Terminal Assignment Based Assessment-Statistics

JOHN MARUTHUKUNNEL JACOB
Department of Computing
National College of Ireland
Dublin, Ireland
x21138494@student.ncirl.ie

***Abstract*—The aim of this project is to investigate the performance of several machine learning algorithms by applying them to three different datasets. The datasets chosen for this project are London Bike Shares- to maximize inventory control of bike shares in stations, NFL outcome- to predict the outcome of NFL games to improve the odds of betting, Solar Radiation prediction- to decide if solar batteries owned will be reasonable power producers in the future. The ML algorithms are compared and the best model is selected for each dataset.**

***Index Terms*—Linear Regression, Random Forest, Decision Tree, XGBoost, Gauss Naive Bayes, Logistic regression**

I. Introduction

In the age of IOT and technologically advanced society the research and implementation of Machine Learning has helped us to interconnect every aspect of our lives from everyday activities to work. With the introduction of AI and ML, technology development has exponentially growed to allow applications and products to perform intelligently. The growing field of Machine Learning is very important in exploring historical data and developing real-world applications. In this project six different ML algorithms are used for predictive analysis. They are Linear Regression, Random Forest, Decision Tree, Logistic regression, Gauss Naive Bayes and XGBoost. The ML techniques are evaluated to find the best fitting model for each dataset.

II. RESEARCH OBJECTIVE

A. London Bike Share Dataset

Rental Bikes are being introduced in most urban cities for more mobility comfort and eco-friendly public transport. It is important for bike sharing systems to meet the demand of bike rentals and reduce the waiting period. The prediction of bike rentals in an hourly period is a challenge for most bike sharing systems. The London bike sharing dataset contains 17,415 observations(rows) of bike shares of every hour of every day from January 2015 to January 2017. The project makes use of the file 'london_merged.csv'.

Research question:
• Can we predict the future bike shares?

ML Algorithms used:
• Linear Regression
• Decision Tree Regressor

B. NFL Dataset

With the growth of available historical data in the sports industry presents opportunity for research to gain a better understanding of the sport dynamics. The NFL is one of the most revenue generating industry in the USA. Aside from the direct revenue, the betting companies are generating huge profits from this sport. The NFL scores and betting dataset contains 13,233 results of NFL games from 1966-2022. It is important to note that this project makes use of the file called "spreadspoke_scores.csv" and "nfl_teams".

Research qn:
• Can we predict the outcome of a NFL game?

ML Algorithms used:
• Logistic Regression
• Gauss Naive Bayes
• XGBoost

C. Solar Radiation Dataset

The prediction of global solar radiation is important for directing solar energy conversion systems, choosing the best regions for solar energy, and making future investment decisions. The third data set is a solar radiation prediction data procured from NASA. As more companies switch to eco-mode and investing in renewable energy production, solar energy still plays a key role in electricity production. This project will predict the solar radiation using ML techniques and help in deciding if solar batteries owned will be reasonable to use in the future. The Solar radiation prediction dataset contains 32,687 records of solar data from September 2016-December 2016. This project makes use of the file called "SolarPrediction.csv".

Research qn:
• Can we predict the solar radiation? • Which are the most important factors predicting the solar radiation?

ML Algorithms used:
• Linear Regression
• Random Forest

III. RELATED WORK

1. Research paper [1] predicts the hourly bike rental demand and date information using data pertaining to weather

information (Humidity, Windspeed, Temperature, Visibility , Dewpoint, Solar Radiation, Snowfall, Rainfall). The paper also explores the possibility of filtering insignificant features that are not predictive and ranks the features based on prediction significance. The regression models used are (a) Linear Regression (b) Gradient Boosting Machine (c) Support Vector Machine (Radial Basis Function Kernel) (d) Boosted Trees, and (e) Extreme Gradient Boosting Trees. Grading Boosting Machine was found to be the best model with R2 value of 0.96 in the training set and 0.92 in the test set.

2. The predictive modelling in Research paper [2] develops models for modeling the availability of bikes in the San Francisco Bay Area Bike Share System (BSS) applying machine learning at two levels: network and station. The regression models used are Random Forest and Least-Squares Boosting for prediction at the station level and Partial Least Squares Regression at the network level. Predictions errors were found to be lower for univariate models while network level prediction was found to be promising for networks with large number of stations.

3. To predict the bike demand for rentals and returns in the city of Thessaloniki, Greece various Machine Learning models were tested in Research paper [3]. The regression models used in this paper are Gradient Boosting, XGBoost, Random Forest and Neural Network. In the train set Gradient Boosting had the highest R2 and least RMSLE for Random Forest while in test set Gradient Boosting was found to be the best model.

4. In Research paper [4] predictive models for buike sharing systems in smart cities is optimized through the use of Machine Learning techniques and IOT. The features such as spatial distribution of bikes, time, weather conditions, seasons are used to build the predictive model. The regression models used in this paper are Random Forestt regressor, Bagging regressor, XGBoost regressor, AdaBoosting regressor, SVM and logistic regressor. Random Forest and Bagging regressors were found to produce the best predictive models followed by XGBR, then ABR. SVM and LR produced the lowest R2 score to give the least accurate model.

5. To understand the variations in demand for bike sharing systems in London an exploratory analysis was done in Research paper [5]. The relation between unemployment rate and bike sharing was investigated mainly. Other features included in the analysis was weather features, temporal features (e.g., day of the week) and number of docking stations. A generalized negative binomial model was used in the paper and it was found unemployment rates was negatively associated with bike rentals along with rainfall, humidity and wind.

6. Research paper [6] uses Dynamic Linear Models to predict the bike counts in a Bike Sharing System in San Francisco Bay Area. First and second order polynomial models were used for prediction and it was found that they are able to predict with a prediction error of 0.37 bikes/station for 15-min prediction period, 1.1 bikes/station for 2 hours prediction period. DLM models were found to be comparable to a random forest model for 15 and 30 min prediction periods.

7. The goal of research paper [7] was to predict the outcome of a One Day International cricket match. The several factors included for prediction are home game advanatage, day/night, toss, innings, physical fitness of teams and dynamic startegies. The models used in this paper are Naïve Bayesian, Random Forest and SVM. All three models produced nearly comparable accuracies with SVM performing better. However it was found that in the case of severe class imbalance Random Forest and SVM fail to perform.

8. In Research paper [8] Bayesian Networks were contrasted with alternative ML techniques(MC4, a decision tree learner; Naive Bayesian learner; Data Driven Bayesian and a K-nearest neighbour learner) for predicting the match outcomes(win, lose or draw) for Tottenham Hotspur FC. The expert BN was found to superior to other models and it showed promise in the ability to predict without requiring much learning data.

9. A combined system of principal components analysis, nonparametric statistical analysis, a support vector machine (SVM), and an ensemble machine learning algorithm to predict whether a hockey team will win a game is explored in research paper [9]. The SVM classifier performed with a accuracy level over 90%. The paper also discusses the limitation of prediction game outcomes for a period over a year due to change in players, coaching staff and team management in teams.

10. The Research paper [10] provides an analysis of literature in ML, focusing on the application Artificial Neural Networking in sports results prediction. The features used in this paper comprises of historical match results, player performance indicators and opposition information. A sport result prediction 'SRP-CRISP-DM' framework is proposed for the complex problem of sports outcome prediction. The paper also discusses the lack of ML techniques to predict outcomes accurately due to the nature of variability in the sport.

11. In Research paper[11] different set of classifiers were implemented to predict football game outcomes and other betting features. Gaussian Naïve Bayes, Logistic Regression and XG Boost are the models used for prediction of game outcomes. Gausse Naïve Bayesian classifier produced better prediction performance, however none of the ML classifiers were able to beat the bookmakers.

12. The prediction accuracy of temperature based solar

radiation models in forecasting the daily global solar radiation for 6 climatic zones of Morocco is explored in research paper [12]. SVM, Decision Tree, Linear Regression, Gaussian Regression are the 4 ML models employed to improve performance of initial models. The ML models were able to improve the R2 from 0.8 to 0.95 for all skies and 0.95 to 0.98 for clear skies.

13. The Research paper [13] explores ML techniques to improve short term accuracy of solar radiation. Exploratory analysis was done to show the significance and relationship between various variables. Logistic regression, Random forest and Decision Tree models were implemented, out of which Decision Tree has shown maximum accuracy.

14. Support Vector Machines, Generalized Linear model and Random Forest models are compared in research paper [14] to predict the solar radiation in 8 different Chinese cities, representing different pollutant and geoclimatic conditions. It was observed that the solar radiation prediction was closely related to weather and pollution condition levels. The SVM model performed better in radiation prediction under slight pollution and stable weather conditions.

15. The aim of research paper [15] is to predict the global solar radiation of 4 provinces in Turkey. The Machine Learning Algorithms used in this paper are support vector machine (SVM), artificial neural network (ANN), kernel and nearest-neighbor (k-NN), and deep learning (DL). The features used for prediction are daily minimum and maximum ambient temperature, cloud cover, daily extraterrestrial solar radiation, day length and solar radiation of these provinces. It was found that all ML algorithms tested in this paper can be used to predict global solar radiation data with high accuracy.

16. In the Research paper [16] several machine learning models were developed and evaluated for real-time and short-term solar energy forecasting to ensure optimized management. The dataset used in this paper had data from 2016 to 2018 for the region of Errachidia, Morocco. Pearson correlation coefficient was used to identify the most significant input features. It was observed Random Forest and ANN provided higher accuracies against Linear Regression and SVM.

IV. DATA MINING METHODOLOGY

For a Data Mining and Machine Learning Project, a study must take a streamlined strategy. A pre-determined strategy ensures that no crucial tasks are overlooked while machine learning models are being created. KDD and CRISP-DM are the most popular Data Mining Frameworks employed. CRISP-DM is the framework used in this project. It serves as a roadmap for planning, organizing, and executing a DMML project. The CRoss Industry Standard Process for Data Mining (CRISP-DM) is a six-phase process paradigm, which are listed below:

1. Business Understanding
2. Data Understanding
3. Data Preparation
4. Modelling
5. Evaluation
6. Deployment

A. DATASET 1-LONDON BIKE SHARING

Business Understanding

The first step, business understanding, focuses on understanding the project objectives and needs from a business standpoint, then transforming that knowledge into a data mining problem statement and a preliminary plan to fulfill the goals.In the first dataset predictive models of london bike shares is created using ML algorithms. For maximum efficiency and least waiting time period for customers, the inventory at each station should meet the demand.Through the use of ML algorithms we can efficiently manage inventory control to improve customer satisfaction, which is necessary for any business growth.The research question for this data set is given below:
Can we predict the future bike shares?

Data Understanding

Once the research question is established we have to explore the data and understand the features present in the data.The dependent and independent variables are identified and data quality is checked. The dataset is shown in fig.1. The first dataset selected is sourced from the below kaggle link:
https://www.kaggle.com/datasets/hmavrodiev/london-bike-sharing-dataset

	timestamp	cnt	t1	t2	hum	wind_speed	weather_code	is_holiday	is_weekend	season
0	2015-01-04 00:00:00	182	3.0	2.0	93.0	6.0	3.0	0.0	1.0	3.0
1	2015-01-04 01:00:00	138	3.0	2.5	93.0	5.0	1.0	0.0	1.0	3.0
2	2015-01-04 02:00:00	134	2.5	2.5	96.5	0.0	1.0	0.0	1.0	3.0
3	2015-01-04 03:00:00	72	2.0	2.0	100.0	0.0	1.0	0.0	1.0	3.0
4	2015-01-04 04:00:00	47	2.0	0.0	93.0	6.5	1.0	0.0	1.0	3.0

Fig. 1. Data Set 1

Data Preparation

In this stage the dataset is loaded into jupyter notebook and data cleaning and preperation is performed as shown in fig.2. The heatmap of the dataset is shown in fig.3. The data cleaning process includes checking null values, outlier treatment, scaling of variables and including dummy variables. The following steps were performed in this dataset:

- checked for null values and there was none present.
- blank strings were converted to NaN values.
- dropping duplicates records.
- timestamp variable was converted to year, month, day-ofweek, and hour.
- dummy variable were created for the variables weather_code,season,hour,month,dayofweek.

- Scaling of the variables t1, hum, wind_speed, cnt is done.
- the columns timestamp, year and t2 were dropped.

Fig. 2. Data Set Preprocessing

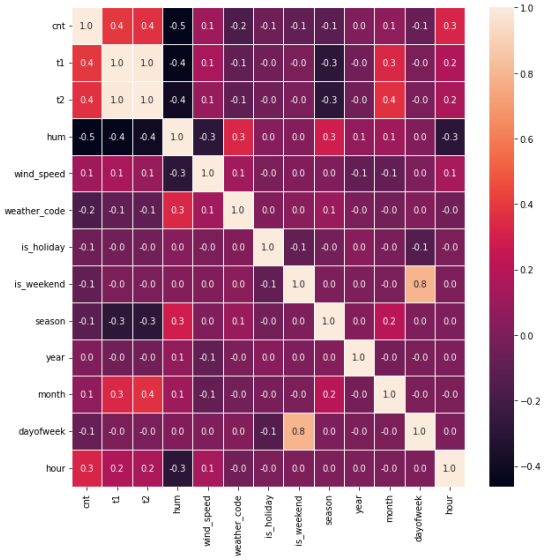


Fig. 3. Heatmap of Dataset 1

Data Modelling

Different machine learning algorithms are applied to the datasets after the preceding processes have been completed. The machine learning algorithm to choose is determined by the business requirement, dataset availability, and intended outcomes.The ML algorithms applied on this dataset are Linear Regression and Decision Tree.

1. Multiple Linear Regression

By fitting a linear equation to the available data, multiple linear regression analyses the relation between two or more independent variables and a continuous dependent variable.The cleaned dataset is splitted into training and testing data. The training data is used to create the linear regression model. Backward selection is used in this model creation. Initially a model was created with all variables in OLS method. Te next model was created after removing the insignificant variables($p>0.05$).

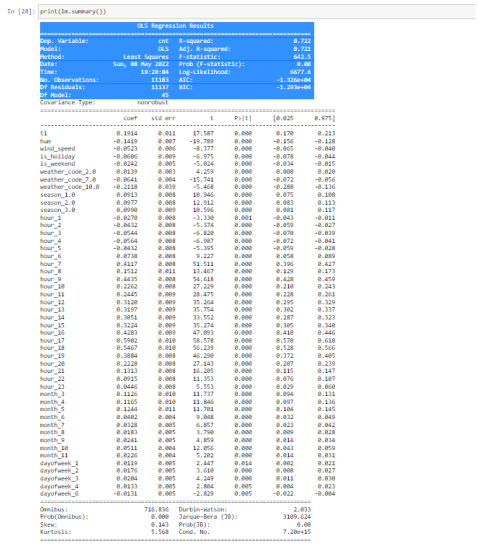
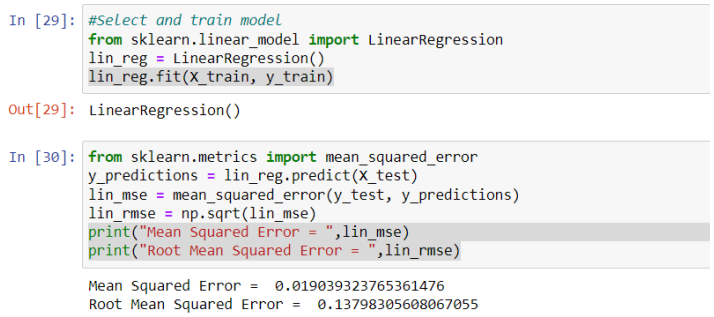


Fig. 4. Linear Regression summary

The above model in fig.4 has an adjusted R_2 value of 0.721.The model created has an accuracy of 72.1%.The same final data was used to create a Linear Regression model which has an RMSE value of 0.1379 as shown in fig.5.



```
In [29]: #Select and train model
         from sklearn.linear_model import LinearRegression
         lin_reg = LinearRegression()
         lin_reg.fit(X_train, y_train)

Out[29]: LinearRegression()

In [30]: from sklearn.metrics import mean_squared_error
         y_predictions = lin_reg.predict(X_test)
         lin_mse = mean_squared_error(y_test, y_predictions)
         lin_rmse = np.sqrt(lin_mse)
         print("Mean Squared Error = ",lin_mse)
         print("Root Mean Squared Error = ",lin_rmse)

Mean Squared Error =  0.019039323765361476
Root Mean Squared Error =  0.13798305608067055
```

Fig. 5. RMSE, MSE values of Linear Regression

The predicted vs actual values are plotted below in fig.6 for this model:

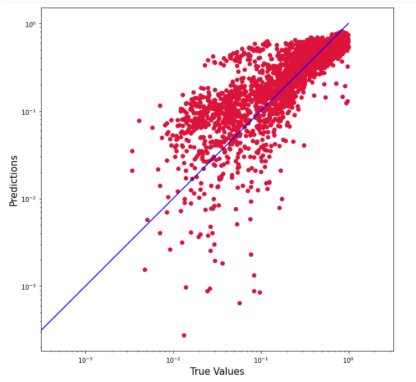


Fig. 6. Predicted vs Actual

2. Decision Tree Regressor

In the shape of a tree structure, a decision tree constructs regression or classification models. It incrementally cuts down a dataset into smaller and smaller sections while also developing an associated decision tree. The final data was used to create decision tree regressor model. The model had an RMSE value of 0.104 as shown in fig.7. The decision tree model is a better fit when comparing RMSE values.

```
In [35]: from sklearn.tree import DecisionTreeRegressor
         tree_reg = DecisionTreeRegressor()
         tree_reg.fit(X_train, y_train)

Out[35]: DecisionTreeRegressor()

In [36]: y_predictions = tree_reg.predict(X_test)
         tree_mse = mean_squared_error(y_test, y_predictions)
         tree_rmse = np.sqrt(tree_mse)
         print("Mean Squared Error = ",tree_mse)
         print("Root Mean Squared Error = ",tree_rmse)

         Mean Squared Error =  0.010957199107684374
         Root Mean Squared Error =  0.10467664069736081
```

Fig. 7. RMSE, MSE values of Decision Tree Regressor

The predicted vs actual values are plotted below in fig.8 for Decision Tree model:

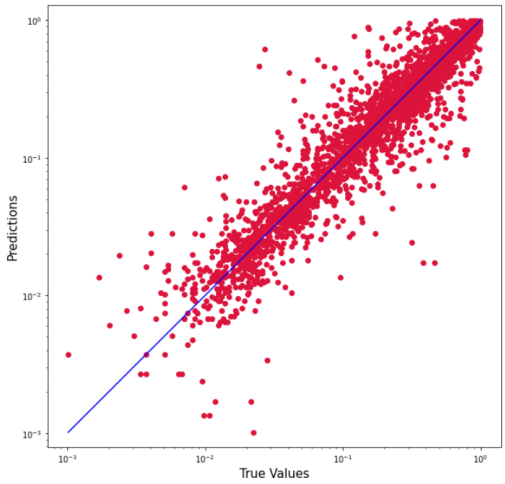


Fig. 8. Predicted vs Actual

B. DATASET 2-NFL

Business Understanding

In the second dataset predictive models of NFL match outcomes is created using ML algorithms.This predictive modeling is built on the idea of improving betting odds for NFL game outcomes. To build a better winning chance and economic growth on betting ML algorithms are used to build a efficient predictive model. The research question for this data set is given below:
Can we predict the outcome of a NFL match?

Data Understanding

The "spreadspoke_scores.csv" and "nfl_teams" datasets are used in this model building.The dataset "nfl_teams" is used to map the team_id and teamnames. The dataset is shown in fig.9. The second dataset selected is sourced from the below kaggle link:
https://www.kaggle.com/datasets/tobycrabtree/nfl-scores-and-betting-data?select=spreadspoke_scores.csv

schedule_dat	schedule_sea	schedule_wee	schedule_pla	team_home	score_home	score_away	team_away	team_favorite	spread_favor	over_under_li	stadium	stadium_neut	weather_tem	weather_win	weather_hum	weather_detail
9/2/1966	1966	1	FALSE	Miami Dolphi	14	23	Oakland Raiders				Orange Bowl	FALSE	83	6	71	
9/3/1966	1966	1	FALSE	Houston Oiler	45	7	Denver Broncos				Rice Stadium	FALSE	81	7	70	
9/4/1966	1966	1	FALSE	San Diego Chi	27	7	Buffalo Bills				Balboa Stadiu	FALSE	70	7	82	
9/9/1966	1966	2	FALSE	Miami Dolphi	14	19	New York Jets				Orange Bowl	FALSE	82	11	78	
9/10/1966	1966	1	FALSE	Green Bay Pac	24	3	Baltimore Colts				Lambeau Fiel	FALSE	64	8	62	
9/10/1966	1966	2	FALSE	Houston Oiler	31	0	Oakland Raiders				Rice Stadium	FALSE	77	6	82	
9/10/1966	1966	2	FALSE	San Diego Chi	24	0	New England Patriots				Balboa Stadiu	FALSE	69	9	81	
9/11/1966	1966	1	FALSE	Atlanta Falcon	14	19	Los Angeles Rams				Atlanta-Fulton	FALSE	71	7	57	
9/11/1966	1966	2	FALSE	Buffalo Bills	20	42	Kansas City Chiefs				War Memorial	FALSE	63	11	73	
9/11/1966	1966	1	FALSE	Detroit Lions	14	3	Chicago Bears				Tiger Stadium	FALSE	67	7	73	
9/11/1966	1966	1	FALSE	Pittsburgh Ste	34	34	New York Giants				Pitt Stadium	FALSE	64	5	70	
9/11/1966	1966	1	FALSE	San Francisco	20	20	Minnesota Vikings				Kezar Stadium	FALSE	60	25	75	
9/11/1966	1966	1	FALSE	St. Louis Card	16	13	Philadelphia Eagles				Busch Memor	FALSE	72	5	70	
9/11/1966	1966	1	FALSE	Washington R	14	38	Cleveland Browns				RFK Memorial	FALSE	65	8	52	
9/16/1966	1966	2	FALSE	Los Angeles R	31	17	Chicago Bears				Los Angeles M	FALSE	72	10	52	
9/18/1966	1966	3	FALSE	Buffalo Bills	58	24	Miami Dolphins				War Memorial	FALSE	61	3	67	

Fig. 9. Data Set 2

Data Preperation

In this stage the dataset is loaded into jupyter notebook and data cleaning and preperation is performed.The data cleaning process includes checking null values, outlier treatment, scaling of variables and including dummy variables as shown in fig. 10. The heatmap for the dataset is shown in fig.11. We use feature selection to select the most significant variables. The following steps were performed in this dataset:

- checked for null values and there was none present.
- blank strings were converted to NaN values, indexes reset and changed data types necessary.
- dropping duplicates records.
- mapping team_id to the correct teams
- created new columns home favorite and away favorite and filled with boolean values.
- named weeks were converted into numerical representation.
- dropping of insignificant columns.
- target variable HTW was created from columns score_home and score_away.

```
In [ ]: # # replacing blank strings with NaN
df = df.replace(r'^\s*$', np.nan, regex=True)

# # # removing rows from specific columns that have null values, resetting index and changing data types
df = df[(df.score_home.isnull() == False) & (df.score_away.isnull() == False) &(df.team_favorite_id.isnull() == False)]

df.reset_index(drop=True, inplace=True)

# # # mapping team_id to the correct teams
df['team_home'] = df.team_home.map(teams.set_index('team_name')['team_id'].to_dict())
df['team_away'] = df.team_away.map(teams.set_index('team_name')['team_id'].to_dict())


# # creating home favorite and away favorite columns (fill na with 0's)
df.loc[df.team_favorite_id == df.team_home, 'home_fav'] = 1
df.loc[df.team_favorite_id == df.team_away, 'away_fav'] = 1
df.home_fav.fillna(0, inplace=True)
df.away_fav.fillna(0, inplace=True)


# # stadium neutral and schedule playoff as boolean
df['stadium_neutral'] = df.stadium_neutral.astype(int)
df['schedule_playoff'] = df.schedule_playoff.astype(int)

#coverting named weeks to numerical weeks
df.loc[(df.schedule_week == 'Wildcard') | (df.schedule_week == 'WildCard'), 'schedule_week'] = '19'
df.loc[(df.schedule_week == 'Division'), 'schedule_week'] = '20'
df.loc[(df.schedule_week == 'Conference'), 'schedule_week'] = '21'
df.loc[(df.schedule_week == 'Superbowl') | (df.schedule_week == 'SuperBowl'), 'schedule_week'] = '22'
df['schedule_week'] = df.schedule_week.astype(int)

# # change data type of date columns
df['schedule_date'] = pd.to_datetime(df['schedule_date'])
```

Fig. 10. Data Set Preprocessing

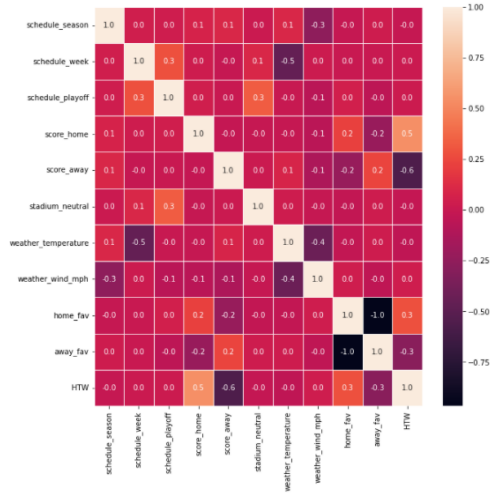


Fig. 11. Heatmap of Dataset 1

Data Modelling

The ML algorithms applied on this dataset are Logistic Regression, Gausse Naive-Bayes and XGBoost.

1. Logistic Regression

When the dependent variable is binary or dichotomous, a machine learning approach called logistic regression is applied.For classification situations, where we need to assess whether a new sample belongs in one of two categories, logistic regression is a valuable machine learning method. Here we are predictiong the outcome of a NFL match. The dependent variable HTW $(HomeTeamWin)$ is a variable containing 1 or 0 where 1 represents a home team win and vice versa.The logistic regression model had an accuracy of 68% as shown in below fig.

```
In [25]: #spliting the dataset into test and train
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
logreg = LogisticRegression()
logreg.fit(X_train, y_train)
D:\New folder\envs\rstudio-lib\site-packages\sklearn\utils\validation.py:993: DataConversionWarning: A column-vector y was pas
sed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
Out[25]: LogisticRegression()

In [26]: y_pred = logreg.predict(X_test)
print('Accuracy of logistic regression classifier on test set: {:.2f}'.format(logreg.score(X_test,y_test)))
accuracy_score(y_test, y_pred)
Accuracy of logistic regression classifier on test set: 0.68
Out[26]: 0.6785894206549118

              precision    recall  f1-score   support

           0       0.63      0.53      0.58       816
           1       0.71      0.78      0.74      1169

    accuracy                           0.68      1985
   macro avg       0.67      0.66      0.66      1985
weighted avg       0.67      0.68      0.67      1985
```

Fig. 12. Logistic Regression summary

The above model in fig.12 has an accuracy value of 0.68.The precision and recall values are shown in fig. The model created has an accuracy of 68%.The ROC curve of the model is shown in fig. 13.

Fig. 13. ROC of Linear Regression

2. Gausse Naive Bayes

Gaussian Naive Bayes supports continuous valued features and models each as conforming to a Gaussian (normal) distribution. The model created has an accuracy of 0.68 which is similar to the logistic regression model above. The confusion matrix and classification report is shown below in fig. 14.

```
In [30]: gNB = GaussianNB()
gNB.fit(X_train, y_train)
D:\New folder\envs\rstudio-lib\site-packages\sklearn\utils\validation.py:993: DataConversionWarning: A column-vector y was pas
sed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
Out[30]: GaussianNB()

In [31]: y_pred_gNB = gNB.predict(X_test)
print('Accuracy of Guassian Navies Base classifier on test set: {:.2f}'.format(gNB.score(X_test, y_test)))
Accuracy of Guassian Navies Base classifier on test set: 0.68

In [33]: print(classification_report(y_test, y_pred_gNB))
              precision    recall  f1-score   support

           0       0.63      0.53      0.58       816
           1       0.71      0.78      0.74      1169

    accuracy                           0.68      1985
   macro avg       0.67      0.66      0.66      1985
weighted avg       0.67      0.68      0.67      1985
```

Fig. 14. Gausse Naive Bayes summary

The ROC curve for this model is shown below in fig.15.

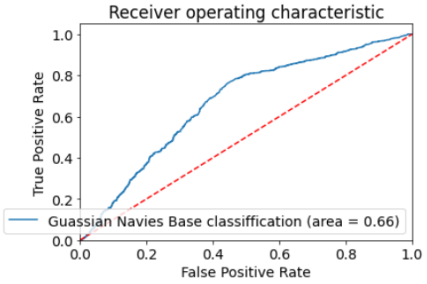
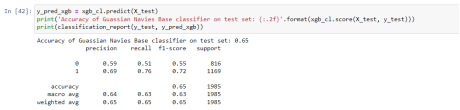


Fig. 15. ROC of Gausse Naive Bayes

3. XGBoost

Gradient boosting, like Random Forest, is an ensembles machine learning algorithm. Trees are introduced to the ensemble one by one and fitted to rectify prior models' prediction mistakes. Boosting is the name for this ensemble machine learning model. The differentiable loss function and gradient descent optimization algorithm are then used to fit the models. The model created has an accuracy of 0.65 which is lesser compared to the two previous models. The confusion matrix and classification report is shown below in fig. 16.



```
In [42]: y_pred_xgb = xgb_cl.predict(X_test)
         print('Accuracy of Guassian Navies Base classifier on test set: {:.2f}'.format(xgb_cl.score(X_test, y_test)))
         print(classification_report(y_test, y_pred_xgb))

Accuracy of Guassian Navies Base classifier on test set: 0.65
              precision    recall  f1-score   support

           0       0.59      0.51      0.55       816
           1       0.69      0.76      0.72      1169

    accuracy                           0.65      1985
   macro avg       0.64      0.63      0.63      1985
weighted avg       0.65      0.65      0.65      1985
```

Fig. 16. XGBoost summary

The ROC curve fopr XGBoost model is shown below in fig.17.

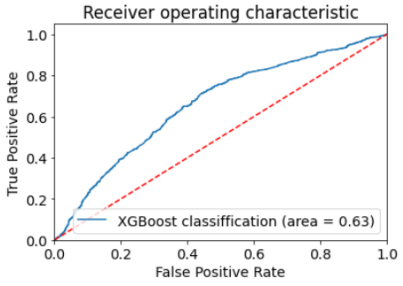


Fig. 17. ROC of XGBoost

C. DATASET 3-SOLAR RADIATION

Business Understanding

Solar energy forecasting represents a key element in increasing the competitiveness of solar power plants in the energy market and reducing the dependence on fossil fuels in economic and social development.As more companies switch to eco-friendly power production, the prediction of solar radiation helps to implement a suitable strategy of instaallation. In the third dataset predictive models of global solar radiation is created using ML algorithms.The research question for this data set is given below:

Can we predict the global solar radiation?

Data Understanding

Once the research question is established we have to explore the data and understand the features present in the data.The third dataset used is "SolarPreddiction.csv". The dependent and independent variables are identified and data quality is checked. The datset is shown in fig.18. The first dataset selected is sourced from the below kaggle link:
https://www.kaggle.com/datasets/dronio/SolarEnergy

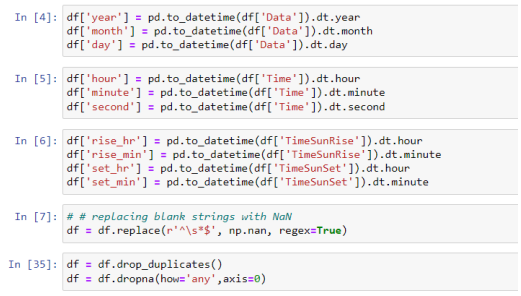
	UNIXTime	Data	Time	Radiation	Temperature	Pressure	Humidity	WindDirection(Degrees)	Speed	TimeSunRise	TimeSunSet
0	1475229326	9/29/2016 12:00:00 AM	23:55:26	1.21	48	30.46	59	177.39	5.62	06:13:00	18:13:00
1	1475229023	9/29/2016 12:00:00 AM	23:50:23	1.21	48	30.46	58	176.78	3.37	06:13:00	18:13:00
2	1475228726	9/29/2016 12:00:00 AM	23:45:26	1.23	48	30.46	57	158.75	3.37	06:13:00	18:13:00
3	1475228421	9/29/2016 12:00:00 AM	23:40:21	1.21	48	30.46	60	137.71	3.37	06:13:00	18:13:00
4	1475228124	9/29/2016 12:00:00 AM	23:35:24	1.17	48	30.46	62	104.95	5.62	06:13:00	18:13:00

Fig. 18. Data Set 3

Data Preperation

In this stage the dataset is loaded into jupyter notebook and data cleaning and preperation is performed.The data cleaning process includes checking null values, outlier treatment, scaling of variables and including dummy variables as shown in fig.19. The heatmap of the dataset is also shown in fig.20. The following steps were performed in this dataset:

- checked for null values and there was none present.
- blank strings were converted to NaN values.
- dropping duplicates records.
- data variable was converted to year, month, day, hour, minute and second.
- TimeSunRise and TimeSunSet variables were converted to hour and minute.
- Scaling of all significant variables are done.



```
In [4]: df['year'] = pd.to_datetime(df['Data']).dt.year
        df['month'] = pd.to_datetime(df['Data']).dt.month
        df['day'] = pd.to_datetime(df['Data']).dt.day

In [5]: df['hour'] = pd.to_datetime(df['Time']).dt.hour
        df['minute'] = pd.to_datetime(df['Time']).dt.minute
        df['second'] = pd.to_datetime(df['Time']).dt.second

In [6]: df['rise_hr'] = pd.to_datetime(df['TimeSunRise']).dt.hour
        df['rise_min'] = pd.to_datetime(df['TimeSunRise']).dt.minute
        df['set_hr'] = pd.to_datetime(df['TimeSunSet']).dt.hour
        df['set_min'] = pd.to_datetime(df['TimeSunSet']).dt.minute

In [7]: # # replacing blank strings with NaN
        df = df.replace(r'^\s*$', np.nan, regex=True)

In [35]: df = df.drop_duplicates()
         df = df.dropna(how='any',axis=0)
```

Fig. 19. Data Set Preprocessing

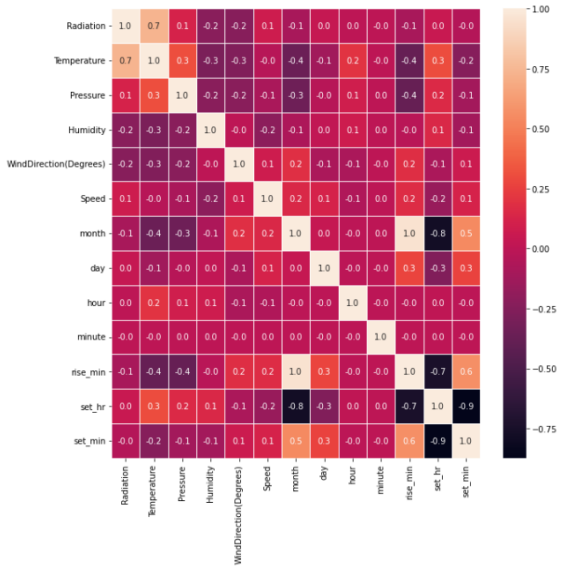


Fig. 20. Heatmap of Dataset 1

Data Modelling

The ML algorithms applied on this dataset are Linear Regression and Random Forest Regressor.

1. Multiple Linear Regression

The cleaned dataset is splitted into training(80%) and testing data(20%). The training data is used to create the linear regression model and the prediction model is tested with testing data. Backward selection is used in this model creation. Initially a model was created with all variables in OLS method. Te next model was created after removing the insignificant variables set_min and speed.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              Radiation   R-squared (uncentered):                   0.711
Model:                            OLS   Adj. R-squared (uncentered):              0.711
Method:                 Least Squares   F-statistic:                              8054.
Date:                Sun, 08 May 2022   Prob (F-statistic):                        0.00
Time:                        20:15:18   Log-Likelihood:                          16990.
No. Observations:               26148   AIC:                                 -3.396e+04
Df Residuals:                   26140   BIC:                                 -3.390e+04
Df Model:                           8
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Temperature     0.8889      0.005    183.559      0.000       0.879       0.898
Pressure       -0.2764      0.005    -60.751      0.000      -0.285      -0.268
Humidity       -0.0651      0.003    -25.685      0.000      -0.070      -0.060
WindDirection(Degrees)    -0.1158      0.003    -35.403      0.000      -0.122      -0.109
month           0.0479      0.002     22.648      0.000       0.044       0.052
day             0.0525      0.003     20.005      0.000       0.047       0.058
hour           -0.1059      0.003    -39.120      0.000      -0.111      -0.101
minute         -0.0201      0.003     -7.916      0.000      -0.025      -0.015
==============================================================================
Omnibus:                     3102.522   Durbin-Watson:                   1.998
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             5271.807
Skew:                           0.816   Prob(JB):                         0.00
Kurtosis:                       4.474   Cond. No.                         11.8
==============================================================================
```

Fig. 21. Linear Regression summary

The above model in fig.21 has an adjusted R_2 value of 0.711.The model created has an accuracy of 71.1%.The same final data was used to create a Linear Regression model which has an RMSE value of 0.1213 as shown in fig.22.

```
In [29]: y_predictions = lin_reg.predict(X_test)
         lin_mse = mean_squared_error(y_test, y_predictions)
         lin_rmse = np.sqrt(lin_mse)
         print("Mean Squared Error = ",lin_mse)
         print("Root Mean Squared Error = ",lin_rmse)

Mean Squared Error =  0.014714517643986475
Root Mean Squared Error =  0.12130341151009098
```

Fig. 22. RMSE, MSE values of Linear Regression

The predicted vs actual values are plotted below in fig.23. for the model:

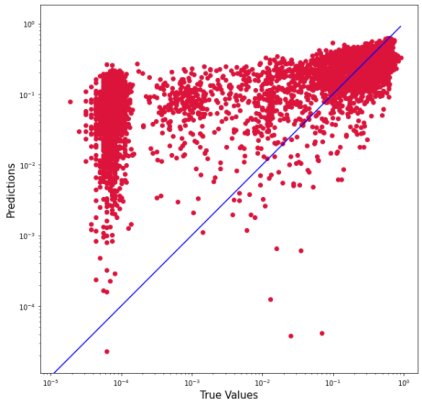


Fig. 23. Predicted vs Actual

2. Random Forest Regressor

Random Forest Regression is a supervised learning technique that does regression using the ensemble learning method. This method accurately predicts by combining predictions from different machine learning algorithms. The same final data is used to create the random forest model. The model had an RMSE value of 0.05 as shown in fig.24. Comparing with the previous model Random Forest model has a better accuracy with respect to RMSE values.

```
In [33]: y_predictions = forest_reg.predict(X_test)
         forest_mse = mean_squared_error(y_test, y_predictions)
         forest_rmse = np.sqrt(forest_mse)
         print("Mean Squared Error = ",forest_mse)
         print("Root Mean Squared Error = ",forest_rmse)

Mean Squared Error =  0.0025056653951757773
Root Mean Squared Error =  0.05005662189137195
```

Fig. 24. RMSE, MSE values of Random Forest Regressor

The predicted vs actual values are plotted below in fig.25 for Decision Tree model:

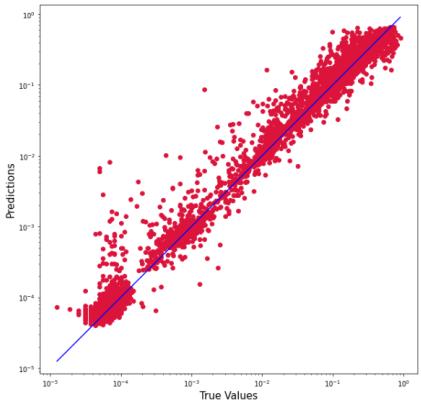


Fig. 25. Predicted vs Actual

V. CONCLUSION AND FUTURE WORK

We have used three datasets in this projects to approach the research questions mentioned in Introduction. The Ml algorithms used for the dataset london_merged.csv are Linear Regression and Decision Tree.The Decision Tree model was found to be a better fit with RMSE values of 0.104. In the NFL dataset we used Logistic Regression, Gauss Naive Bayes and XGBoost. The LR and GNb model performed best with accuracy of 68%. For the final dataset SolarRadiation.csv we have created models using Linear Regression and Random Forest Regressor. The RF regressor was more suitable for the dataset because of a lower RMSE of 0.05. Evaluation for the models were done using methods such as adjusted R_2, ROC curve, confusion matrix and plots for predicted vs actual. This project provides insight into a few ML algorithms, paving way for more research of algorithms, parameter tuning. More models can be created using othe ML algorithms to find a better fitting model.

REFERENCES

[1] Sathishkumar V E, Jangwoo Park, Yongyun Cho, "Using data mining techniques for bike sharing demand prediction in metropolitan city", Computer Communications, Volume 153,2020,Pages 353-366,ISSN 0140-3664,2020.https://www.sciencedirect.com/science/article/pii/S0140366419318997

[2] Huthaifa I. Ashqar, Mohammed Elhenawy, Hesham A. Rakha, Mohammed Almannaa, Leanna House, Network and station-level bike-sharing system prediction: a San Francisco bay area case study, Journal of Intelligent Transportation Systems, 2022, https://www.sciencedirect.com/science/article/pii/S1547245022003796

[3] Neofytos Boufidis, Andreas Nikiforiadis, Katerina Chrysostomou, Georgia Aifadopoulou,Development of a station-level demand prediction and visualization tool to support bike-sharing systems' operators,Transportation Research Procedia,Volume 47,2020,Pages 51-58,ISSN 2352-1465, https://www.sciencedirect.com/science/article/pii/S2352146520302593

[4] El Arbi Abdellaoui Alaoui, Stephane Cedric Koumetio Tekouabou, Intelligent management of bike sharing in smart cities using machine learning and Internet of Things, Sustainable Cities and Society, Volume 67, 2021, 102702, ISSN 2210-6707, https://www.sciencedirect.com/science/article/pii/S2210670720309161

[5] Joseph Chibwe, Shahram Heydari, Ahmadreza Faghih Imani, Aneta Scurtu, An exploratory analysis of the trend in the demand for the London bike-sharing system: From London Olympics to Covid-19 pandemic, Sustainable Cities and Society, Volume 69, 2021, 102871, ISSN 2210-6707, https://www.sciencedirect.com/science/article/pii/S221067072100161X

[6] Mohammed H. Almannaa, Mohammed Elhenawy, Hesham A. Rakha, Dynamic linear models to predict bike availability in a bike sharing system, International Journal of Sustainable Transportation, Volume 14, Issue 3, 2020, Pages 232-242, ISSN 1556-8318, https://www.sciencedirect.com/org/science/article/abs/pii/S155683182200288X

[7] Neeraj Pathak, Hardik Wadhwa, Applications of Modern Classification Techniques to Predict the Outcome of ODI Cricket, Procedia Computer Science, Volume 87, 2016, Pages 55-60, ISSN 1877-0509, https://www.sciencedirect.com/science/article/pii/S1877050916304653

[8] A. Joseph, N.E. Fenton, M. Neil, Predicting football results using Bayesian nets and other machine learning techniques, Knowledge-Based Systems, Volume 19, Issue 7, 2006, Pages 544-553, ISSN 0950-7051, https://www.sciencedirect.com/science/article/pii/S0950705106000724

[9] Wei Gu, Krista Foster, Jennifer Shang, Lirong Wei, A game-predicting expert system using big data and machine learning, Expert Systems with Applications, Volume 130, 2019, Pages 293-305, ISSN 0957-4174, https://www.sciencedirect.com/science/article/pii/S0957417419302556

[10] Rory P. Bunker, Fadi Thabtah, A machine learning framework for sport result prediction, Applied Computing and Informatics, Volume 15, Issue 1, 2019, Pages 27-33, ISSN 2210-8327, https://www.sciencedirect.com/science/article/pii/S2210832717301485

[11] Igor Barbosa da Costa, Leandro Balby Marinho, Carlos Eduardo Santos Pires, Forecasting football results and exploiting betting markets: The case of "both teams to score", International Journal of Forecasting, 2021, ISSN 0169-2070, https://www.sciencedirect.com/science/article/pii/S0169207021001084

[12] Y. El Mghouchi, On the prediction of daily global solar radiation using temperature as input. An application of hybrid machine learners to the six climatic Moroccan zones, Energy Conversion and Management: X, Volume 13, 2022, 100157, ISSN 2590-1745, https://www.sciencedirect.com/science/article/pii/S2590174521000829

[13] Neha Singh, Satyaranjan Jena, Chinmoy Kumar Panigrahi, A novel application of Decision Tree classifier in solar irradiance prediction, Materials Today: Proceedings, 2022,ISSN 2214-7853, https://www.sciencedirect.com/science/article/pii/S2214785322008124

[14] Dongyu Jia, Liwei Yang, Tao Lv, Weiping Liu, Xiaoqing Gao, Jiaxin Zhou, Evaluation of machine learning models for predicting daily global and diffuse solar radiation under different weather/pollution conditions, Renewable Energy, Volume 187, 2022, Pages 896-906, ISSN 0960-1481, https://www.sciencedirect.com/science/article/pii/S0960148122001410

[15] Ümit Ağbulut, Ali Etem Gürel, Yunus Biçen, Prediction of daily global solar radiation using different machine learning algorithms: Evaluation and comparison, Renewable and Sustainable Energy Reviews, Volume 135, 2021, 110114, ISSN 1364-0321, https://www.sciencedirect.com/science/article/pii/S1364032120304056

[16] Juan Antonio Bellido-Jiménez, Javier Estévez Gualda, Amanda Penélope García-Marín, Assessing new intra-daily temperature-based machine learning models to outperform solar radiation predictions in different conditions, Applied Energy, Volume 298, 2021, 117211, ISSN 0306-2619, https://www.sciencedirect.com/science/article/pii/S0306261921006358