Final report :

Conclusion drawn

Difficulties encountered during the project

- **What was the main scientific obstacle encountered during this project?**

The toughest part of this project was figuring out the data and making the code work. The data was a bit tricky – to understand at first but also to analyze, and getting the code to run without errors took a lot of effort. It felt like solving puzzles, but every problem I faced helped me get better at understanding the data and coding. The project was a big challenge for me, but I also felt that I learned a lot from it.

- **For each of the following points, if you encountered difficulties, detail how they slowed you down in setting up your project.**

  - **Forecast: tasks that took longer than expected, etc.**

    Cleaning the dataset/getting it ready for modeling. Running the machine learning algorithms (I had to fix errors all the time). I had to go back and update some of the preprocessing steps once I started working on the modeling part because I realized that some of my assumptions no longer suit me. Interpreting the results also took more time than I expected.

  - **Datasets: acquisition, volumetry, processing, aggregation, etc.**

    The dataset was not easy/intuitive to understand at first, I spent a lot of time learning what the variables are and how to use them for what. The dataset was also huge, I realized too late that maybe I should have used only part of the data (sorted maybe by date) to make the code run easier and to interpret the results easier. The preprocessing was challenging too, as I wasn't sure if I'm doing the right thing throughout. Also, picking the target variable and the explanatory ones was difficult. I felt like there's so many directions that the project can go to.

  - **Technical/theoretical skills: timing of skill acquisition, skill not offered in training, etc.**

    I had to go over the course material many times and research lots of things online. And from there, sort of, form my opinion on things. I felt like the project was more complex than the examples we've done in training and often I wasn't sure if I'm doing the right thing.

  - **Relevance: of the approach, model, data, etc.**

    Not sure what this question means.

  - **IT: storage power, computational power, etc.**

I started the project in Google Collab but after the preprocessing part, I started running into issues with the algorithms because there was not enough space/RAM to execute these in the notebook. So, I created a Kaggle account and started executing the code there and it worked better.

Report

- **Detail what was your main contribution to achieving the project's goals.**

My main contribution to achieving the project's goals involved implementing and adapting machine learning models to predict tennis match outcomes. I preprocessed and cleaned the dataset to ensure its quality and relevance for the models. Throughout the project, I was dealing with various challenges, such as analyzing the data, running the code, and trying to fix errors. Despite these obstacles, I followed the steps we learned in the training and applied different strategies to optimize and improve the results. Although, I did not achieve the objective fully (beat the bookmaker's algorithms), I still think it was a successful project.

- **Have you changed the model since the last iteration? If yes, provide details.**

No.

- **Present the results obtained and compare them to the benchmark.**

My most successful model was the AdaBoost model, which gave these results:
Test Accuracy: 0.419200954084675
Precision: 0.4280393161228633
Recall: 0.419200954084675
F1-Score: 0.4173056320120354

I tried to compare the model results to the benchmark (the bookmaker's odds from the initial dataset) but I ran into issues I couldn't resolve. My code looked like this:

```
import pandas as pd
import numpy as np
from sklearn.metrics import accuracy_score, precision_score, recall_score

# 1: calculate probabilities from bookmakers' Odds
df['B365W_Prob'] = 1 / df['B365W']
df['B365L_Prob'] = 1 / df['B365L']
df['PSW_Prob'] = 1 / df['PSW']
df['PSL_Prob'] = 1 / df['PSL']

# 2: compare model predictions to bookmakers probabilities
y_true = df['Winner']  # Actual outcomes
y_pred_model = best_adaboost_model.predict(X_test_final)

# convert model predictions to probabilities
df['Model_Prob_Winner'] = best_adaboost_model.predict_proba(X_test_final)[:, 1]
df['Model_Prob_Loser'] = 1 - df['Model_Prob_Winner']

# check model performance
# a threshold for converting probabilities to binary predictions
threshold = 0.5

# convert probabilities to binary predictions
df['Model_Pred_Winner'] = (df['Model_Prob_Winner'] > threshold).astype(int)
df['Model_Pred_Loser'] = 1 - df['Model_Pred_Winner']

```

```
# accuracy, precision, and recall for both winner and loser
accuracy_winner = accuracy_score(y_true, df['Model_Pred_Winner'])
precision_winner = precision_score(y_true, df['Model_Pred_Winner'])
recall_winner = recall_score(y_true, df['Model_Pred_Winner'])

accuracy_loser = accuracy_score(1 - y_true, df['Model_Pred_Loser'])
precision_loser = precision_score(1 - y_true, df['Model_Pred_Loser'])
recall_loser = recall_score(1 - y_true, df['Model_Pred_Loser'])

# results
print("Winner Metrics:")
print("Model Accuracy:", accuracy_winner)
print("Model Precision:", precision_winner)
print("Model Recall:", recall_winner)

print("\nLoser Metrics:")
print("Model Accuracy:", accuracy_loser)
print("Model Precision:", precision_loser)
print("Model Recall:", recall_loser)
```

But I kept getting errors for this that I just couldn't fix after a few days of trying. I kept updating my preprocessing step to get it to work but it still wouldn't. I also tried another approach using ROI (Return of investment) as a metric and that wouldn't work either. So, I gave up on this.

I found another possibility, which is to use a dummy classifier that predicts the most frequent class in the training data.

```
from sklearn.dummy import DummyClassifier

# create a dummy classifier that predicts the most frequent class
dummy_classifier = DummyClassifier(strategy='most_frequent')
dummy_classifier.fit(X_train_final, y_train)
y_dummy_pred = dummy_classifier.predict(X_test_final)

# evaluate dummy classifier predictions
accuracy_dummy = accuracy_score(y_test, y_dummy_pred)
precision_dummy = precision_score(y_test, y_dummy_pred, average='weighted')
recall_dummy = recall_score(y_test, y_dummy_pred, average='weighted')
f1_dummy = f1_score(y_test, y_dummy_pred, average='weighted')

# results
print("Dummy (Most Frequent) Metrics:")
print("Accuracy:", accuracy_dummy)
print("Precision:", precision_dummy)
print("Recall:", recall_dummy)
print("F1-Score:", f1_dummy)

print("\nAdaBoost Metrics:")
print("Accuracy:", accuracy_adaboost)
print("Precision:", precision_adaboost)
print("Recall:", recall_adaboost)
print("F1-Score:", f1_adaboost)
```

Results:
Dummy (Most Frequent) Metrics:
Accuracy: 0.022063208109719738
Precision: 0.0004867851520928028
Recall: 0.022063208109719738
F1-Score: 0.0009525539090544109

AdaBoost Metrics:
Accuracy: 0.419200954084675
Precision: 0.4280393161228633
Recall: 0.419200954084675
F1-Score: 0.4173056320120354

These results indicate that the model performs better than the benchmark but it isn't as certain as the direct comparison that I wanted to have working initially.

- **For each of the project's goals, detail how they were achieved or not.**

In terms of following the steps of the Analysis process, I think I achieved my goal to analyze, clean, preprocess the data and get results from the machine learning models. I also ran multiple models and tried to understand their results afterwards. This gave me the opportunity to look for ways to improve the most promising model. The main part that I wasn't satisfied with is not being able to run the comparison correctly to see how well my model performs compared to the bookmaker's presented in the initial dataset.

- **If they have been reached, in which process(es) can your model fit? Detail.**

Data Analysis and Preprocessing: *Successfully analyzed and cleaned the data, preparing for machine learning modeling.*

Modeling and Evaluation: *Ran multiple machine learning models, analyzed their results, and worked on improving the most promising one.*

Model Comparison and Improvement: *It was challenging running the comparison, but there was a clear intention to assess the model against bookmaker predictions.*

Continuation of the project

- **What avenues for improvement do you suggest to increase the performance of your model?**

  - *Fine-tuning parameters in the AdaBoost model*
  - *Investigate the data more thoroughly to identify patterns or outliers that I might have overlooked*
  - *I would also consider changing the preprocessing to only look at data from a certain date and ignore the very old matches*
  - *Try suggested techniques that I haven't (SHAP, LIME, Skater)*
  - *Make the comparison work as intended.*

- **How has your project contributed to an increase in scientific knowledge?**

Not sure. Probably not, but it increased my knowledge ☺