Practical Machine Learning

DA 224, 2022 batch

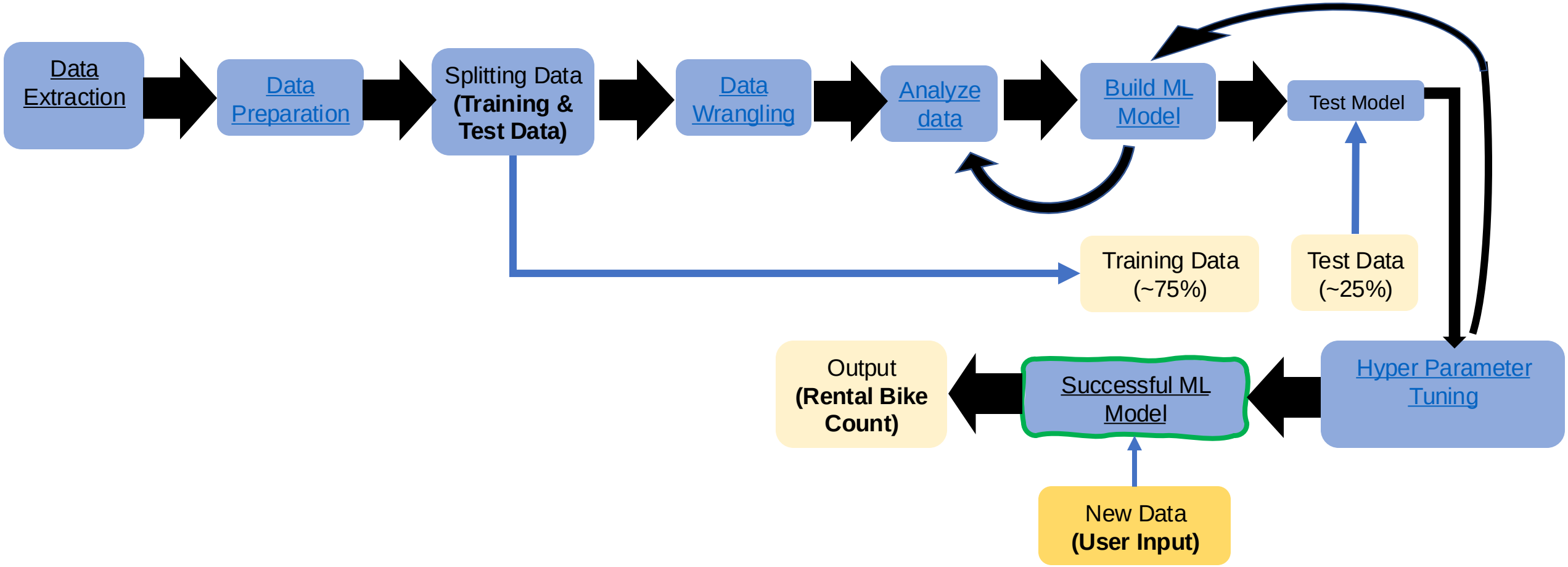
Mini Project - 1

15 August 2024

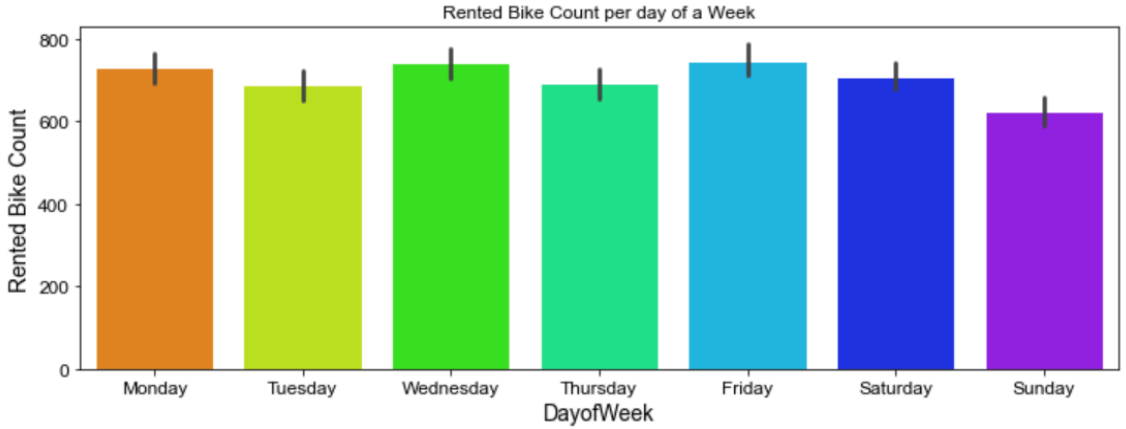
Anisha
Anwesha Mohanty
Arkadip Basu
Darakshan Jamal
Parth Dhir
Naveen R
Yathish Reddy

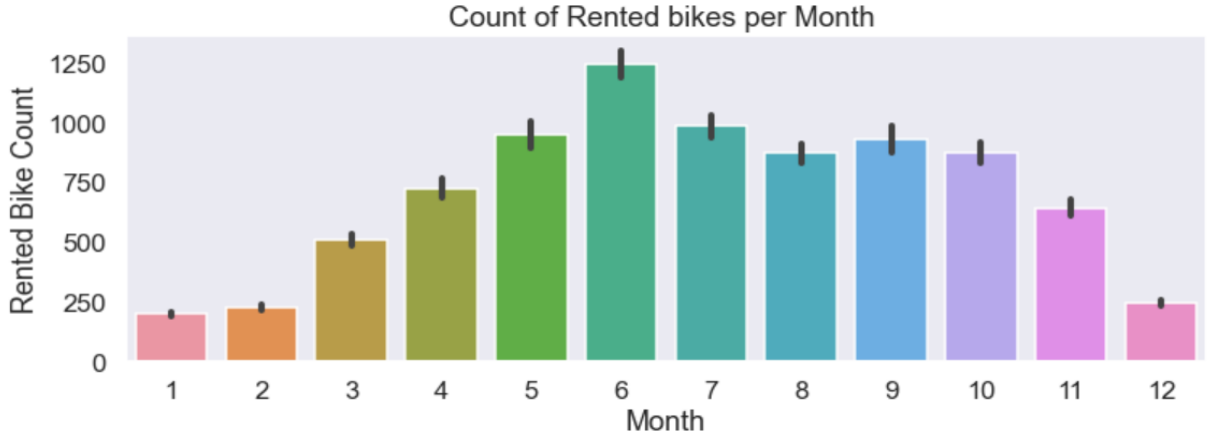

vélo'v

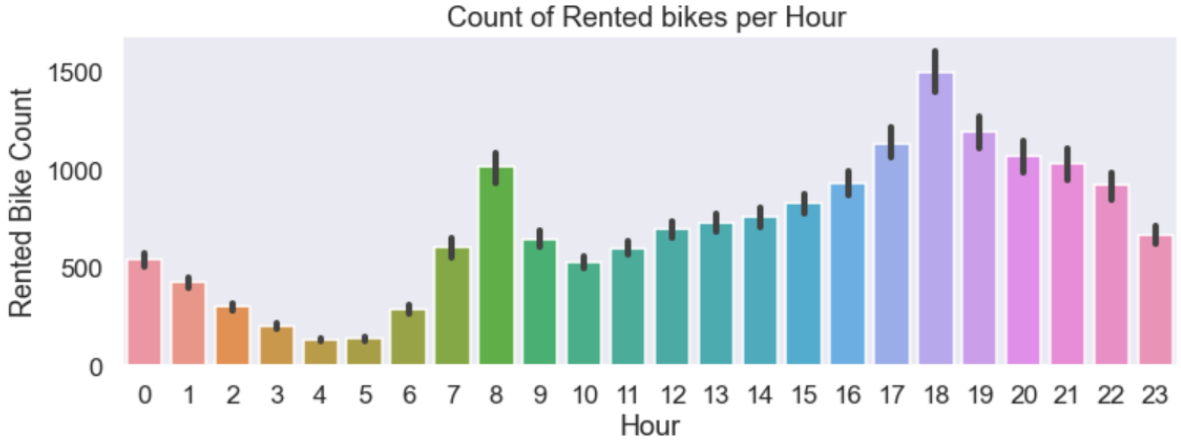
Process Flow

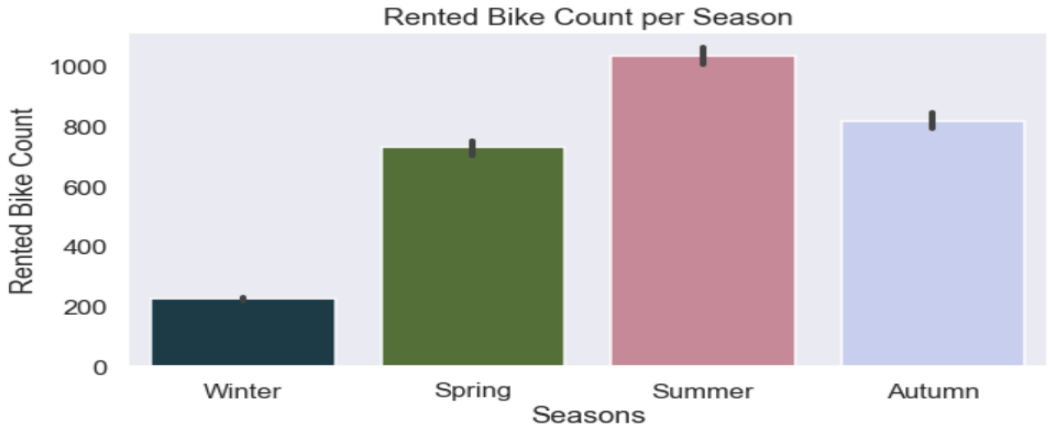


Data Wrangling (EDA)



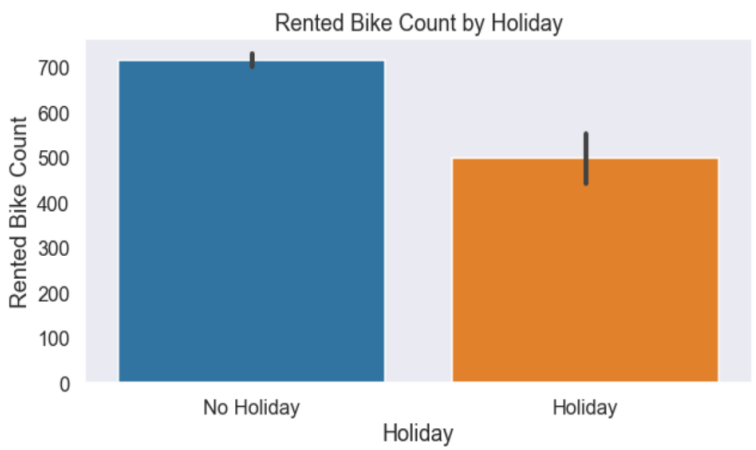


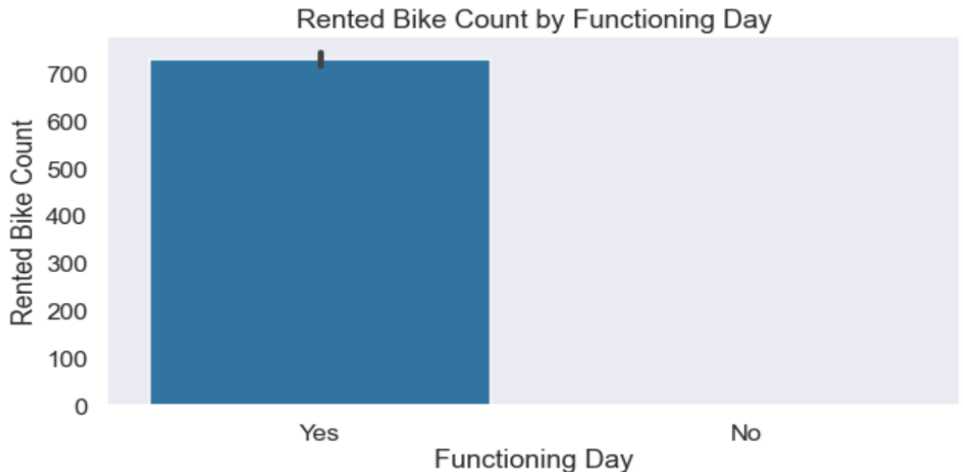




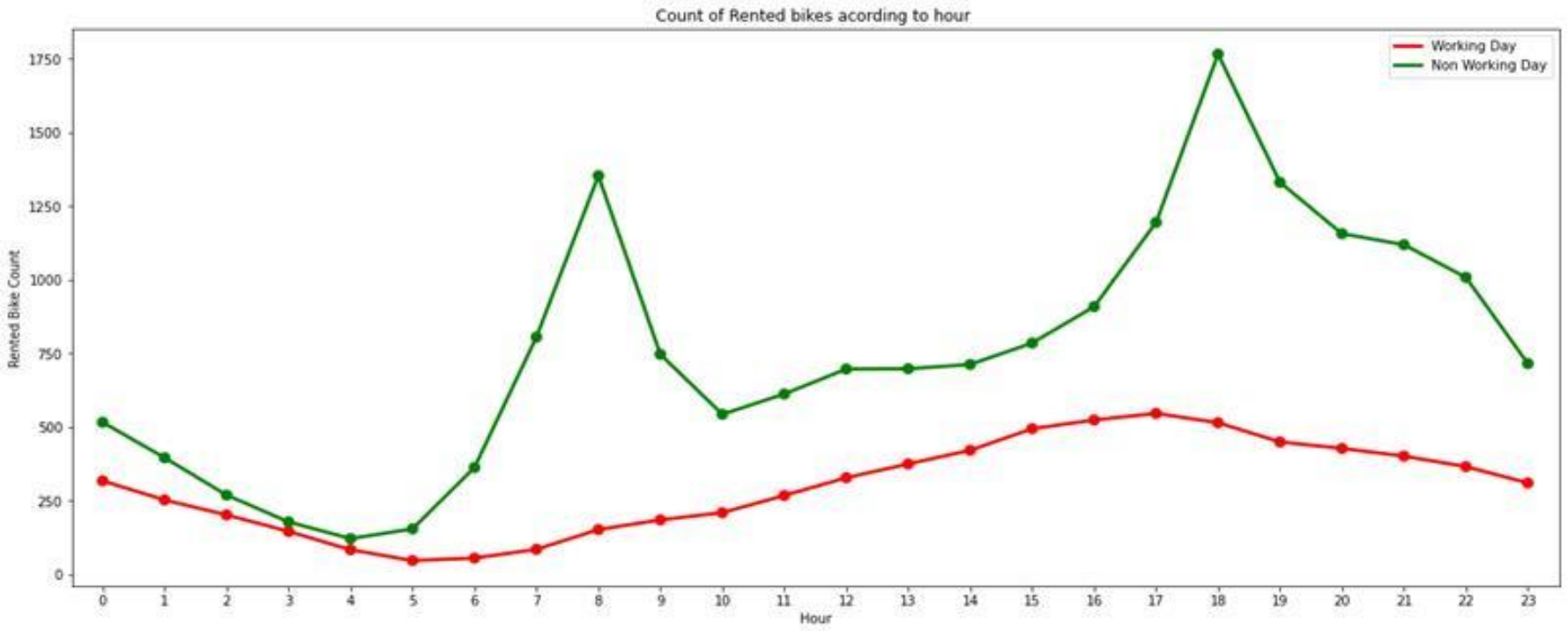
- Bike Count is higher in weekdays as compared to weekends
- Bike demand is more during the months of 5-10 (May – October) as compared to other months
- Demand for bikes is higher from 7 am-9 am and 5 pm-7 pm
- Bike demand is higher during Summer season

Data Wrangling (EDA)



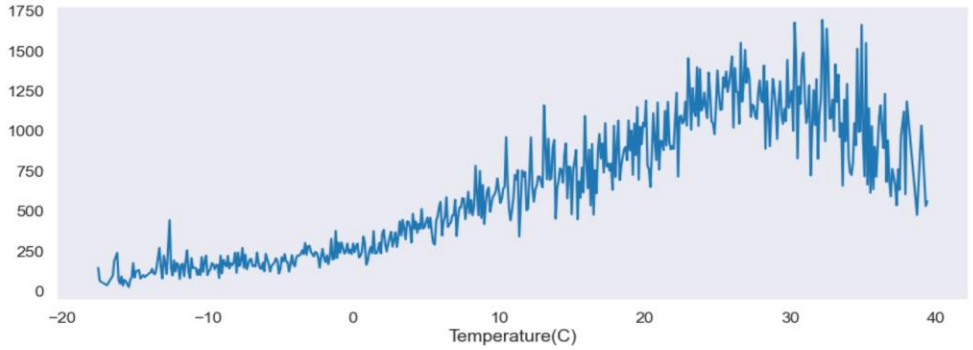


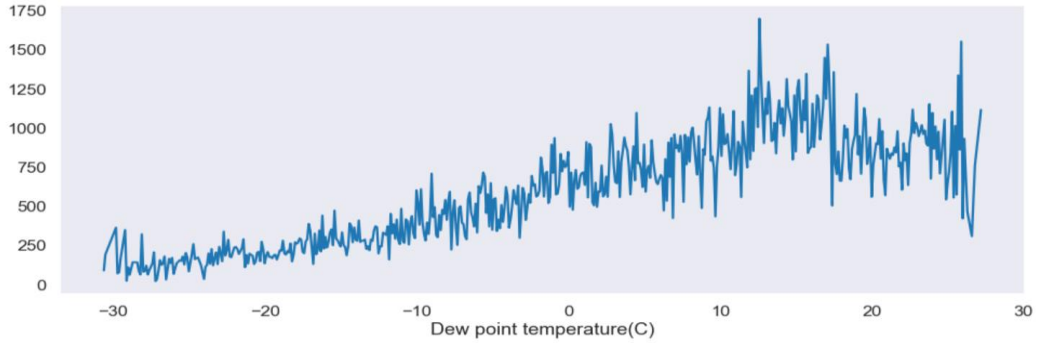
- During non-holidays bike count is high.
- On a non-functional day, there is zero demand for bikes.

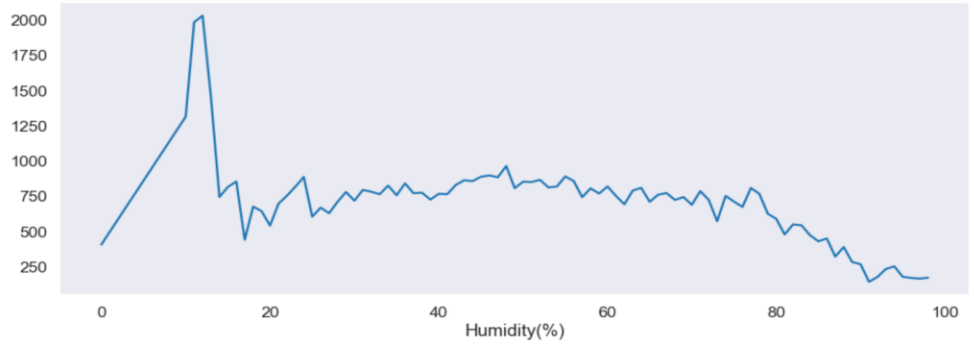


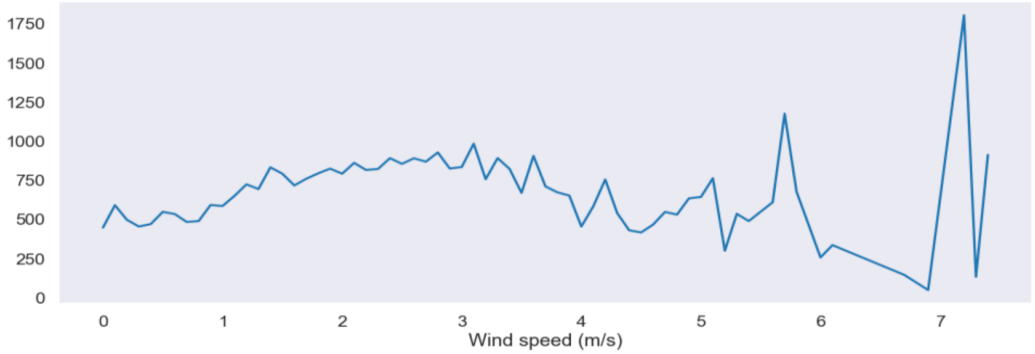
- On any Working day, the demand is generally high around 06:00 am to 10:00 am & From 04:00 pm to 08:00pm.
- On Non-Working day, the demand increases slowly around 11:00 am and saturates by 07:00 pm.

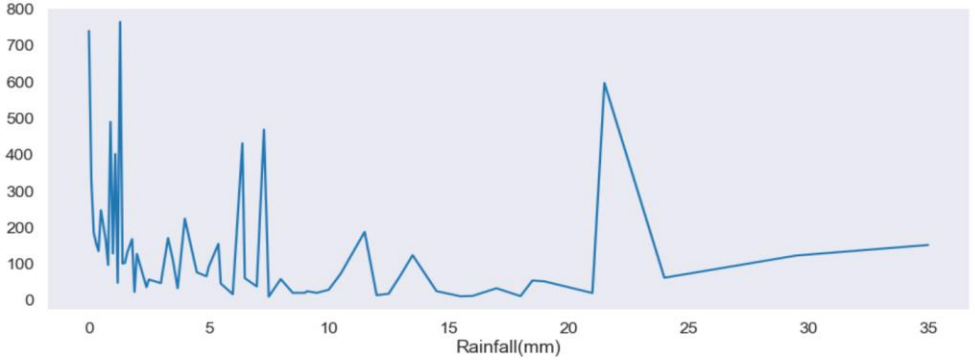
Data Wrangling (EDA)

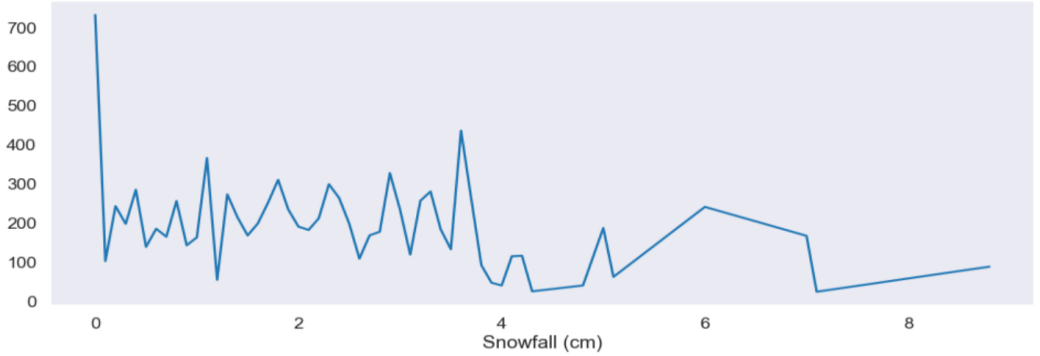








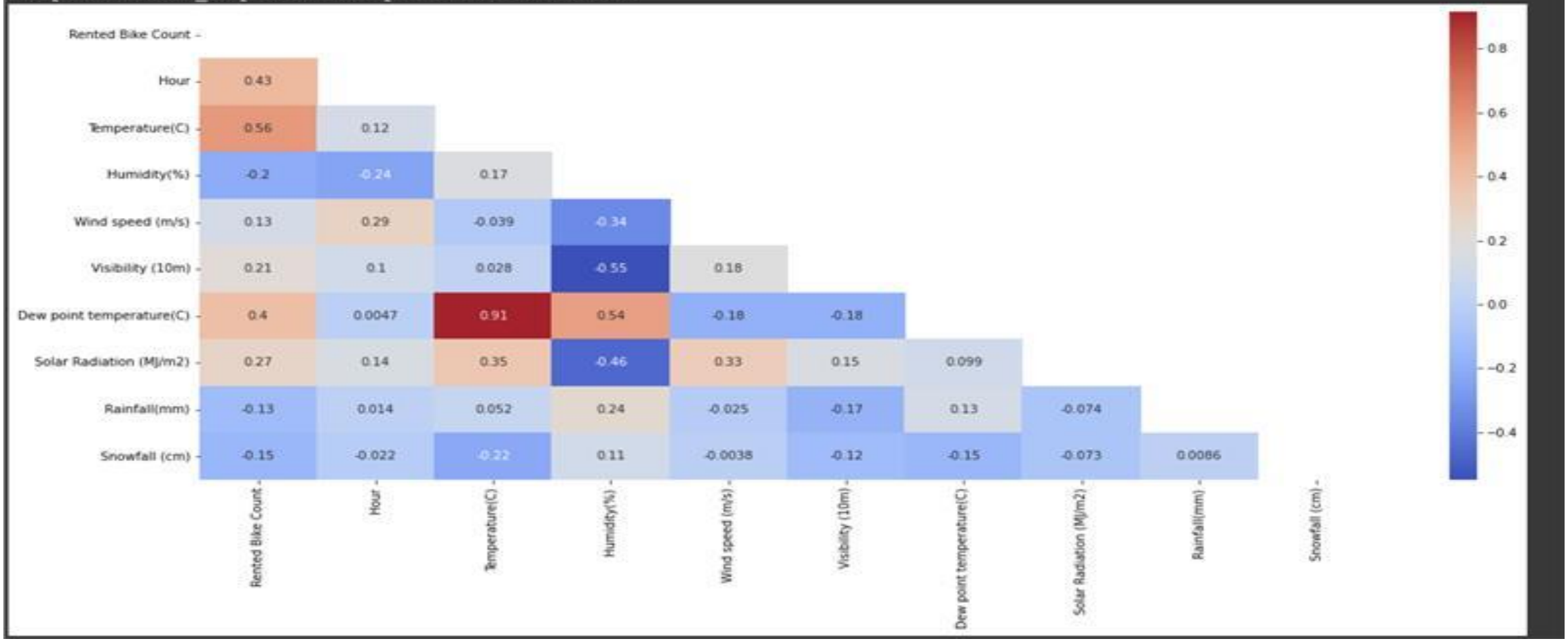




Correlation Matrix

```
plt.figure(figsize=(20,8))
correlation=rental_bikes_copy.corr()
mask = np.triu(np.ones_like(correlation, dtype=bool))
sns.heatmap((correlation),mask=mask, annot=True,cmap='coolwarm')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3b7b50e810>
```



Conclusion of Data Analysis

- As deduced from EDA process, bike count is 0 on a non-functional day, irrespective of other variables.
- Business rule employed: Records of non-functional days removed from analysis and model building process.
- Temperature column had missing values: Out of 8760 records, 500 were missing.
- Employed different Imputation methods:
  - Simple imputation - Imputation by median value
  - KNN Imputation - Imputation by K nearest neighbors
  - Regression with time-Imputation by regressing Temperature against time variable(Hour)
  - Regression with weather variables - Imputation by regressing Temperature against other weather variables

Imputation

Temperature Models		
ML Model	Train RMSE	Test RMSE
Median	17.605	17.442
KNN (n=3)	1.062	1.622
LR with weather	1.29	1.234
LR with time	5.301	5.427

- For temperature imputation, Linear regression with weather variable performed best
- Weather variable has dew point temperature as one of the predictor which has high correlation with temperature

Temperature Imputation Methods					
ML Model	Methodology	Train RMSE	Test RMSE	Train R2 Score	Test R2 Score
Linear Regression	Temperature imputation-SimpleImputation(Median)	429.040	413.366	0.559	0.565
Linear Regression	Temperature imputation-KNN(n_neighbors=3)	429.073	413.355	0.559	0.564
Linear Regression	Temperature Imputation-Linear Regression of time variable	429.055	413.309	0.559	0.565
Linear Regression	Tempertaure Imputation-Linear Regression using other Weather Variables	428.696	428.675	0.554	0.558
Linear Regression with Polynomial Features	Temperature imputation-SimpleImputation (Median)	278.479	287.885	0.809	0.811
Linear Regression with Polynomial Features	Temperature imputation-KNN (n_neighbors=3)	272.147	282.764	0.818	0.818
Linear Regression with Polynomial Features	Temperature Imputation-Linear Regression of time variable	273.816	283.849	0.815	0.817
Linear Regression with Polynomial Features	Tempertaure Imputation-Linear Regression using other Weather Variables	272.155	282.875	0.818	0.818
KNN	Temperature imputation-SimpleImputation (Median)	290.716	319.686	0.798	0.740
KNN	Temperature imputation-KNN (n_neighbors=3)	334.482	366.791	0.732	0.657
KNN	Temperature Imputation-Linear Regression of time variable	289.713	319.115	0.799	0.741

Feature Engineering & Encoding

```
rental_bike_without_precipitation = rental_bikes_copy.copy()
rental_bike_precipitation = rental_bikes_copy.copy()
rental_bike_precipitation['Precep'] = np.where( (rental_bike_precipitation['Rainfall(mm)']>0) | (rental_bike_precipitation['Snowfall (cm)']), 1, 0 )
rental_bike_precipitation['rain_bool'] = np.where(( rental_bike_precipitation['Rainfall(mm)']>0), True, False )
rental_bike_precipitation['snow_bool'] = np.where( (rental_bike_precipitation['Snowfall (cm)']>0), True, False )

rental_bike_precipitation.head()
```

	Rented Bike Count	Hour	Temperature(C)	Humidity(%)	Wind speed (m/s)	Visibility (10m)	Dew point temperature(C)	Solar Radiation (MJ/m2)	Rainfall(mm)	Snowfall (cm)	Seasons	Holiday	dayofweek	month	date_day	Peak_Hour	Precep	rain_bool	snow_bool
0	254	0	-5.2	37	2.2	2000	-17.6	0.0	0.0	0.0	Winter	No Holiday	4	12	1	False	0	False	False
1	204	1	-5.5	38	0.8	2000	-17.6	0.0	0.0	0.0	Winter	No Holiday	4	12	1	False	0	False	False
2	173	2	-6.0	39	1.0	2000	-17.7	0.0	0.0	0.0	Winter	No Holiday	4	12	1	False	0	False	False
3	107	3	-6.2	40	0.9	2000	-17.6	0.0	0.0	0.0	Winter	No Holiday	4	12	1	False	0	False	False
4	78	4	-6.0	36	2.3	2000	-18.6	0.0	0.0	0.0	Winter	No Holiday	4	12	1	False	0	False	False

- Day_of_week, Month, Date_day
- Peak_Hour
- Precipitation, Rain_bool, Snow_bool

One_Hot_Encoding for Encoding the categorical features & Cyclic encoding of Time based variables has been used.

Model Creation

ML Model	Methodology	Train RMSE	Test RMSE	Train R2 Score	Test R2 Score
KNN	KNN Model	192.36	249.63	0.9105	0.848
Linear Regression	Linear Regression without Polynomial Features	396.59	405.78	0.6125	0.6251
Linear Regression with Polynomial Features	Linear Regression with 225 Polynomial Features	272.12	282.87	0.8175	0.8178
Elastic Net with Polynomial Features	Elastic Net with Polynomial Features	263.04	273.87	0.8295	0.8292

- Linear regression model which gave us very high RMSE score on both Train & Test data
- Linear regression with Polynomial features of multiple degrees where 225 predictor variables was the optimal point which performed better than Linear regression model.
- K Nearest neighbor with neighboring parameter as 4. From here we got the best RMSE.

Model Creation with seasons

Seasonal Models					
Season	Methodology	Train RMSE	Test RMSE	Train R2 Score	Test R2 Score
Autumn	KNN Model with Precepetation coulmn being added	254.058	374.734	0.8293	0.6397
Spring	KNN Model with Precepetation coulmn being added	212.663	312.218	0.8822	0.7422
Summer	KNN Model with Precepetation coulmn being added	264.833	415.275	0.8555	0.615
Winter	KNN Model with Precepetation coulmn being added	73.463	117.408	0.76614	0.348

Autumn	Linear Regression without Polynomial Features	339.541	341.012	0.709	0.6343
Spring	Linear Regression without Polynomial Features	351.036	344.154	0.6777	0.6905
Summer	Linear Regression without Polynomial Features	401.076	408.642	0.6633	0.6413
Winter	Linear Regression without Polynomial Features	93.01	99.721	0.6042	0.609
	Seasonal RMSE Weighted Average	295.599	297.868		
	Linear Regression without Polynomial Features (Season stratification)	399.286	395.374		

Model Creation with precipitation/no precipitation

Precip / No Precip Models					
ML Model	Methodology	Train RMSE	Test RMSE	Train R2 Score	Test R2 Score
Linear Regression with Polynomial Features	Linear Regression without Polynomial Features (precipitation==True)	426.792	370.966	0.566	-3.133
Linear Regression with Polynomial Features	Linear Regression without Polynomial Features (precipitation==False)	195.2	164.163	0.301	0.191
	RMSE Weighted Average	220.117	186.472		
	Linear Regression without Polynomial Features (Precipitation stratification)	426.792	370.966		
KNN	KNN Regression with precipitation column present as predictor	214.3	280.13	0.888	0.808
KNN	KNN Regression without precipitation column present as a predictor	199.62	255.04	0.903	0.841
Linear Regression	Linear Regression without precipitation column present as a predictor	408.877	414.294	0.588	0.609
Linear Regression	Linear Regression with precipitation column present as a predictor	401.729	409.809	0.602	0.618
Linear Regression with Polynomial Features	Linear Regression with Polynomial Features without precipitation column preser	276.083	278.301	0.812	0.824
Linear Regression with Polynomial Features	Linear Regression with Polynomial Features with precipitation column present a	267.139	279.364	0.824	0.822
K Neighbors Regression	K Neighbors Regression with precipitation=True	148.04	134.14	0.611	0.448
K Neighbors Regression	K Neighbors Regression with precipitation=False (No Precipitation)	211.08	280.09	0.893	0.807
K Neighbors Regression	K Neighbors Regression Weighted score with Precipitation	204.15	134.14	0.862	0.768

Precipitation is inversely correlated with Rented Bike Count (-0.3). This makes precipitation a strong predictor for the model

Hyper Parameter Tuning

- Temperature imputation with KNN works best with neighbor = 3
- KNN for rental bike regression, works best with neighbor = 4
- Polynomial feature works best with 225 quadratic features added.
- For Elastic Net, L1_ratio = 1 and alpha =0.1

Colab Links

- **Final -** https://colab.research.google.com/drive/1mZcbZ9VuN3jkpK83VRPWcCWS3814ZojN?authuser=1#scrollTo=y-nZav3bqR5d

- LR- Temperature Imputation with other weather variables: https://colab.research.google.com/drive/1RK9yJYwhfp-U4DvX2AnhsAZjK8sKwtGo#scrollTo=B9ttu1tZMZw8

- EDA - https://colab.research.google.com/drive/1Sw69LU1UzHE9_7lw6VpKd9XnKzE_GMQ1?usp=sharing

Thank you

Question, Feedback, Suggestions

Happy Biking

15 August 2024

Anisha
Anwesha Mohanty
Arkadip Basu
Darakshan Jamal
Parth Dhir
Naveen R
Yathish Reddy

DA 224, Team 5