b) PRIMARY_SPORT

Figure 11: Waterfall Plots after removing outliers and null values

Data Visualization - Stacked Bar Plots after removing outliers and null values



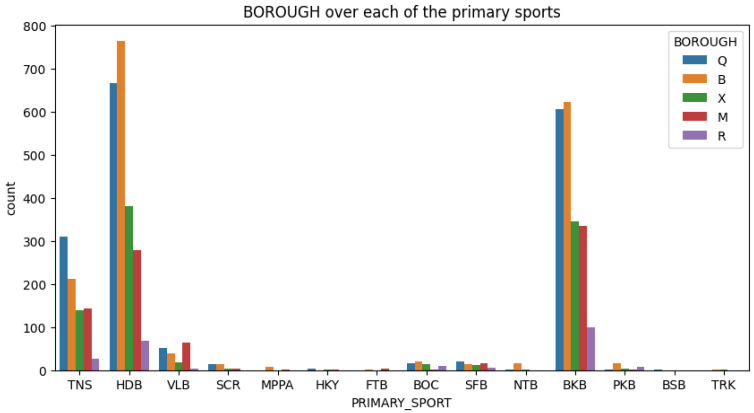
(a) PRECINCT



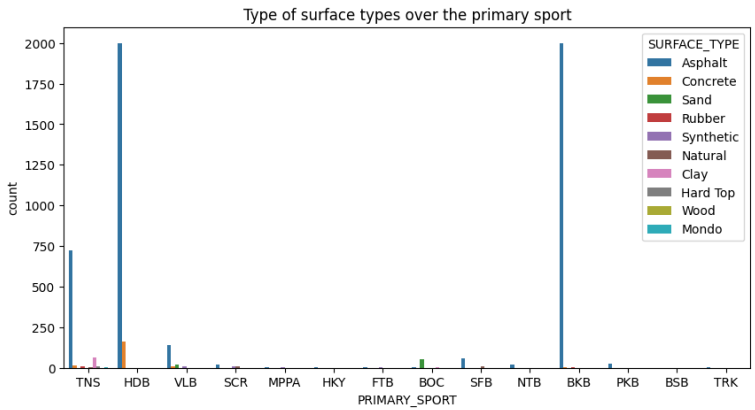
(b) SURFACE_TYPE

Figure 12: Stacked Bar Plots after removing outliers and null values

Data Visualization - Grouped Bar Plots after removing outliers and null values



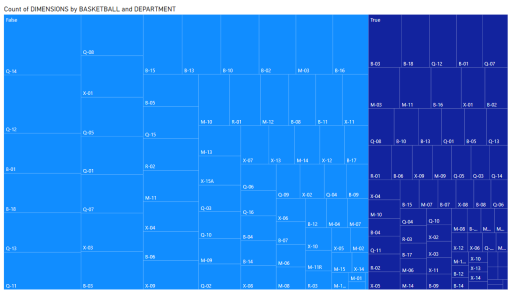
(a) BOROUGH



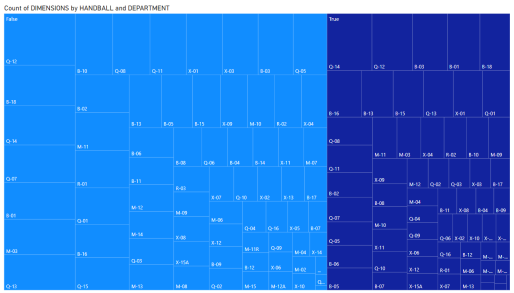
(b) SURFACE_TYPE

Figure 13: Grouped Bar Plots after removing outliers and null values

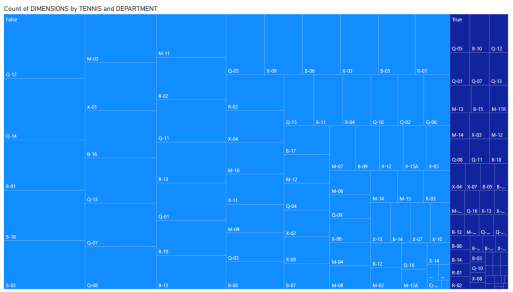
Data Visualization - Tree Plots after removing outliers and null values



(a) BASKETBALL



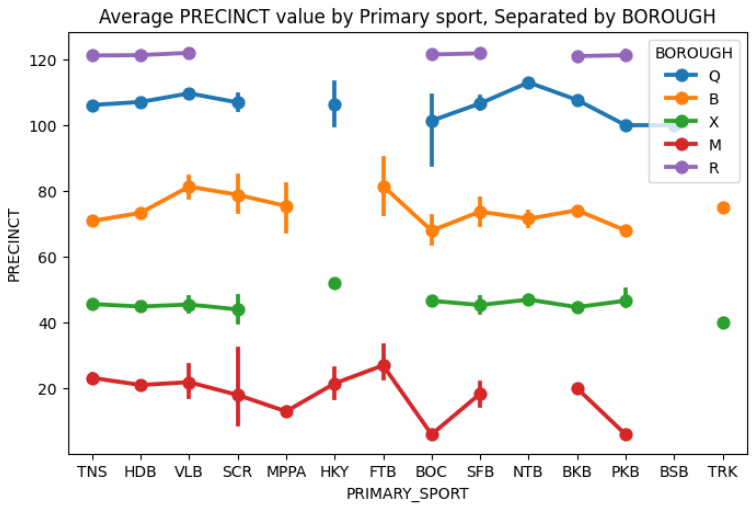
(b) HANDBALL



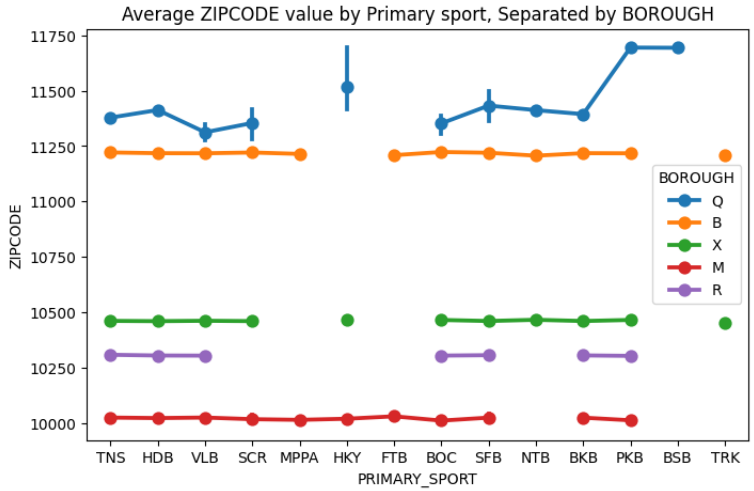
(c) TENNIS

Figure 14: Tree Plots after removing outliers and null values

Data Visualization - Point Plots after removing outliers and null values



(a) PRECINCT



(b) ZIPCODE

Figure 15: Point Plots after removing outliers

Data Visualization - Correlation Map after removing outliers and null values

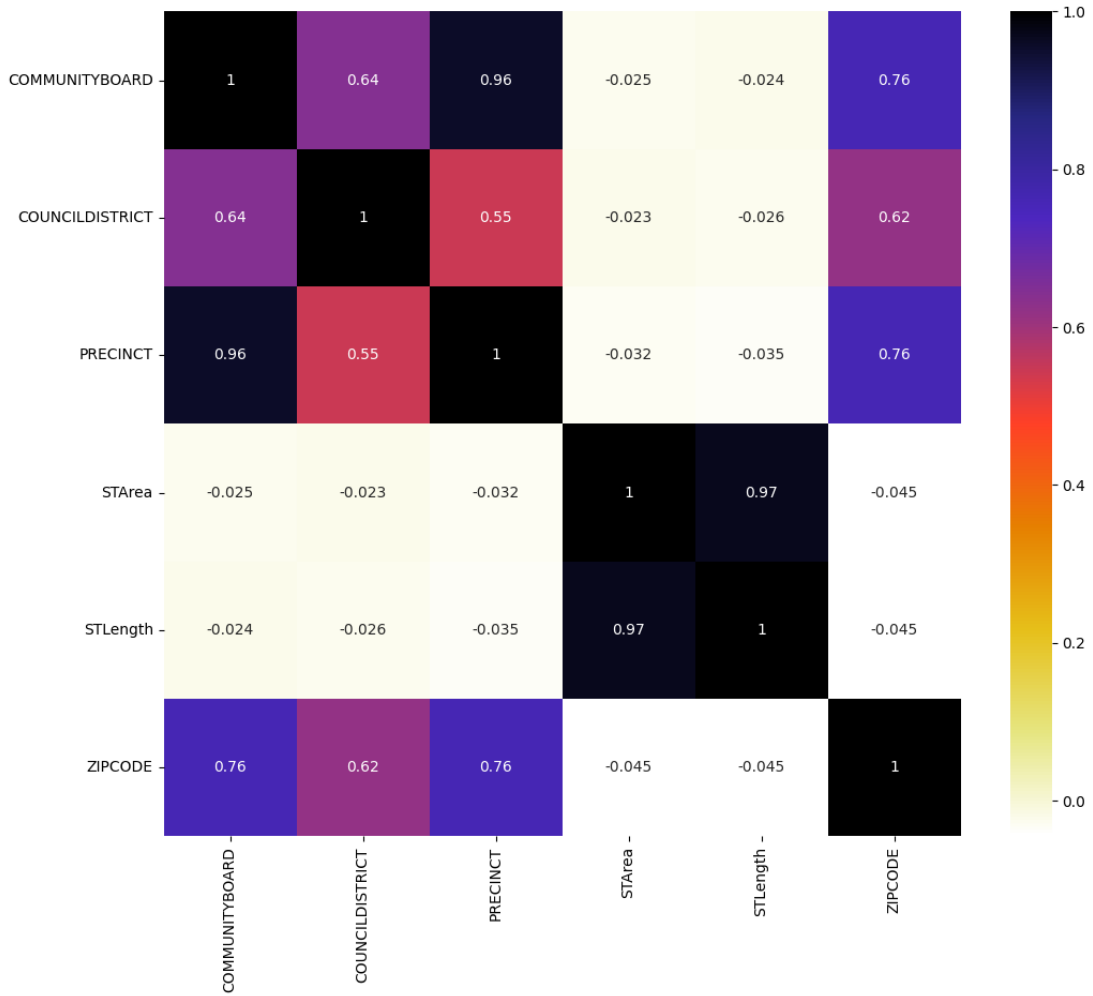
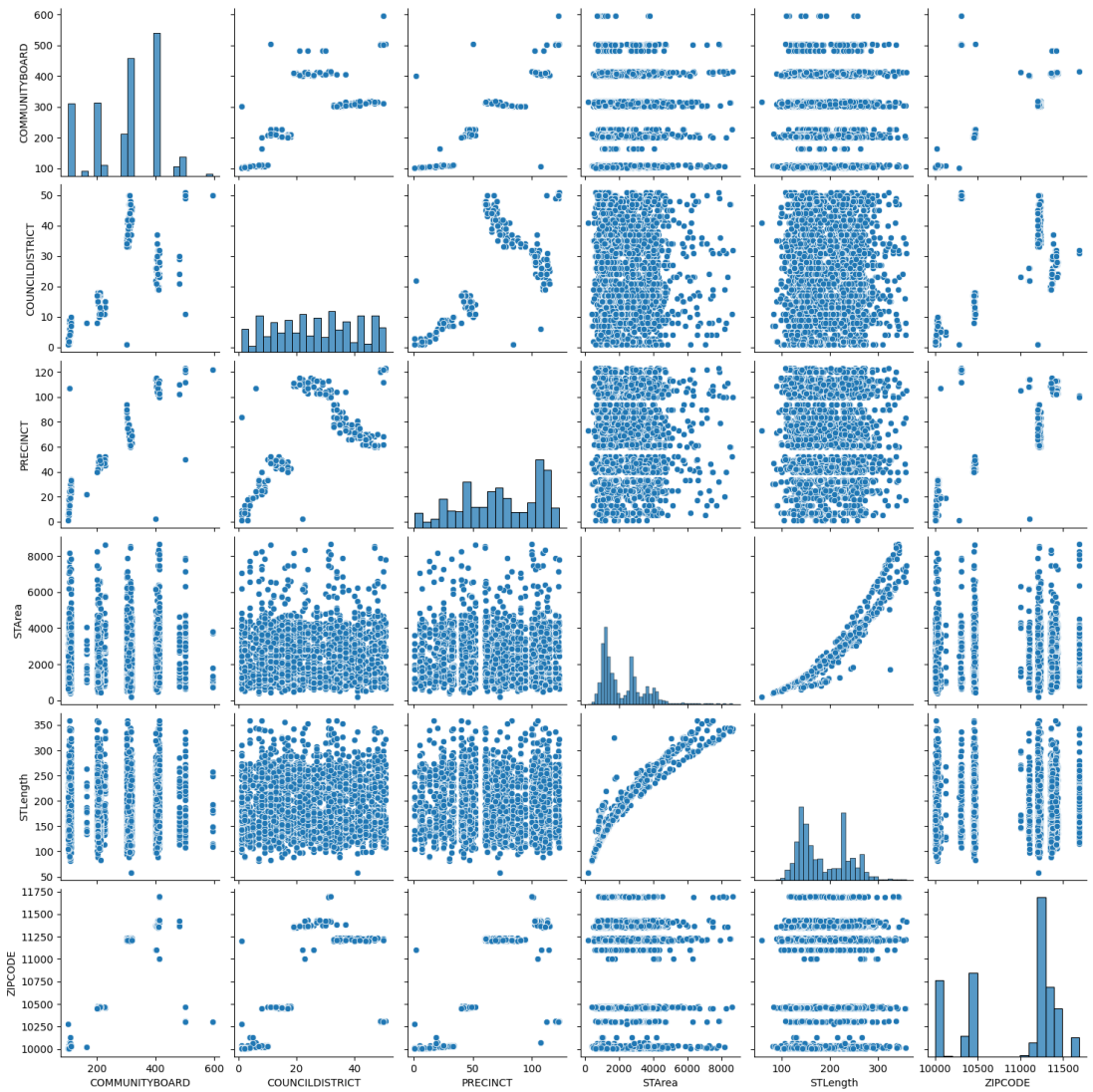


Figure 16: Correlation Map after removing outliers and null values

Data Visualization - Pair Plot after removing outliers and null values



Most Popular Sport

Which is the most popular sport?

```
most_popular_sport = df['PRIMARY_SPORT'].mode()[0]
most_popular_sport_count = df['PRIMARY_SPORT'].value_counts().max()

print("\nMost Popular Sport:", most_popular_sport)
print("Number of Occurrences:", most_popular_sport_count)
```

```
Most Popular Sport: HDB
Number of Occurrences: 2159
```

Handball (HDB) is the most popular sport as it has the most occurences.

Count of Basketball facilities

Count of basketball facilities: Get the total count of entries where BASKETBALL is present.

```
basketball_count = df['BASKETBALL'].sum()
print("Total count of basketball facilities:", basketball_count)
```

```
Total count of basketball facilities: 1498
```

'BASKETBALL' facilities appears the most with 1498 entries.

Accessible facilities with Basketball

Percentage of accessible facilities with basketball

```
[ ]  accessible_basketball_percentage = df[(df['ACCESSIBLE'] == 1) & (df['BASKETBALL'] == 1)].shape[0] / df.shape[0] * 100
     print("Percentage of accessible facilities with basketball:", accessible_basketball_percentage)
```

```
Percentage of accessible facilities with basketball: 0.01849796522382538
```

The percentage of facilities where Basketball is accessible is 0.0185%.

Athletic Facilities per Zipcode

```
ZIP code: 11361.0
Sport(s) with highest true count: ['TENNIS']
True count: 12

ZIP code: 11209.0
Sport(s) with highest true count: ['HANDBALL']
True count: 17

ZIP code: 11228.0
Sport(s) with highest true count: ['HANDBALL']
True count: 20

ZIP code: 11217.0
Sport(s) with highest true count: ['HANDBALL']
True count: 13

ZIP code: 11201.0
Sport(s) with highest true count: ['HANDBALL']
True count: 29

ZIP code: 11213.0
Sport(s) with highest true count: ['HANDBALL']
True count: 29

ZIP code: 11220.0
Sport(s) with highest true count: ['BASKETBALL']
True count: 22

ZIP code: 11232.0
Sport(s) with highest true count: ['BASKETBALL', 'HANDBALL']
True count: 8

ZIP code: 11203.0
Sport(s) with highest true count: ['HANDBALL']
True count: 20
```

Sport with Maximum total Area and Sport with Maximum total Length among active sports

Find the sport with the maximum total area and the sport with the maximum total length among active sports, and display their respective total areas and lengths

```
[ ]  active_sports = df[df['FEATURESTATUS'] == 'Active']
     result = active_sports.groupby('PRIMARY_SPORT').agg({'STArea': 'sum', 'STLength': 'sum'}).reset_index()
     max_area_sport = result.loc[result['STArea'].idxmax()]
     max_length_sport = result.loc[result['STLength'].idxmax()]

     print("Sport with maximum total STArea (Active):", max_area_sport['PRIMARY_SPORT'], "Total STArea:", max_area_sport['STArea'])
     print("Sport with maximum total STLength (Active):", max_length_sport['PRIMARY_SPORT'], "Total STLength:", max_length_sport['STLength'])

     Sport with maximum total STArea (Active): BKB Total STArea: 4782239.8940765
     Sport with maximum total STLength (Active): BKB Total STLength: 363053.57075475
```

Basketball is the sport with maximum total area i.e 4,782,239.89 square units. It is also the sport with Maximum total length i.e 363,053.57 square units.

Feature Selection

- In Our dataset, categorical features are nominal. One Hot Encoding has been used to convert the categorical features into numerical features for training purposes.
- After Encoding - 5404 rows and 6372 columns.
- Two suitable methods for Feature Selection have been used:
  - Mutual Information
  - Variance Thresholding

Mutual Information

- Target Variable - STArea
- Works by Quantifying the dependency between two variables
- Calculates the quantity of information gained concerning the target variable.
- Selects the most informative features based on relation to target variable.

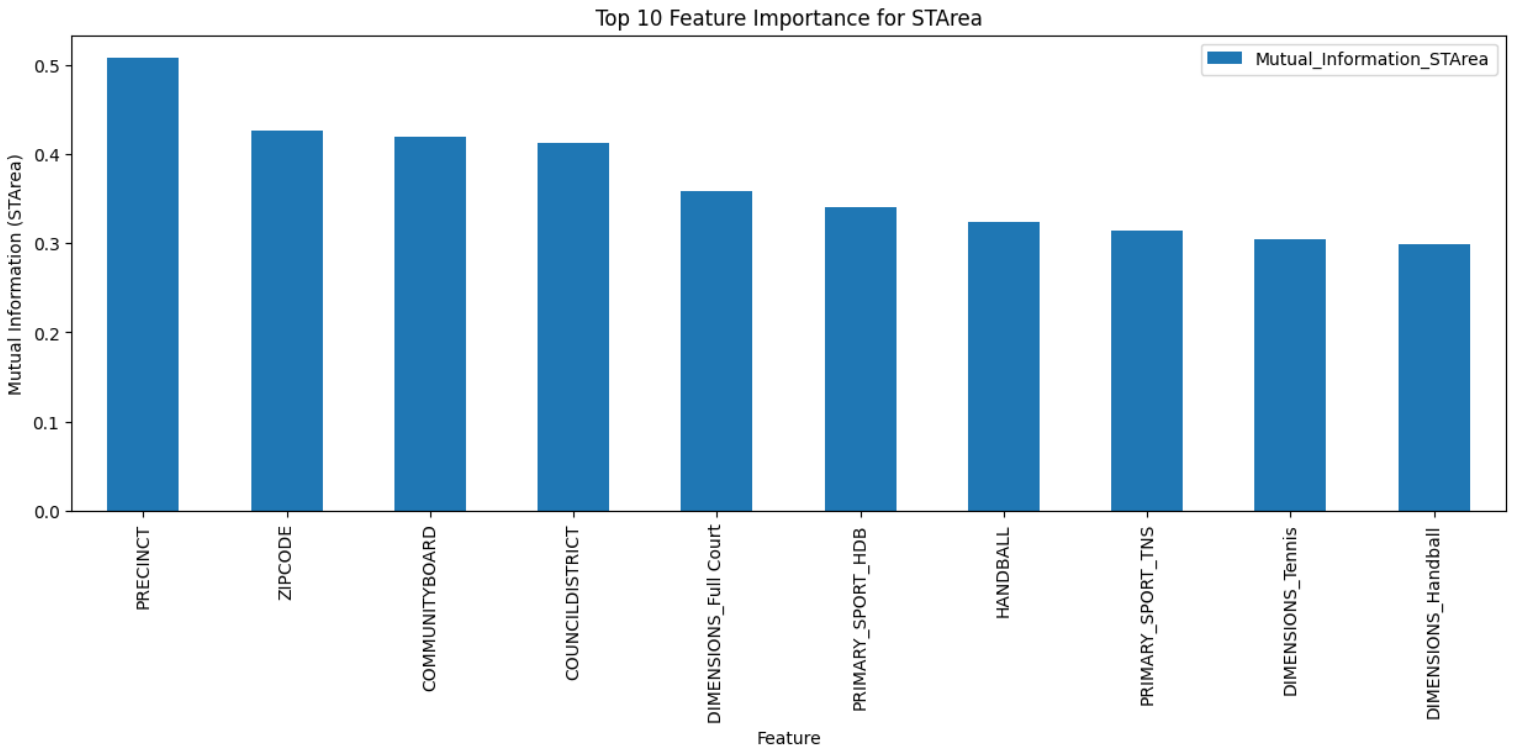
Mutual Information

```
print("Feature Importance for STArea:")
print(feature_importance_area)
```

```
Feature Importance for STArea:
                    Feature  Mutual_Information_STArea
22                PRECINCT                   0.508358
32                 ZIPCODE                   0.427194
6            COMMUNITYBOARD                   0.419734
7           COUNCILDISTRICT                   0.413238
111  DIMENSIONS_Full Court                   0.358278
..                     ...                        ...
140       FIELD_NUMBER_04A                   0.000000
191        FIELD_NUMBER_48                   0.000000
163        FIELD_NUMBER_23                   0.000000
153        FIELD_NUMBER_15                   0.000000
159   FIELD_NUMBER_2 and 4                   0.000000

[219 rows x 2 columns]
```

Top 10 features with Higher Mutual Information



PRECINCT column has the most substantial mutual information of 0.508358 with STArea.

Top 20 Percentile Feature Selection

```
X.columns[selected_top_columns_area.get_support()]
```

```
Index(['BASKETBALL', 'COMMUNITYBOARD', 'COUNCILDISTRICT', 'FIELD_LIGHTED',
       'HANDBALL', 'LL_SOFTBALL', 'MAINTENANCEAGREEMENT', 'PRECINCT', 'TENNIS',
       'VOLLEYBALL', 'ZIPCODE', 'BOROUGH_M', 'BOROUGH_Q', 'BOROUGH_R',
       'BOROUGH_X', 'DEPARTMENT_B-02', 'DEPARTMENT_B-13', 'DEPARTMENT_B-14',
       'DEPARTMENT_B-18', 'DEPARTMENT_M-11R', 'DEPARTMENT_Q-05',
       'DEPARTMENT_Q-13', 'DEPARTMENT_Q-14', 'DEPARTMENT_Q-15',
       'DEPARTMENT_Q-16', 'DEPARTMENT_R-02', 'DEPARTMENT_X-04',
       'DIMENSIONS_Bocce', 'DIMENSIONS_Full Court', 'DIMENSIONS_Half Court',
       'DIMENSIONS_Handball', 'DIMENSIONS_Tennis', 'DIMENSIONS_Volleyball',
       'FEATURESTATUS_Inactive', 'FIELD_NUMBER_1', 'PRIMARY_SPORT_BOC',
       'PRIMARY_SPORT_HDB', 'PRIMARY_SPORT_SCR', 'PRIMARY_SPORT_SFB',
       'PRIMARY_SPORT_TNS', 'PRIMARY_SPORT_VLB', 'SURFACE_TYPE_Clay',
       'SURFACE_TYPE_Concrete', 'SURFACE_TYPE_Sand'],
      dtype='object')
```

```
len(X.columns[selected_top_columns_area.get_support()])
```

```
44
```

Applied the SelectPercentile() function to select the top 20 percentile.

Variance Threshold

- Target Variable - STLength
- After applying Variance Threshold Feature Selection, the revised dataset has 5406 rows and 221 columns.
- Discards features having a variance lower than a pre set threshold (in our case, 0.1).
- Lower the dimension by discarding identical features
- Lower variance features contribute less to the predictions and are not important.

Variance Threshold

```
[ ]  selector_vt = VarianceThreshold(threshold=0.1)
     X_selected_vt = selector_vt.fit_transform(X)
```

```
[ ]  selected_features_vt = X.columns[selector_vt.get_support()]
```

```
     print("Selected Features after Variance Thresholding:")
     print(selected_features_vt)
```

```
Selected Features after Variance Thresholding:
Index(['BASKETBALL', 'COMMUNITYBOARD', 'COUNCILDISTRICT', 'HANDBALL',
       'PRECINCT', 'TENNIS', 'ZIPCODE', 'BOROUGH_M', 'BOROUGH_Q', 'BOROUGH_X',
       'DIMENSIONS_Full Court', 'DIMENSIONS_Half Court', 'DIMENSIONS_Handball',
       'DIMENSIONS_Tennis', 'FIELD_NUMBER_1', 'FIELD_NUMBER_2',
       'FIELD_NUMBER_3', 'PRIMARY_SPORT_HDB', 'PRIMARY_SPORT_TNS'],
      dtype='object')
```

```
[ ]  len(X.columns[selector_vt.get_support()])
```

```
19
```

19 features were selected who have variance higher than pre set threshold, 0.1.

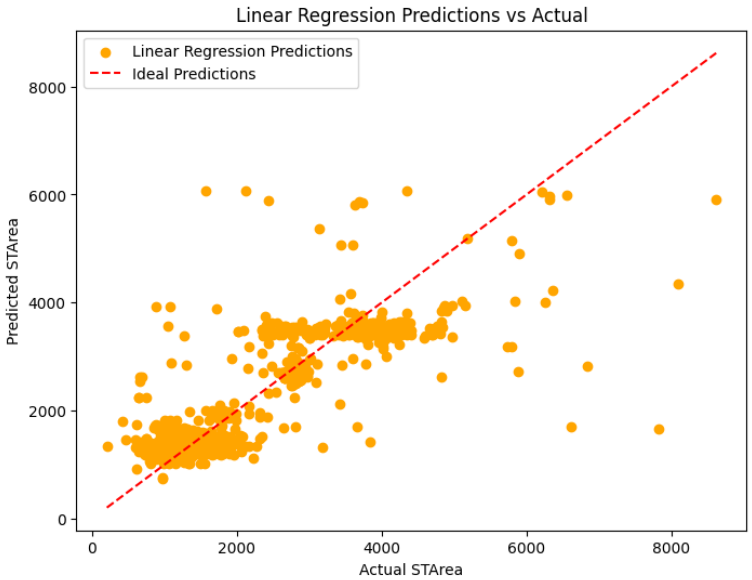
Model Fitting

- So we all know that we do model fitting so that our predicted values differ least from the true values of the unseen data or test data.
- So here in our analysis we have used different regression and classification techniques for model fitting.
- So in our main problem to predict the **STArea** of the new unseen data, we will consider that variable as the target variable. So as we have previously visualized, columns like **STArea and STLength have continuous values** despite discrete classes.So for these we use **Regression**.
- While if we look for the **boolean columns**, they can be easily classified in 0 or 1 class.So they can be classified into discrete classes. So for these columns, we use a **classification problem.**

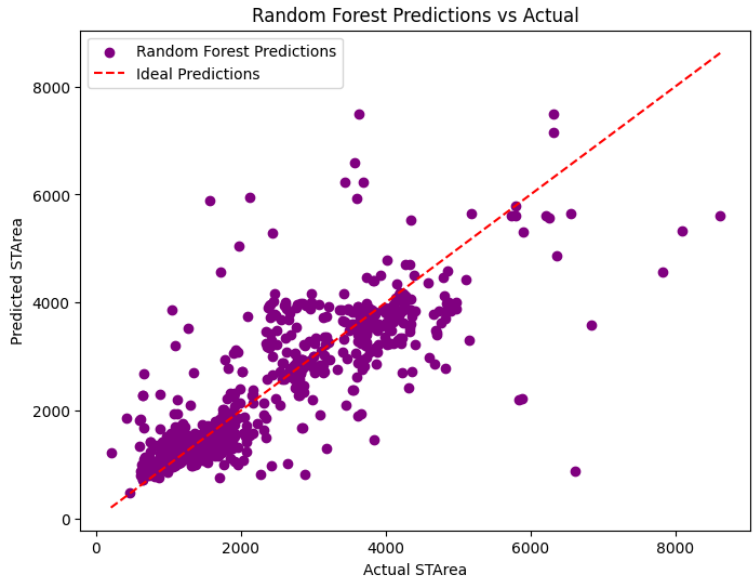
Regression Problem: Predicting STArea (Main Problem)

- So as discussed in Feature Selection Technique ,we have used STArea as target variable and then applied Mutual Information Feature Selection Technique for selecting those features with **top 20 percentile** values of Mutual Information with STArea.
- So there are **44 features** which satisfy these conditions.
- So to predict unseen data, we will use these **44 features** only.
- In our analysis, we have predicted STArea's test data by 2 regression techniques namely **Linear Regression** and **Random Forest Regression.**
- After that we have used the R2 score measurement(evaluation metric) to see which model predicted the best.
- We have plotted the predicted values vs true values graph for linear regression and random forest regression problem.

Regression Problem: Predicting STArea (Main Problem) - Continued



(a) Linear Regression Prediction



(b) Random Forest Regression Prediction

Figure 18: Predicted vs Values for Linear and Random Forest Regression

Regression Problem: Predicting STArea (Main Problem) - Continued

```
Linear Regression : R2 score: 0.6973729153929771
 Random Forest Regression : R2_Score 0.7158996549844858
```

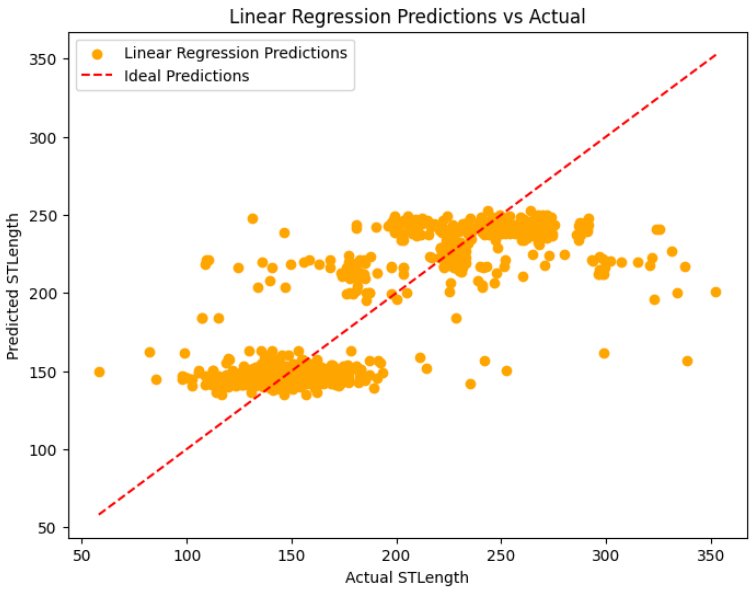
Figure 19: R2 score for linear and random forest regression

We can see that **R2 score for the random forest regressor is near or close to 1** as compared to Linear Regression . So we can say that Random Forest Regression outperforms Linear Regression.

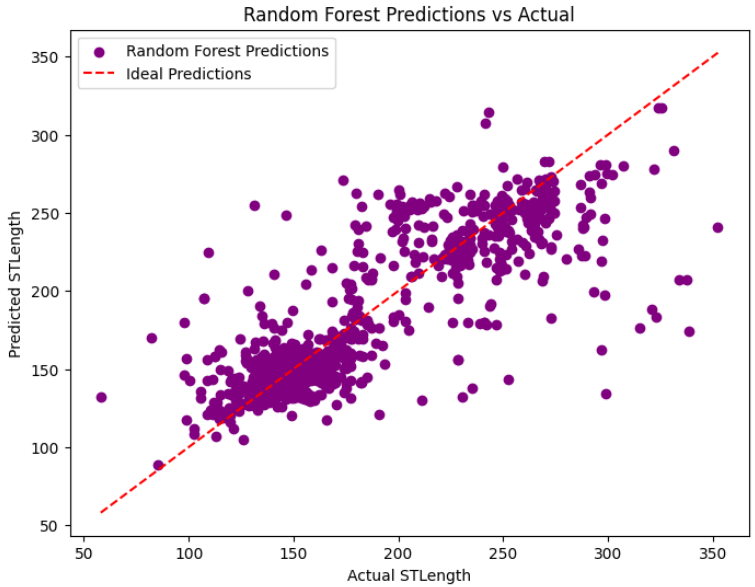
Regression Problem: Predicting STLength

- So as discussed in Feature Selection Technique,we have used STLength as the target variable and then applied Variance Thresholding Feature Selection Technique for selecting those features with variance threshold set to be 0.1.
- So there are **19 features** that satisfy these conditions.
- So to predict unseen data, we will use these **19 features** only.
- Again we have used **Linear Regression** and **Random Forest Regression** to predict the unseen values of STLength.
- After that we have used the R2 score measurement(evaluation metric) to see which model predicted the best.
- We have plotted the predicted values vs true values graph for linear regression and random forest regression problem.

Regression Problem: Predicting STLength - Continued



(a) Linear Regression Prediction



(b) Random Forest Regression Prediction

Figure 20: Predicted vs Values for Linear and Random Forest Regression

Regression Problem: Predicting STLength - Continued

```
Linear Regression : R2 score: 0.7105962149189429
 Random Forest Regression : R2_Score 0.7273728941995254
```

Figure 21: R2 score for linear and random forest regression

We can see that for this also **R2 score for the random forest regressor is near or close to 1** as compared to Linear Regression. So we can say that Random Forest Regression outperforms Linear Regression.

Classification Problem: Predicting BASKETBALL boolean column

- As we are dealing with the boolean variables, we know that they can be clearly classified into **2 classes true and false or 0 and 1.**
- So here as discussed we have predicted the **boolean column named BASKETBALL**.Note here we have not used any feature selection technique.
- In this we have 3 classification techniques or 3 classification methods namely **Logistic Regression , Random Forest Classifier and SVM(Support Vector Machines).**
- So we have used various evaluation metrics for these classification problems namely F1 score,accuracy,precision and recall.

Classification Problem: Predicting BASKETBALL boolean column - Continued

```
Model: Logistic Regression
Accuracy: 0.9723
Precision: 0.9262
Recall: 0.9718
F1 Score: 0.9485

Model: Random Forest Classifier
Accuracy: 0.9750
Precision: 0.9269
Recall: 0.9824
F1 Score: 0.9538

Model: Support Vector Machine
Accuracy: 0.9695
Precision: 0.9169
Recall: 0.9718
F1 Score: 0.9436
```

Figure 22: Accuracy,Precision,Recall and F1 Score for all 3 classification techniques

We can see clearly from the evaluation metrics that Random Forest Classifier dominates Logistic Regression and SVM in almost every evaluation metrics.

Group Contribution

All the three members have contributed equally to the EDA Final Project. Total Work was divided into three parts and each member was given a certain time period to complete the task. In this way the project was carried out phase wise within one month.

Member 1 Introduction and how big is the data (shape,etc.), Visualisations

Member 2 Model Training, Statistical Analysis and Data Cleaning

Member 3 Defining Problems and other queries, Data Collection, Feature Engineering, Data Transformation

THANK YOU