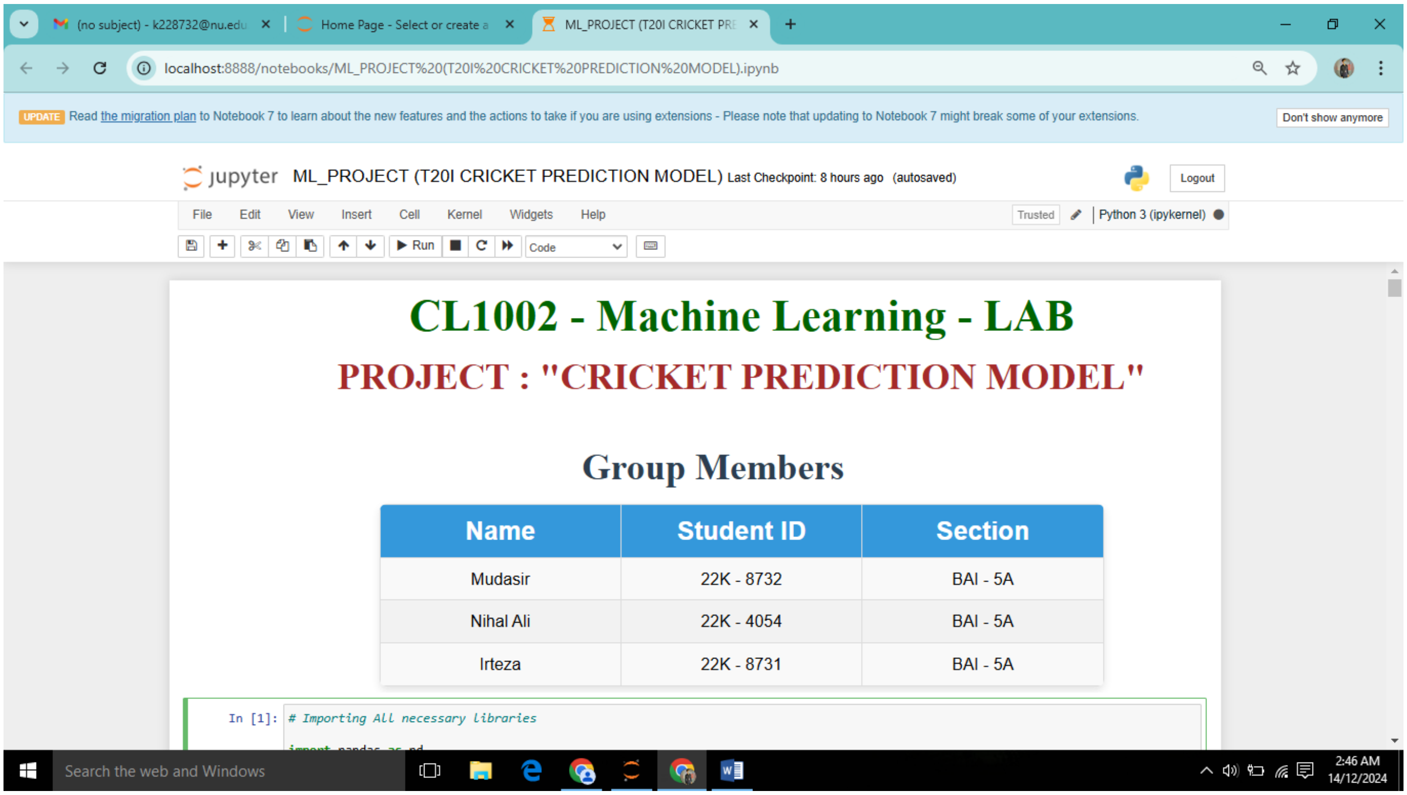
National University
of computer and emerging sciences

National University of Computer & Emerging Sciences

Karachi Campus


(no subject) - k228732@nu.edu
Home Page - Select or create a
ML_PROJECT (T20I CRICKET PRE
localhost:8888/notebooks/ML_PROJECT%20(T20I%20CRICKET%20PREDICTION%20MODEL).ipynb
UPDATE
Don't show anymore
jupyter
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3 (ipykernel)
Run
Code
Search the web and Windows
2:46 AM
14/12/2024

Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.

ML_PROJECT (T20I CRICKET PREDICTION MODEL) Last Checkpoint: 8 hours ago (autosaved)

CL1002 - Machine Learning - LAB

PROJECT : "CRICKET PREDICTION MODEL"

Group Members

Name	Student ID	Section
Mudasir	22K - 8732	BAI - 5A
Nihal Ali	22K - 4054	BAI - 5A
Irteza	22K - 8731	BAI - 5A

```
In [1]: # Importing All necessary libraries
        import pandas as pd
```

Class/Section: BAI-5A

Date: 14/Dec/2024

Instructor: Sir Usama Bin Umar

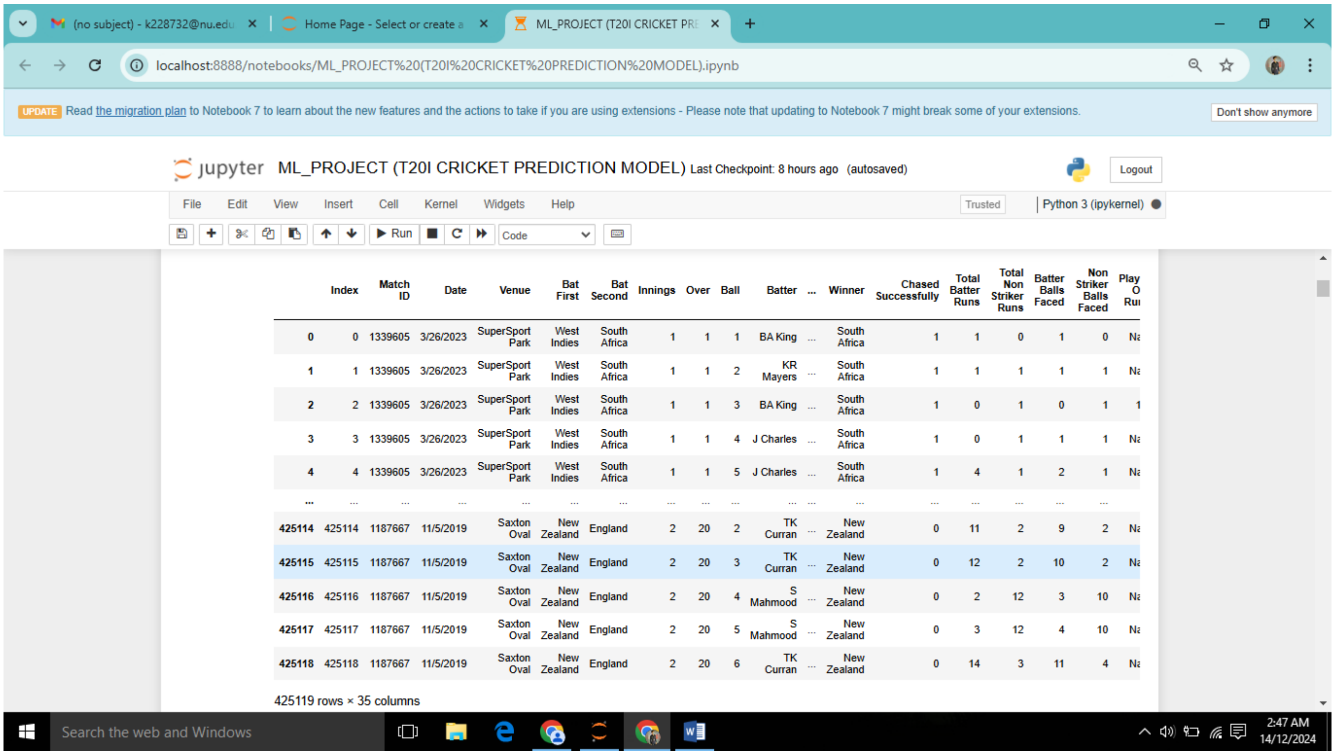
T20I CRICKET PREDICTION MODEL

This project aims to predict the winner of T20 International cricket matches based on historical data and statistical analysis. The model uses a Random Forest Classifier, which outperformed other algorithms, achieving an accuracy of 0.95 or 95%. Various machine learning models were evaluated, and an interactive GUI was developed to enhance user engagement.

DATASET OVERVIEW

The dataset is structured on a ball-by-ball basis, later aggregated by match date to eliminate duplicate rows. Key attributes include:

- Match ID, Date, Venue
- Batting and bowling statistics (runs, wickets, overs, etc.)
- Match outcomes like Winner and Toss Decision (Bat First/Second).


(no subject) - k228732@nu.edu
Home Page - Select or create a
ML_PROJECT (T20I CRICKET PR
localhost:8888/notebooks/ML_PROJECT%20(T20I%20CRICKET%20PREDICTION%20MODEL).ipynb
UPDATE
Don't show anymore
jupyter
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3 (ipykernel)
Run
Code
Search the web and Windows
2:47 AM
14/12/2024

Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.

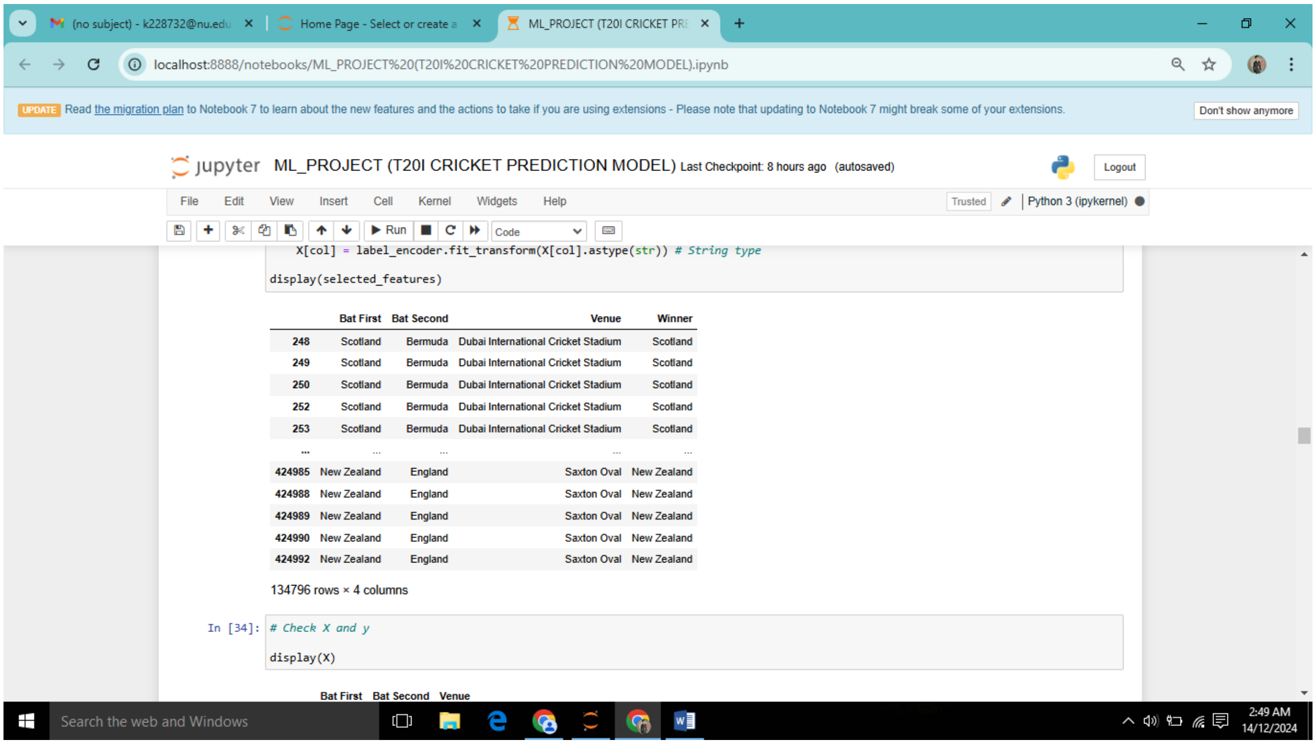
ML_PROJECT (T20I CRICKET PREDICTION MODEL) Last Checkpoint: 8 hours ago (autosaved)

	Index	Match ID	Date	Venue	Bat First	Bat Second	Innings	Over	Ball	Batter	...	Winner	Chased Successfully	Total Batter Runs	Total Non Striker Runs	Batter Balls Faced	Non Striker Balls Faced	Play O Ru
0	0	1339605	3/26/2023	SuperSport Park	West Indies	South Africa	1	1	1	BA King	...	South Africa	1	1	0	1	0	Na
1	1	1339605	3/26/2023	SuperSport Park	West Indies	South Africa	1	1	2	KR Mayers	...	South Africa	1	1	1	1	1	Na
2	2	1339605	3/26/2023	SuperSport Park	West Indies	South Africa	1	1	3	BA King	...	South Africa	1	0	1	0	1	1
3	3	1339605	3/26/2023	SuperSport Park	West Indies	South Africa	1	1	4	J Charles	...	South Africa	1	0	1	1	1	Na
4	4	1339605	3/26/2023	SuperSport Park	West Indies	South Africa	1	1	5	J Charles	...	South Africa	1	4	1	2	1	Na
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
425114	425114	1187667	11/5/2019	Saxton Oval	New Zealand	England	2	20	2	TK Curran	...	New Zealand	0	11	2	9	2	Na
425115	425115	1187667	11/5/2019	Saxton Oval	New Zealand	England	2	20	3	TK Curran	...	New Zealand	0	12	2	10	2	Na
425116	425116	1187667	11/5/2019	Saxton Oval	New Zealand	England	2	20	4	S Mahmood	...	New Zealand	0	2	12	3	10	Na
425117	425117	1187667	11/5/2019	Saxton Oval	New Zealand	England	2	20	5	S Mahmood	...	New Zealand	0	3	12	4	10	Na
425118	425118	1187667	11/5/2019	Saxton Oval	New Zealand	England	2	20	6	TK Curran	...	New Zealand	0	14	3	11	4	Na

425119 rows × 35 columns

Selected Features for the Model:

- Bat First (toss winner)
- Bat Second
- Venue
- Winner (label/target column)


(no subject) - k228732@nu.edu
Home Page - Select or create a
ML_PROJECT (T20I CRICKET PRE
localhost:8888/notebooks/ML_PROJECT%20(T20I%20CRICKET%20PREDICTION%20MODEL).ipynb
Don't show anymore
jupyter
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3 (ipykernel)
Run
Code
Search the web and Windows
2:49 AM
14/12/2024

UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.

ML_PROJECT (T20I CRICKET PREDICTION MODEL) Last Checkpoint: 8 hours ago (autosaved)

```
    X[col] = label_encoder.fit_transform(X[col].astype(str)) # String type

display(selected_features)
```

	Bat First	Bat Second	Venue	Winner
248	Scotland	Bermuda	Dubai International Cricket Stadium	Scotland
249	Scotland	Bermuda	Dubai International Cricket Stadium	Scotland
250	Scotland	Bermuda	Dubai International Cricket Stadium	Scotland
252	Scotland	Bermuda	Dubai International Cricket Stadium	Scotland
253	Scotland	Bermuda	Dubai International Cricket Stadium	Scotland
...	...	...	...	...
424985	New Zealand	England	Saxton Oval	New Zealand
424988	New Zealand	England	Saxton Oval	New Zealand
424989	New Zealand	England	Saxton Oval	New Zealand
424990	New Zealand	England	Saxton Oval	New Zealand
424992	New Zealand	England	Saxton Oval	New Zealand

134796 rows × 4 columns

```
In [34]: # Check X and y

         display(X)
```

Bat First Bat Second Venue

The dataset considers the toss-winning team as the team batting first.

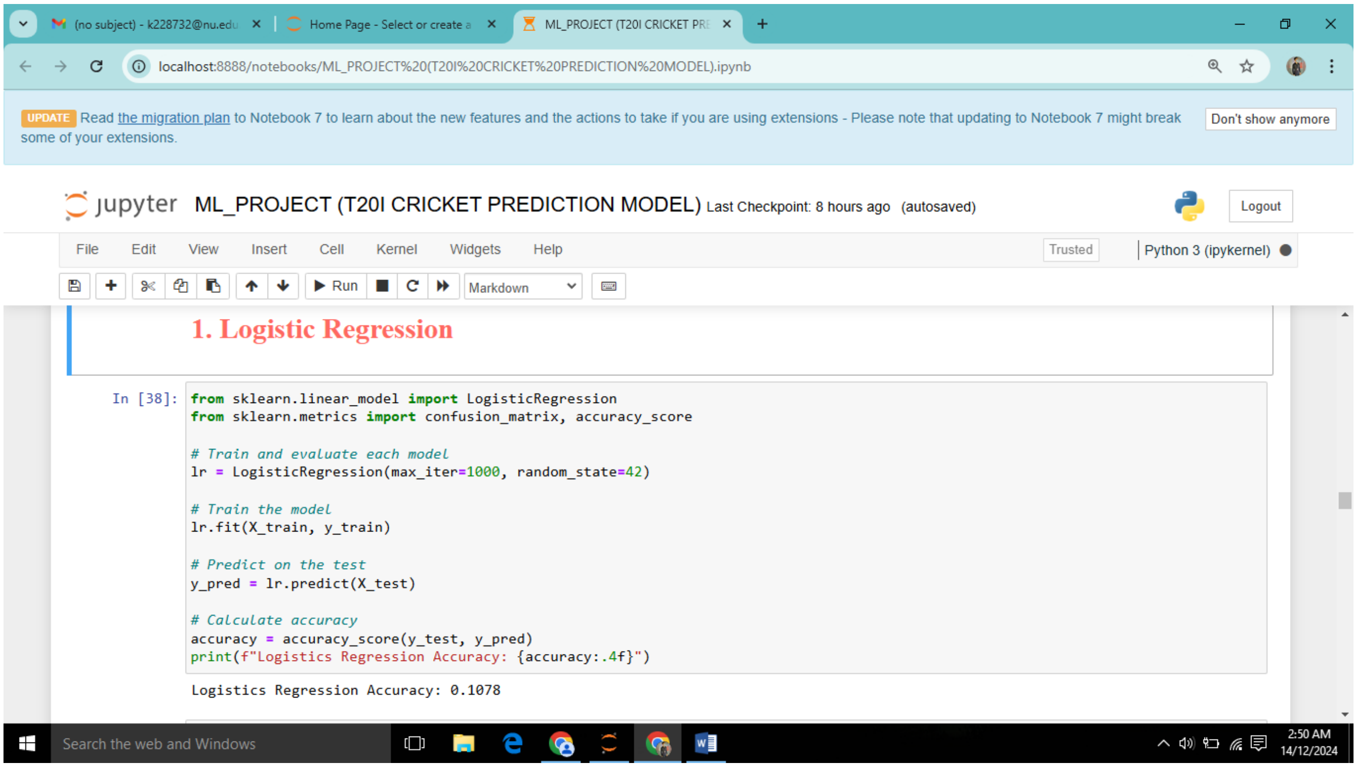
Machine Learning Models Evaluated

1. Logistic Regression

- ➢ Testing Accuracy: 0.1078 or 10.78%

- ➢ Reason for Low Accuracy

Logistic Regression is primarily suited for binary classification. Since the target variable includes multiple teams as possible winners, the algorithm fails to generalize.


(no subject) - k228732@nu.edu
Home Page - Select or create a
ML_PROJECT (T20I CRICKET PR
localhost:8888/notebooks/ML_PROJECT%20(T20I%20CRICKET%20PREDICTION%20MODEL).ipynb
UPDATE
Don't show anymore
jupyter ML_PROJECT (T20I CRICKET PREDICTION MODEL) Last Checkpoint: 8 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Markdown
Search the web and Windows
2:50 AM
14/12/2024

Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.

1. Logistic Regression

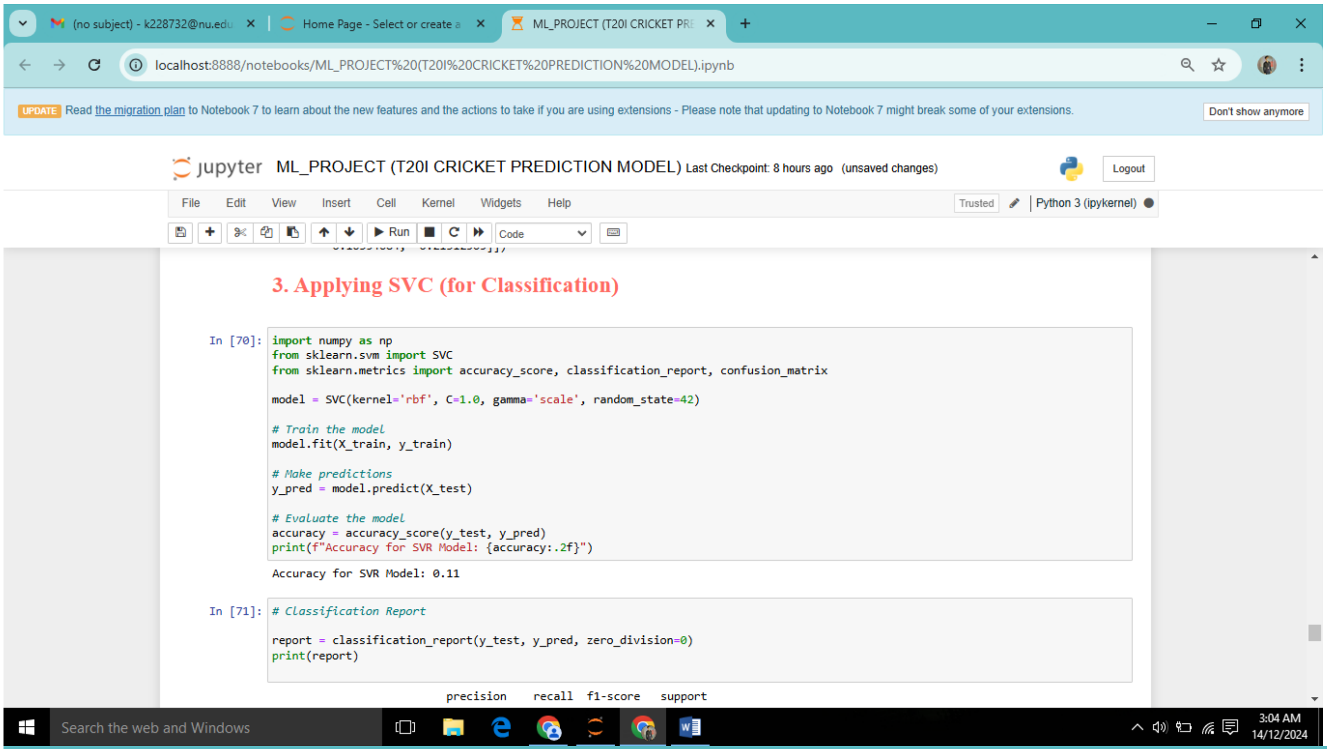
```
In [38]: from sklearn.linear_model import LogisticRegression
         from sklearn.metrics import confusion_matrix, accuracy_score

         # Train and evaluate each model
         lr = LogisticRegression(max_iter=1000, random_state=42)

         # Train the model
         lr.fit(X_train, y_train)

         # Predict on the test
         y_pred = lr.predict(X_test)

         # Calculate accuracy
         accuracy = accuracy_score(y_test, y_pred)
         print(f"Logistics Regression Accuracy: {accuracy:.4f}")
```

```
Logistics Regression Accuracy: 0.1078
```

2. Support Vector Classifier (SVC)

- Testing Accuracy: 0.11 or 11%
- Reason for Low Performance

  SVC struggles with multi-class classification when the dataset is large and complex. The need for significant parameter tuning also impacted its performance.



(no subject) - k228732@nu.edu
Home Page - Select or create a
ML_PROJECT (T20I CRICKET PR
localhost:8888/notebooks/ML_PROJECT%20(T20I%20CRICKET%20PREDICTION%20MODEL).ipynb
UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.
Don't show anymore
jupyter ML_PROJECT (T20I CRICKET PREDICTION MODEL) Last Checkpoint: 8 hours ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
Search the web and Windows
3:04 AM
14/12/2024

3. Applying SVC (for Classification)

In [70]:

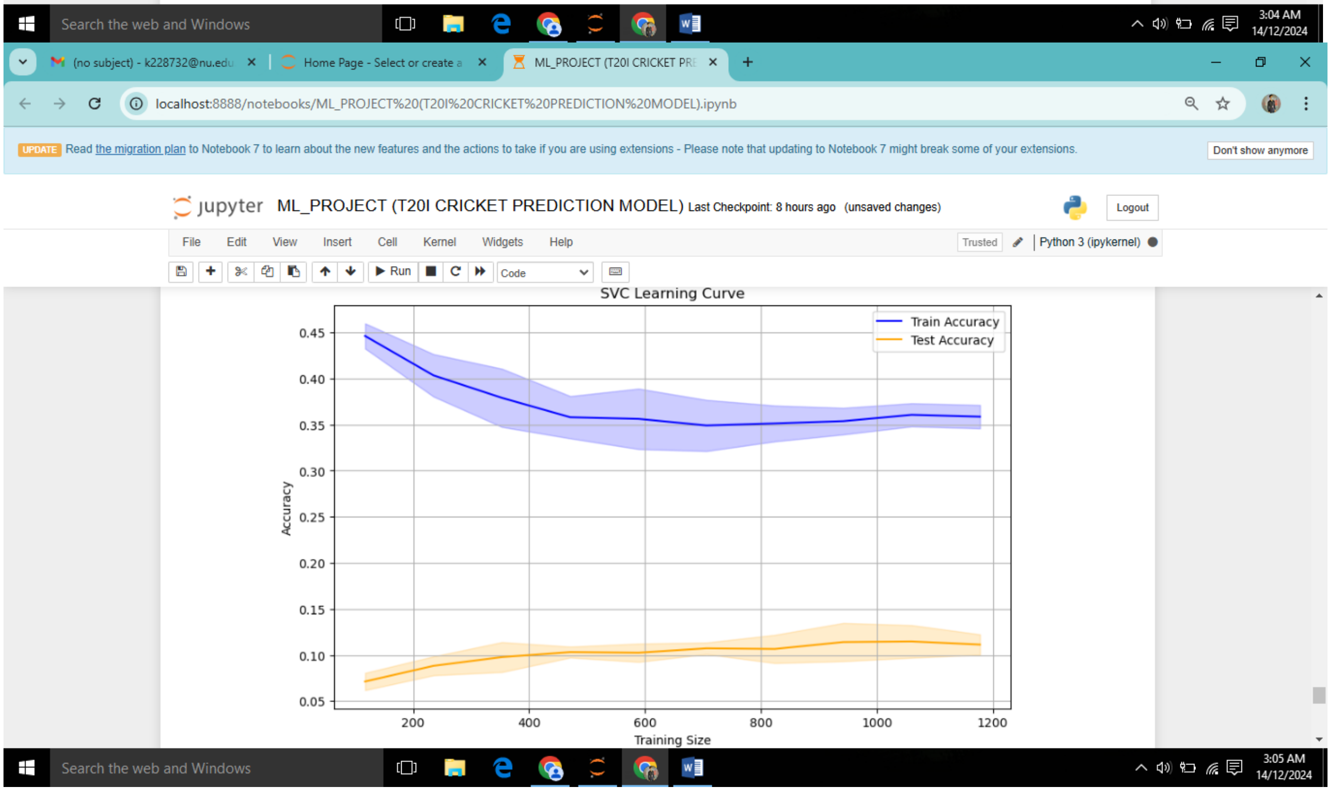
```
import numpy as np
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

model = SVC(kernel='rbf', C=1.0, gamma='scale', random_state=42)

# Train the model
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy for SVR Model: {accuracy:.2f}")
```

```
Accuracy for SVR Model: 0.11
```

In [71]:

```
# Classification Report

report = classification_report(y_test, y_pred, zero_division=0)
print(report)
```

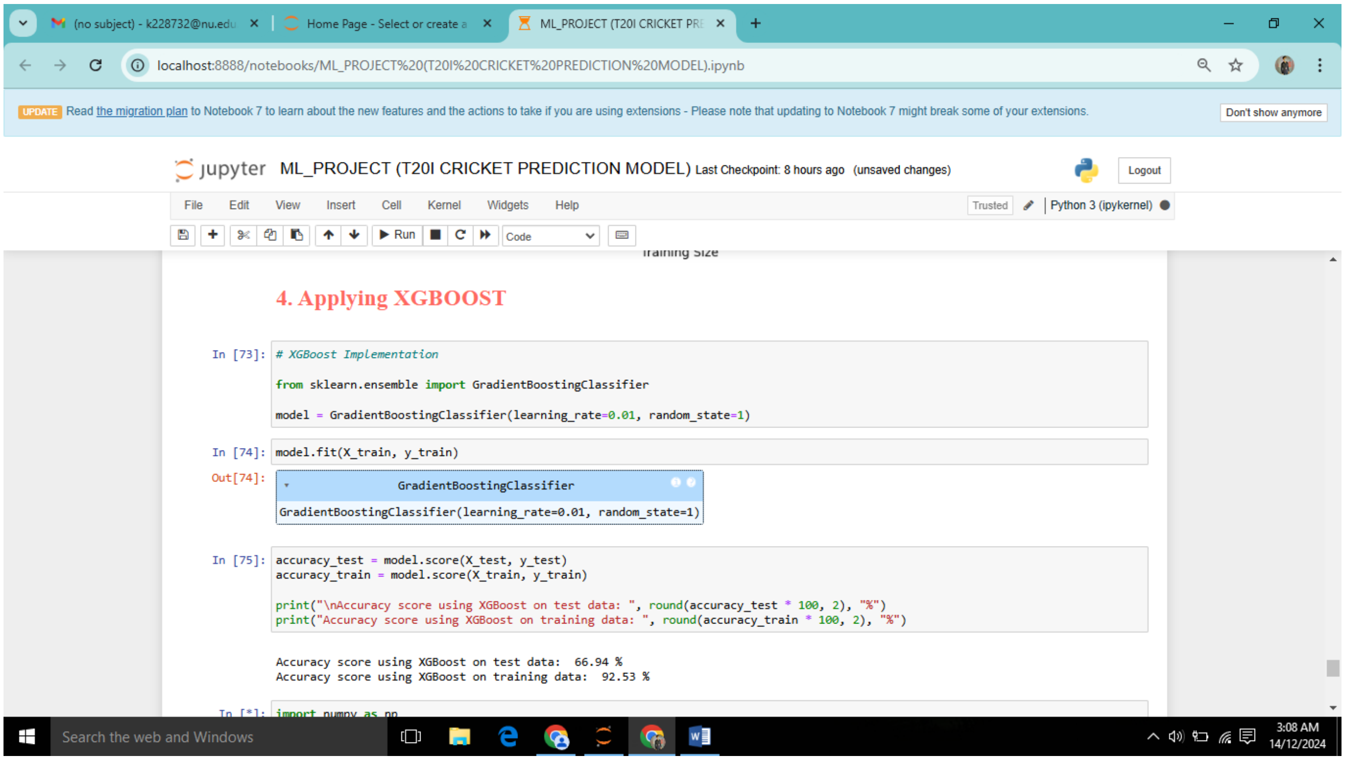
```
              precision    recall  f1-score   support
```



3. XGBoost Classifier

- Testing Accuracy: 0.6694 or 66.94%
- Reason for Partial Success:

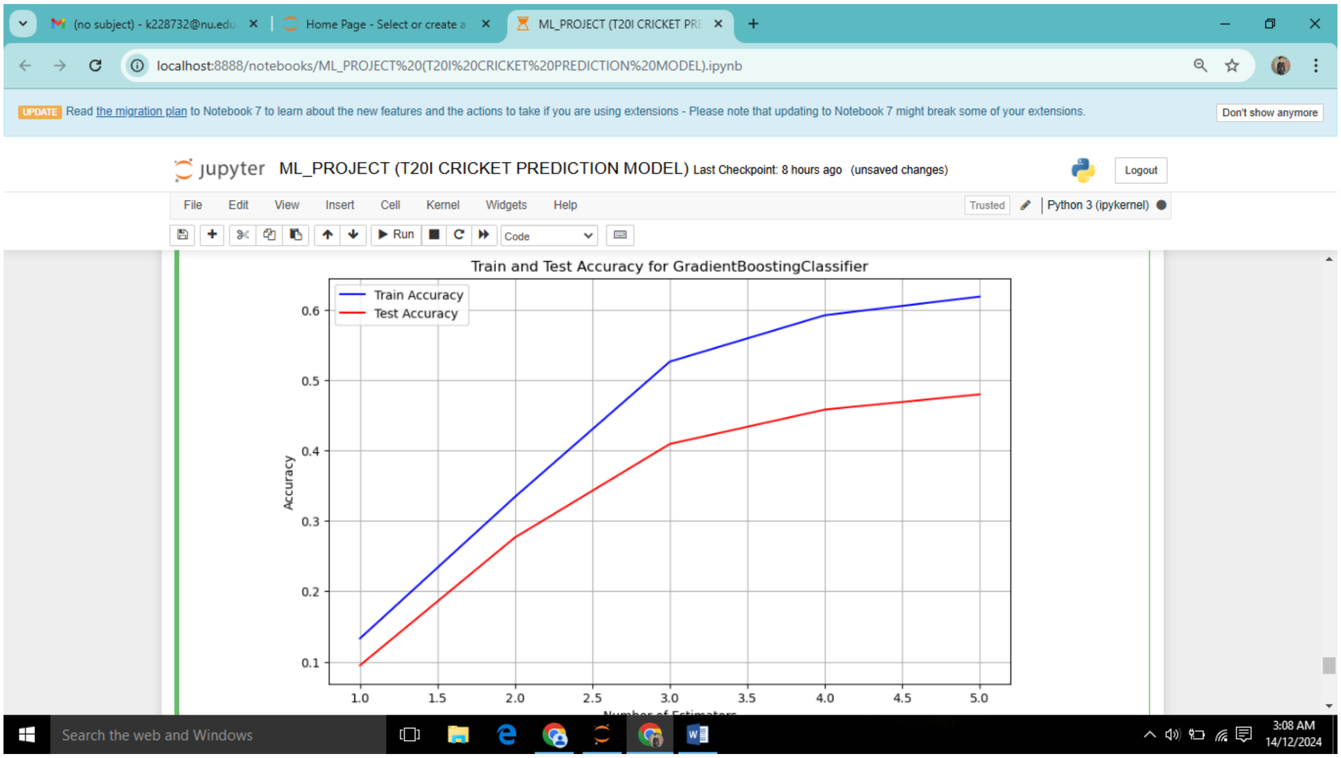
  XGBoost handles classification tasks better but requires fine-tuned hyperparameters and specific handling of categorical features for optimal performance.


(no subject) - k228732@nu.edu
Home Page - Select or create a
ML_PROJECT (T20I CRICKET PRE
localhost:8888/notebooks/ML_PROJECT%20(T20I%20CRICKET%20PREDICTION%20MODEL).ipynb
UPDATE
Don't show anymore
jupyter
ML_PROJECT (T20I CRICKET PREDICTION MODEL) Last Checkpoint: 8 hours ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3 (ipykernel)
Run
Code
Training Size
Search the web and Windows
3:08 AM
14/12/2024

Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.

4. Applying XGBOOST

```
In [73]: # XGBoost Implementation

         from sklearn.ensemble import GradientBoostingClassifier

         model = GradientBoostingClassifier(learning_rate=0.01, random_state=1)

In [74]: model.fit(X_train, y_train)

Out[74]:          GradientBoostingClassifier
         GradientBoostingClassifier(learning_rate=0.01, random_state=1)

In [75]: accuracy_test = model.score(X_test, y_test)
         accuracy_train = model.score(X_train, y_train)

         print("\nAccuracy score using XGBoost on test data: ", round(accuracy_test * 100, 2), "%")
         print("Accuracy score using XGBoost on training data: ", round(accuracy_train * 100, 2), "%")

         Accuracy score using XGBoost on test data:  66.94 %
         Accuracy score using XGBoost on training data:  92.53 %

In [*]: import numpy as np
```

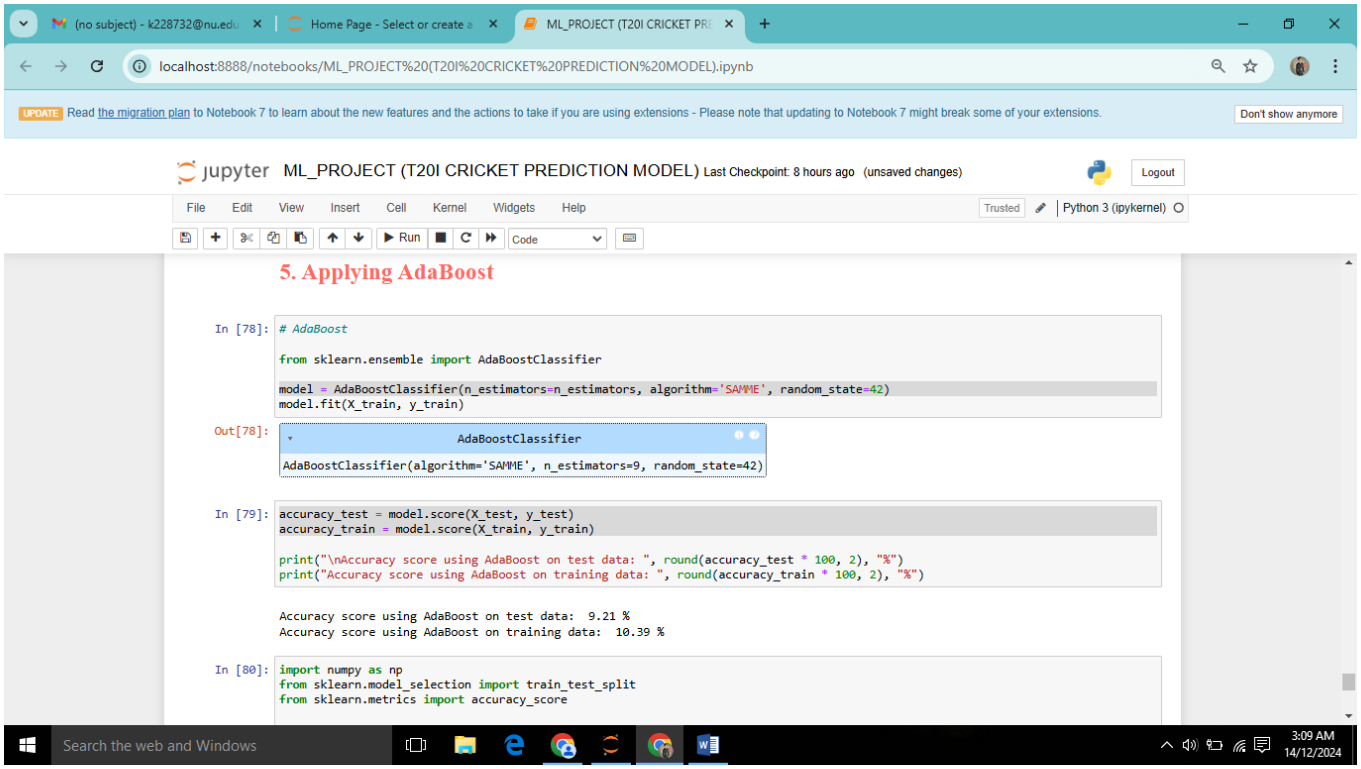

(no subject) - k228732@nu.edu
Home Page - Select or create a
ML_PROJECT (T20I CRICKET PRE
localhost:8888/notebooks/ML_PROJECT%20(T20I%20CRICKET%20PREDICTION%20MODEL).ipynb
UPDATE
Don't show anymore
jupyter
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3 (ipykernel)
Run
Code
Train and Test Accuracy for GradientBoostingClassifier
Train Accuracy
Test Accuracy
Accuracy
0.1
0.2
0.3
0.4
0.5
0.6
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Number of Estimators
Search the web and Windows
3:08 AM
14/12/2024

Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.

ML_PROJECT (T20I CRICKET PREDICTION MODEL) Last Checkpoint: 8 hours ago (unsaved changes)

4. ADABOOST CLASSIFIER

- Testing Accuracy: 0.0921 or 9.21%
- Reason for Lower Performance:
  AdaBoost relies heavily on weak learners and performs well for simpler datasets. The complexity and scale of the dataset limited its effectiveness.


(no subject) - k228732@nu.edu
Home Page - Select or create a
ML_PROJECT (T20I CRICKET PRE
localhost:8888/notebooks/ML_PROJECT%20(T20I%20CRICKET%20PREDICTION%20MODEL).ipynb
UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.
Don't show anymore
jupyter ML_PROJECT (T20I CRICKET PREDICTION MODEL) Last Checkpoint: 8 hours ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
Search the web and Windows
3:09 AM
14/12/2024

5. Applying AdaBoost

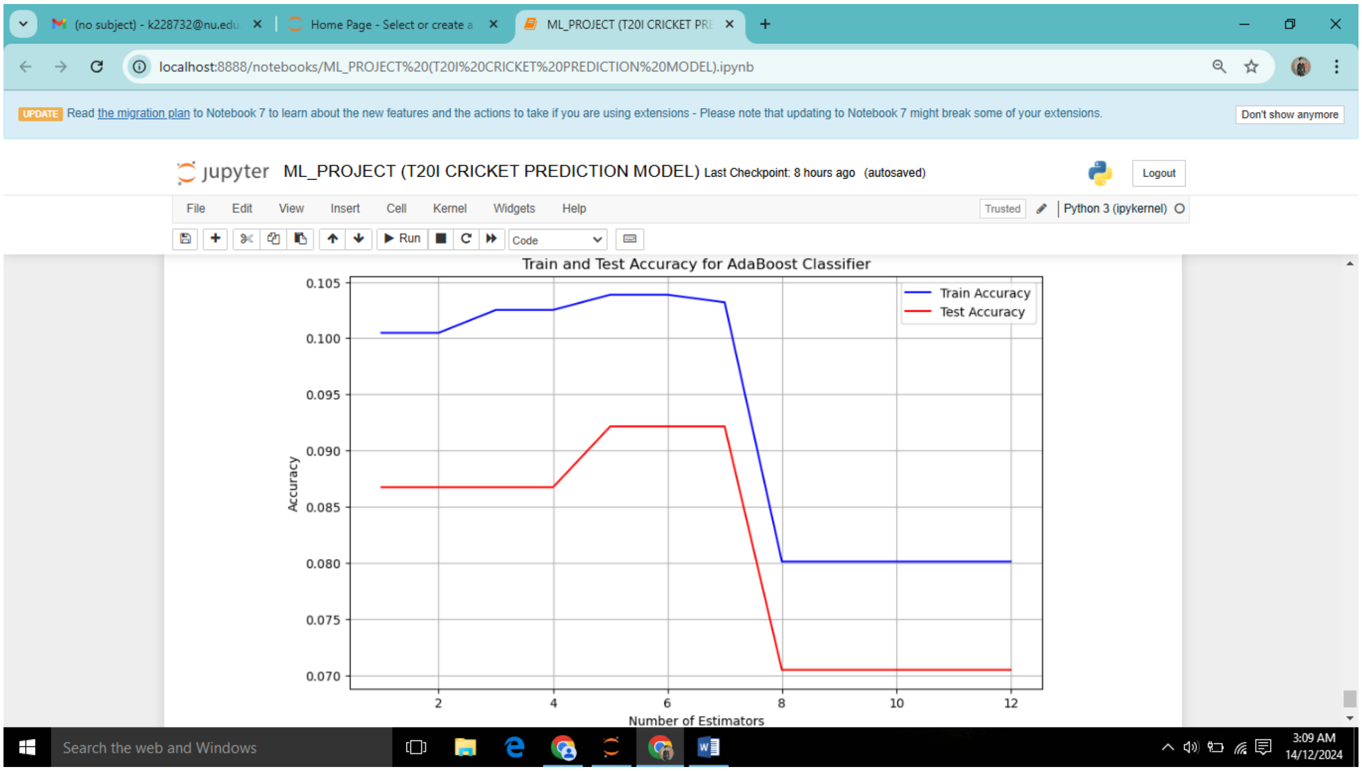
In [78]:
```
# AdaBoost

from sklearn.ensemble import AdaBoostClassifier

model = AdaBoostClassifier(n_estimators=n_estimators, algorithm='SAMME', random_state=42)
model.fit(X_train, y_train)
```

Out[78]:
```
AdaBoostClassifier
AdaBoostClassifier(algorithm='SAMME', n_estimators=9, random_state=42)
```

In [79]:
```
accuracy_test = model.score(X_test, y_test)
accuracy_train = model.score(X_train, y_train)

print("\nAccuracy score using AdaBoost on test data: ", round(accuracy_test * 100, 2), "%")
print("Accuracy score using AdaBoost on training data: ", round(accuracy_train * 100, 2), "%")
```

```
Accuracy score using AdaBoost on test data:  9.21 %
Accuracy score using AdaBoost on training data:  10.39 %
```

In [80]:
```
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

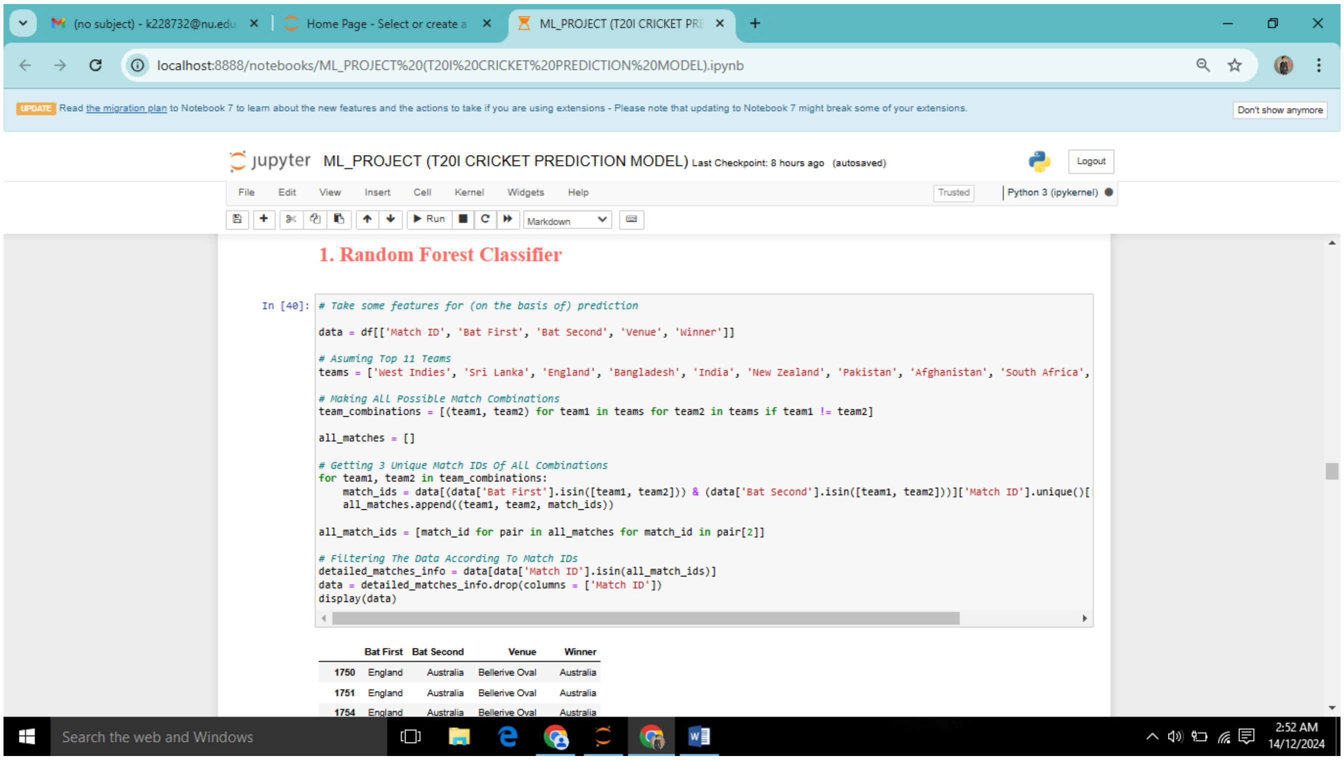


5. Random Forest Classifier

- Testing Accuracy: 0.95 or 95%

➢ Why It Succeeded

Random Forest effectively handles categorical variables, captures non-linear relationships, and prevents overfitting with its ensemble approach.



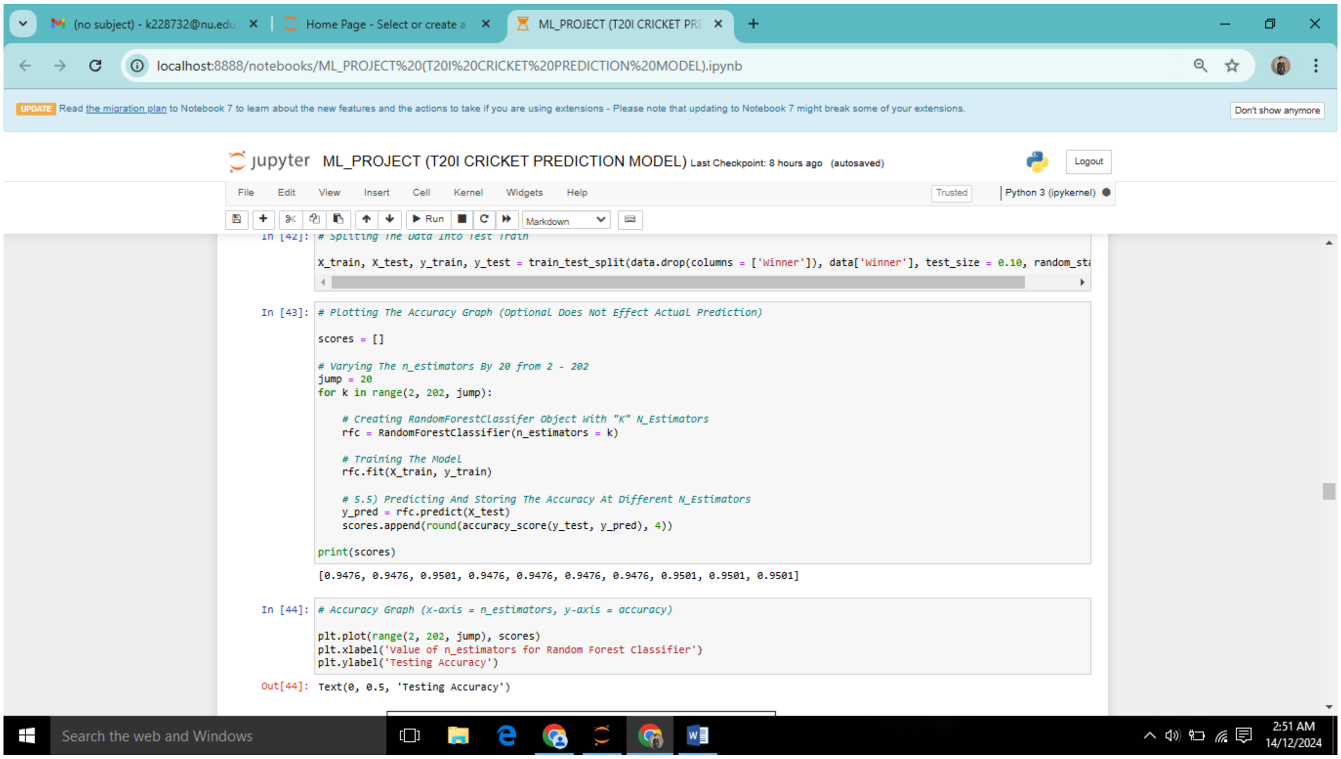
localhost:8888/notebooks/ML_PROJECT%20(T20I%20CRICKET%20PREDICTION%20MODEL).ipynb

jupyter ML_PROJECT (T20I CRICKET PREDICTION MODEL) Last Checkpoint: 8 hours ago (autosaved)

1. Random Forest Classifier

```
In [40]: # Take some features for (on the basis of) prediction

data = df[['Match ID', 'Bat First', 'Bat Second', 'Venue', 'Winner']]

# Asuming Top 11 Teams
teams = ['West Indies', 'Sri Lanka', 'England', 'Bangladesh', 'India', 'New Zealand', 'Pakistan', 'Afghanistan', 'South Africa',

# Making All Possible Match Combinations
team_combinations = [(team1, team2) for team1 in teams for team2 in teams if team1 != team2]

all_matches = []

# Getting 3 Unique Match IDs Of All Combinations
for team1, team2 in team_combinations:
    match_ids = data[(data['Bat First'].isin([team1, team2])) & (data['Bat Second'].isin([team1, team2]))]['Match ID'].unique()[
    all_matches.append((team1, team2, match_ids))

all_match_ids = [match_id for pair in all_matches for match_id in pair[2]]

# Filtering The Data According To Match IDs
detailed_matches_info = data[data['Match ID'].isin(all_match_ids)]
data = detailed_matches_info.drop(columns = ['Match ID'])
display(data)
```

	Bat First	Bat Second	Venue	Winner
1750	England	Australia	Bellerive Oval	Australia
1751	England	Australia	Bellerive Oval	Australia
1754	England	Australia	Bellerive Oval	Australia



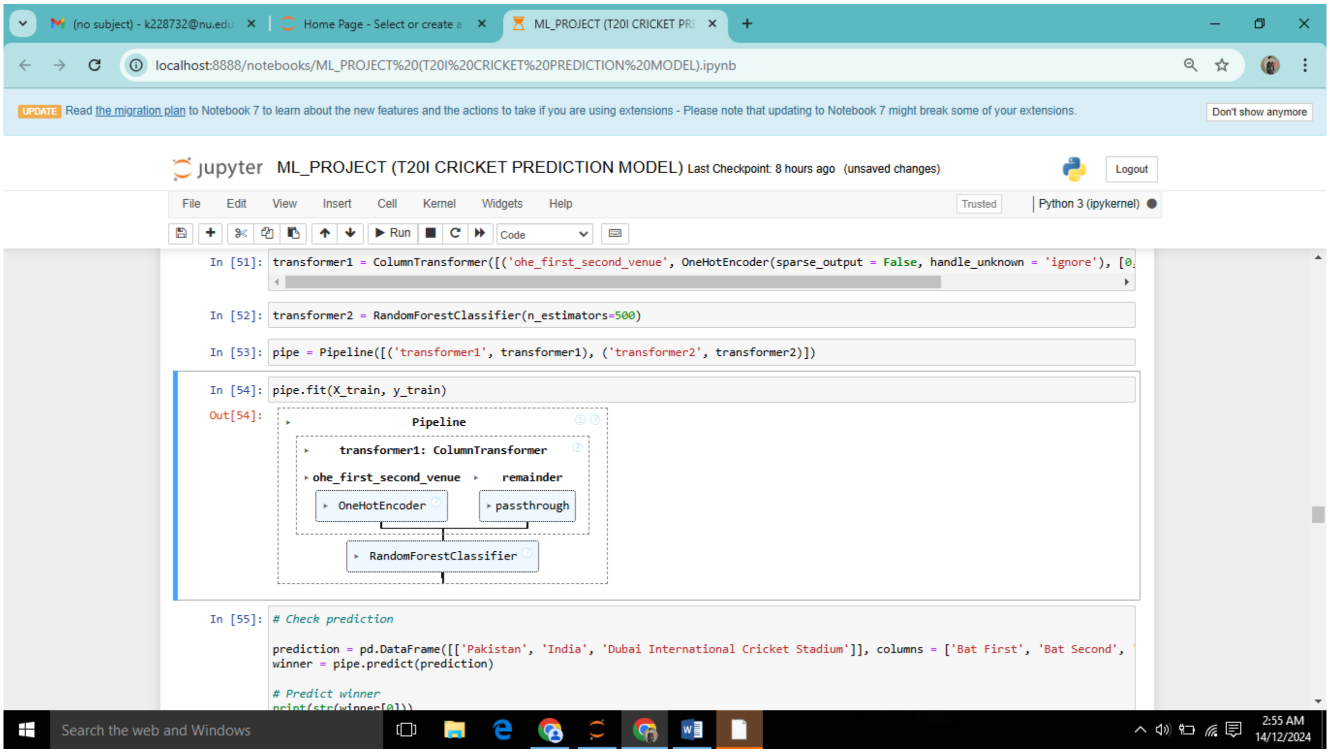
localhost:8888/notebooks/ML_PROJECT%20(T20I%20CRICKET%20PREDICTION%20MODEL).ipynb

jupyter ML_PROJECT (T20I CRICKET PREDICTION MODEL) Last Checkpoint: 8 hours ago (autosaved)

```
In [42]: # Spliting The Data Into Test Train

x_train, x_test, y_train, y_test = train_test_split(data.drop(columns = ['Winner']), data['Winner'], test_size = 0.10, random_sta
```

```
In [43]: # Plotting The Accuracy Graph (Optional Does Not Effect Actual Prediction)

scores = []

# Varying The n_estimators By 20 from 2 - 202
jump = 20
for k in range(2, 202, jump):

    # Creating RandomForestClassifer Object With "K" N_Estimators
    rfc = RandomForestClassifier(n_estimators = k)

    # Training The Model
    rfc.fit(x_train, y_train)

    # 5.5) Predicting And Storing The Accuracy At Different N_Estimators
    y_pred = rfc.predict(x_test)
    scores.append(round(accuracy_score(y_test, y_pred), 4))

print(scores)

[0.9476, 0.9476, 0.9501, 0.9476, 0.9476, 0.9476, 0.9476, 0.9501, 0.9501, 0.9501]
```

```
In [44]: # Accuracy Graph (x-axis = n_estimators, y-axis = accuracy)

plt.plot(range(2, 202, jump), scores)
plt.xlabel('Value of n_estimators for Random Forest Classifier')
plt.ylabel('Testing Accuracy')

Out[44]: Text(0, 0.5, 'Testing Accuracy')
```



```
In [51]: transformer1 = ColumnTransformer([('ohe_first_second_venue', OneHotEncoder(sparse_output = False, handle_unknown = 'ignore'), [0,

In [52]: transformer2 = RandomForestClassifier(n_estimators=500)

In [53]: pipe = Pipeline([('transformer1', transformer1), ('transformer2', transformer2)])

In [54]: pipe.fit(X_train, y_train)
Out[54]:
```

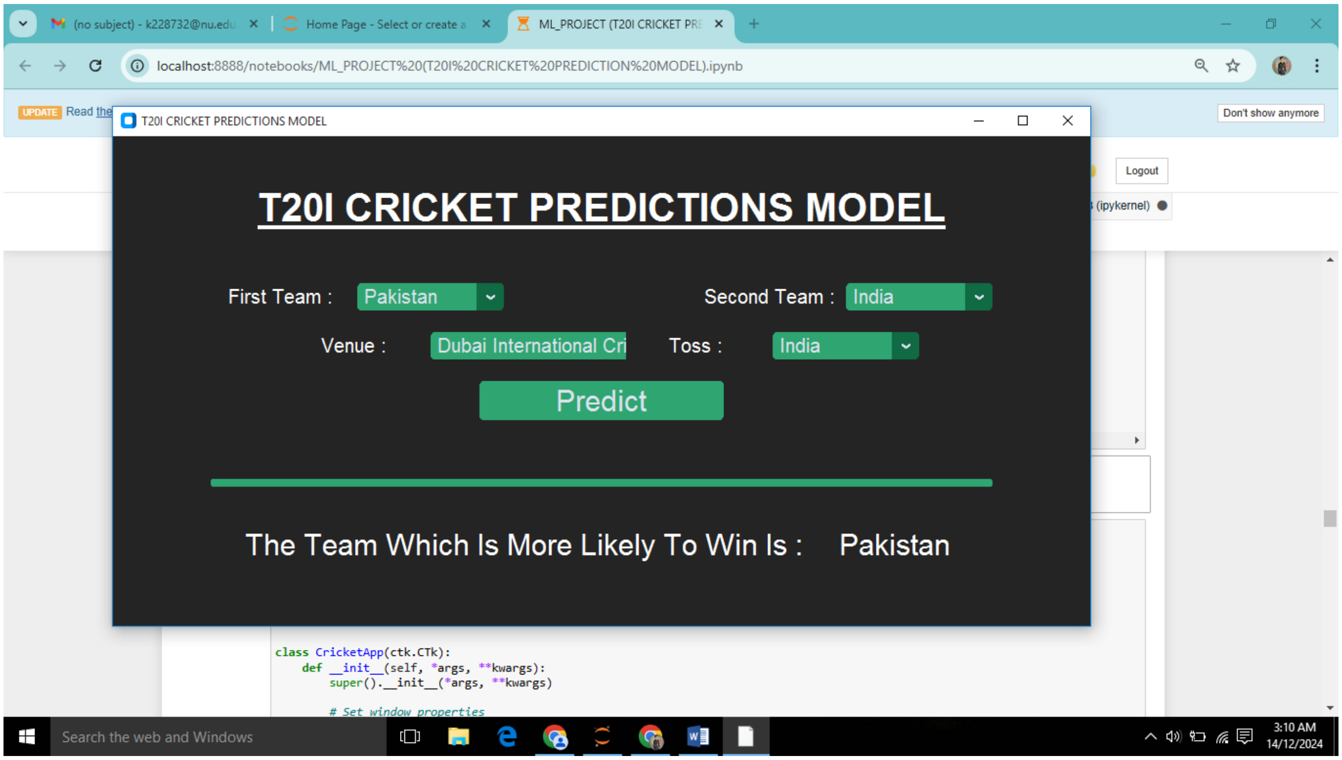
```
In [55]: # Check prediction

         prediction = pd.DataFrame([['Pakistan', 'India', 'Dubai International Cricket Stadium']], columns = ['Bat First', 'Bat Second', '
         winner = pipe.predict(prediction)

         # Predict winner
         print(str(winner[0]))
```

Creating GUI For Prediction and Statistics Of Matches

1. Match Outcome Prediction

- Functionality: Users can select:
  - Team 1
  - Team 2
  - Venue
  - Toss Prediction (the team bat first)

Result: The model predicts the winning team based on past match winning statistics on this specific venue.



(no subject) - k228732@nu.edu
Home Page - Select or create a
ML_PROJECT (T20I CRICKET PRE
localhost:8888/notebooks/ML_PROJECT%20(T20I%20CRICKET%20PREDICTION%20MODEL).ipynb
UPDATE Read the
Don't show anymore
Logout
(ipykernel)
T20I CRICKET PREDICTIONS MODEL
Search the web and Windows
3:10 AM
14/12/2024

T20I CRICKET PREDICTIONS MODEL

First Team : Pakistan

Second Team : India

Venue : Dubai International Cri

Toss : India

Predict

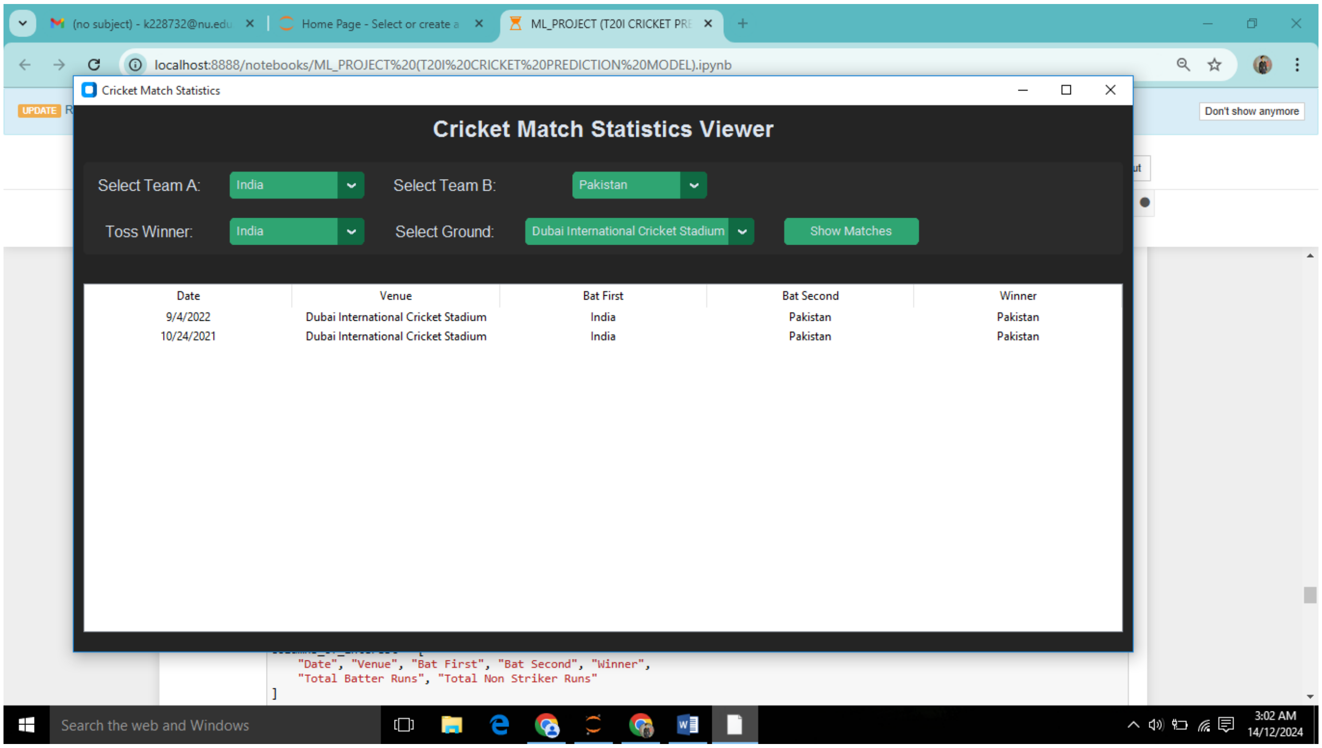
The Team Which Is More Likely To Win Is : Pakistan

```
class CricketApp(ctk.CTk):
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)

        # Set window properties
```

2. CHECKING OUR PREDICTION BY PAST MATCHES STATISTICS OF THIS SPECIFIC SELECTED FIELDS (GUI)

Feature: Displays all historical match data for the selected teams.

Purpose: Users can verify the accuracy of model predictions by comparing them with actual outcomes.


(no subject) - k228732@nu.edu
Home Page - Select or create a
ML_PROJECT (T20I CRICKET PR
localhost:8888/notebooks/ML_PROJECT%20(T20I%20CRICKET%20PREDICTION%20MODEL).ipynb
UPDATE
Don't show anymore
Cricket Match Statistics
Show Matches
Search the web and Windows
3:02 AM
14/12/2024

Cricket Match Statistics Viewer

Select Team A: India
Select Team B: Pakistan
Toss Winner: India
Select Ground: Dubai International Cricket Stadium

Date	Venue	Bat First	Bat Second	Winner
9/4/2022	Dubai International Cricket Stadium	India	Pakistan	Pakistan
10/24/2021	Dubai International Cricket Stadium	India	Pakistan	Pakistan

```
    "Date", "Venue", "Bat First", "Bat Second", "Winner",
    "Total Batter Runs", "Total Non Striker Runs"
]
```

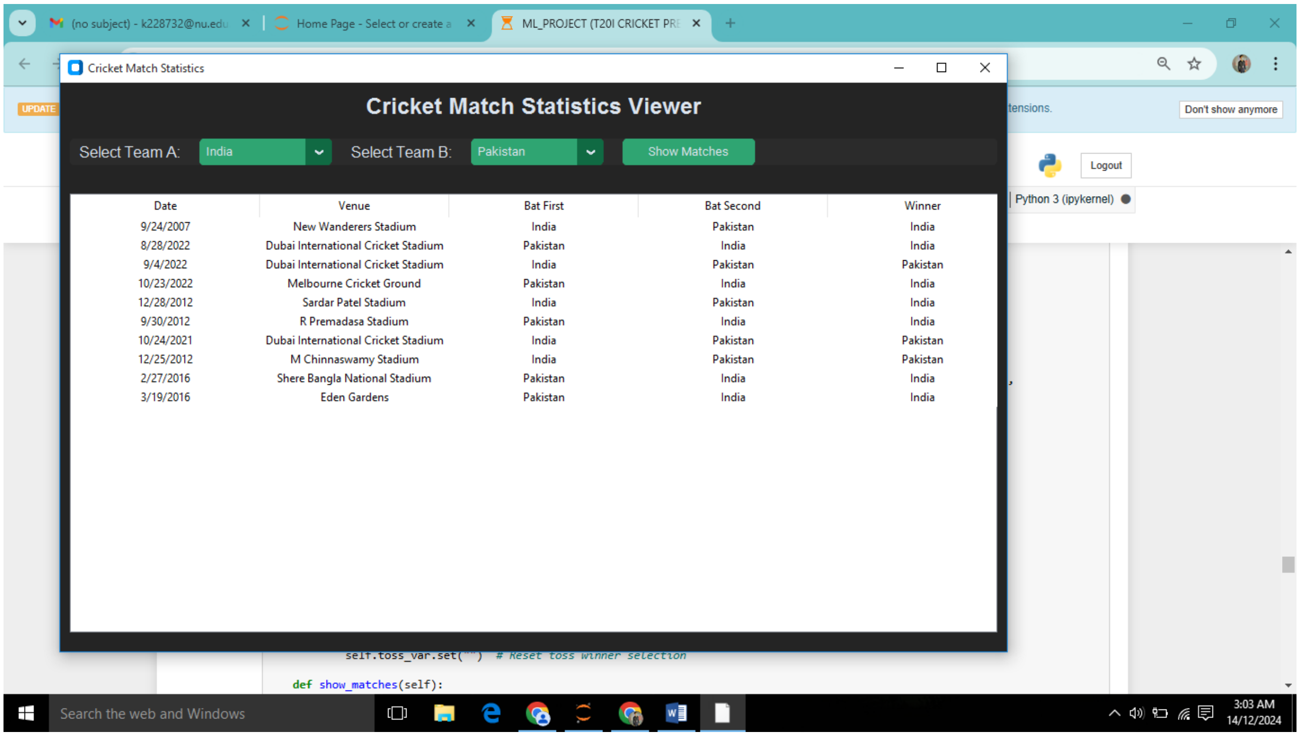
Testing Prediction:

- ✓ 2 matches played in the past between India and Pakistan at Dubai International Stadium where India was bat first in both games and Pakistan were won the both matches. Now when I am predicting for future match with same venue and other statistics. Our model is predicting Winner as "Pakistan".

3. Ground Specific Statistics (GUI)

Feature: Users select Team 1, Team 2, and a specific venue.

Output: Displays past match statistics specific to the selected ground to help users understand performance trends.



Date	Venue	Bat First	Bat Second	Winner
9/24/2007	New Wanderers Stadium	India	Pakistan	India
8/28/2022	Dubai International Cricket Stadium	Pakistan	India	India
9/4/2022	Dubai International Cricket Stadium	India	Pakistan	Pakistan
10/23/2022	Melbourne Cricket Ground	Pakistan	India	India
12/28/2012	Sardar Patel Stadium	India	Pakistan	India
9/30/2012	R Premadasa Stadium	Pakistan	India	India
10/24/2021	Dubai International Cricket Stadium	India	Pakistan	Pakistan
12/25/2012	M Chinnaswamy Stadium	India	Pakistan	Pakistan
2/27/2016	Shere Bangla National Stadium	Pakistan	India	India
3/19/2016	Eden Gardens	Pakistan	India	India

```
    self.toss_var.set("")  # Reset toss winner selection

def show_matches(self):
```

MODEL DEVELOPMENT AND ANALYSIS

1. DATASET PREPROCESSING (PERFORMED EDA)

- Aggregated the dataset by date to ensure each match is represented as a single entry.
- Eliminated duplicate rows and retained only relevant features.

2. FEATURE ENGINEERING AND SCALING

- Encoded categorical variables such as teams and venues using one-hot encoding and label encoding.p
- Implement Standard Scaler to make sure the data is in small range.

3. Splitting In Training and Testing Data:

- Split the dataset into training (80%) and testing (20%) sets.

Results and Conclusion

- The Random Forest Classifier emerged as the best model, achieving a 95% testing accuracy.
- GUI enhancements provide a user-friendly interface for exploring predictions and historical data.