

# ПРОГНОЗУВАННЯ ПЕРЕМОЖЦЯ У РАУНДІ НА ОСНОВІ ПОКАЗНИКІВ СТАНУ ІГРОВОГО ПОЛЯ У ГРІ COUNTER-STRIKE: GLOBAL OFFENSIVE

**Паламарчук О.О. ІП-13**

# МЕТА

Знайти метод, який зможе найкраще прогнозувати переможця у раунді базуючись на інформації про стан ігрового поля.

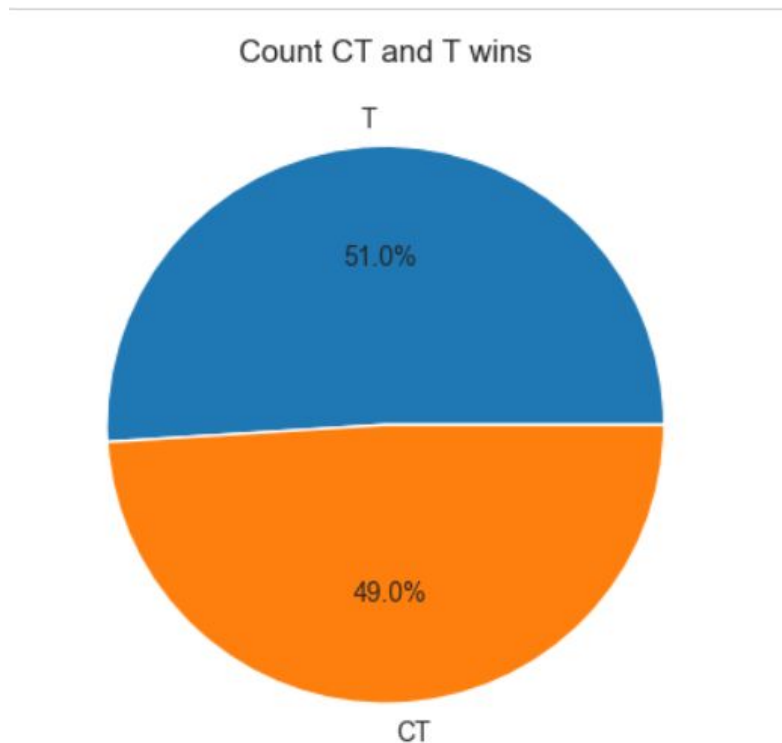
# ПЛАН РОБОТИ

1. Проаналізувати предметну область;
2. Провести попередню роботу з даними;
3. Провести інтелектуальний аналіз даних;
4. Побудувати та проаналізувати моделі для прогнозування переможця у раунді на основі показників стану ігрового поля у грі Counter-Strike: Global Offensive;
5. Порівняти отримані результати;

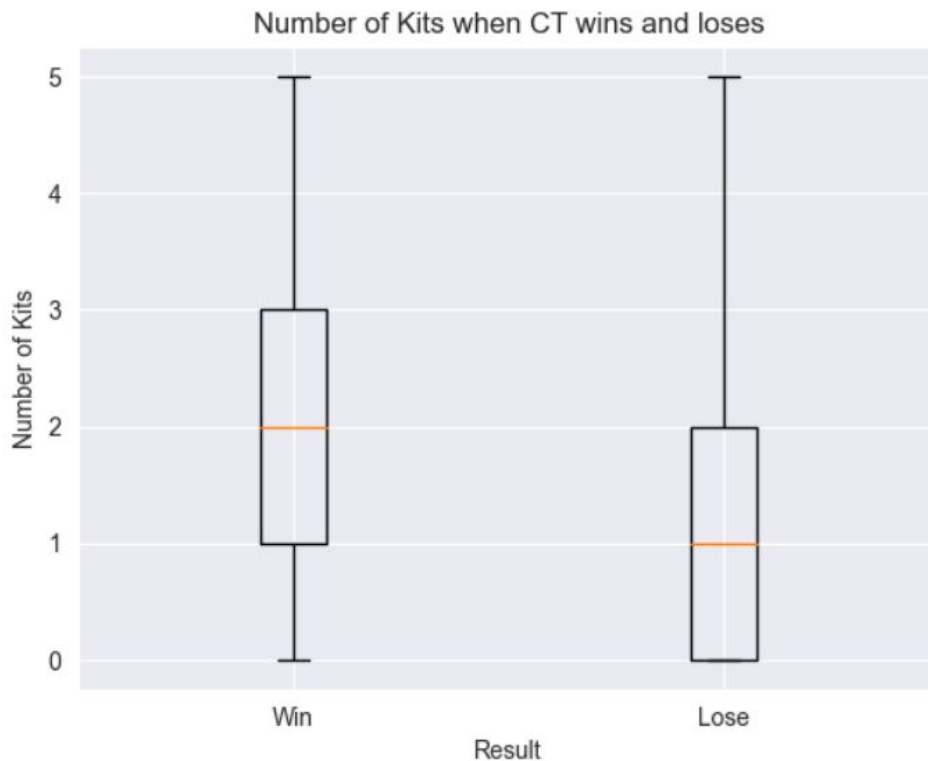
# ПЕРЕЛІК ЛІТЕРАТУРИ

1. Python. [Електронний ресурс] – URL: <https://www.python.org/>
2. Pandas. [Електронний ресурс] – URL: <https://pandas.pydata.org/docs/>
3. Matplotlib. [Електронний ресурс] – URL: <https://matplotlib.org/stable/>
4. Sklearn. [Електронний ресурс] – URL: [https://devdocs.io/scikit\\_learn/](https://devdocs.io/scikit_learn/)
5. Seaborn. [Електронний ресурс] – URL: <https://seaborn.pydata.org/>
6. NumPy. [Електронний ресурс] – URL: <https://numpy.org/>

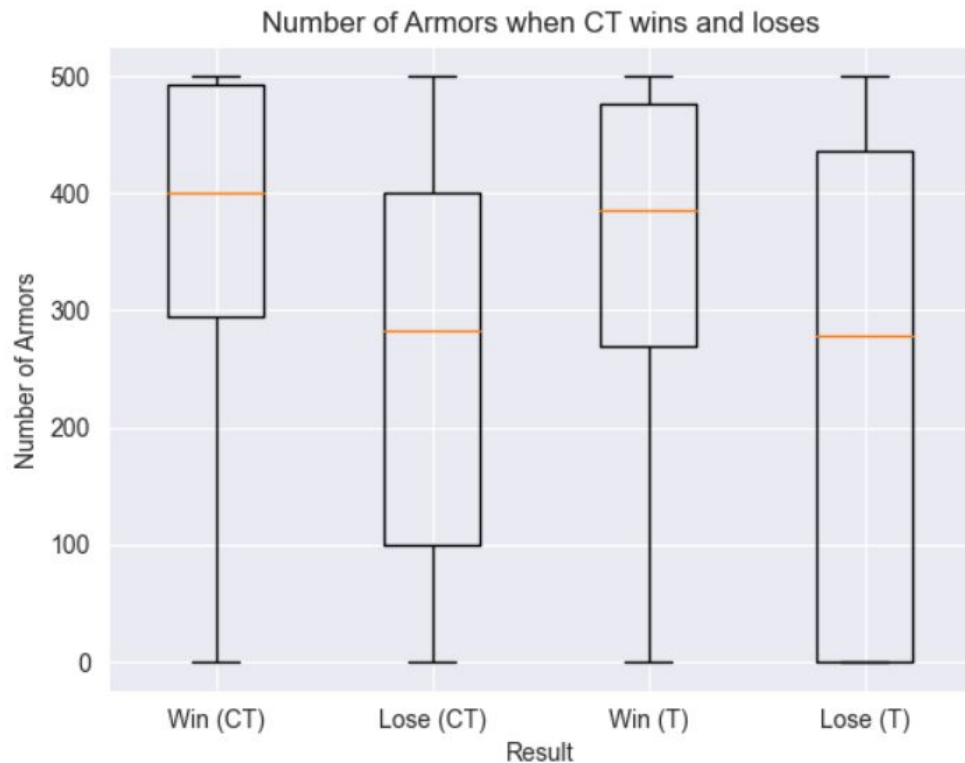
# ПОРІВНЯННЯ КІЛЬКОСТІ ПЕРЕМОГ ЗА СТОРОНИ СТ ТА Т



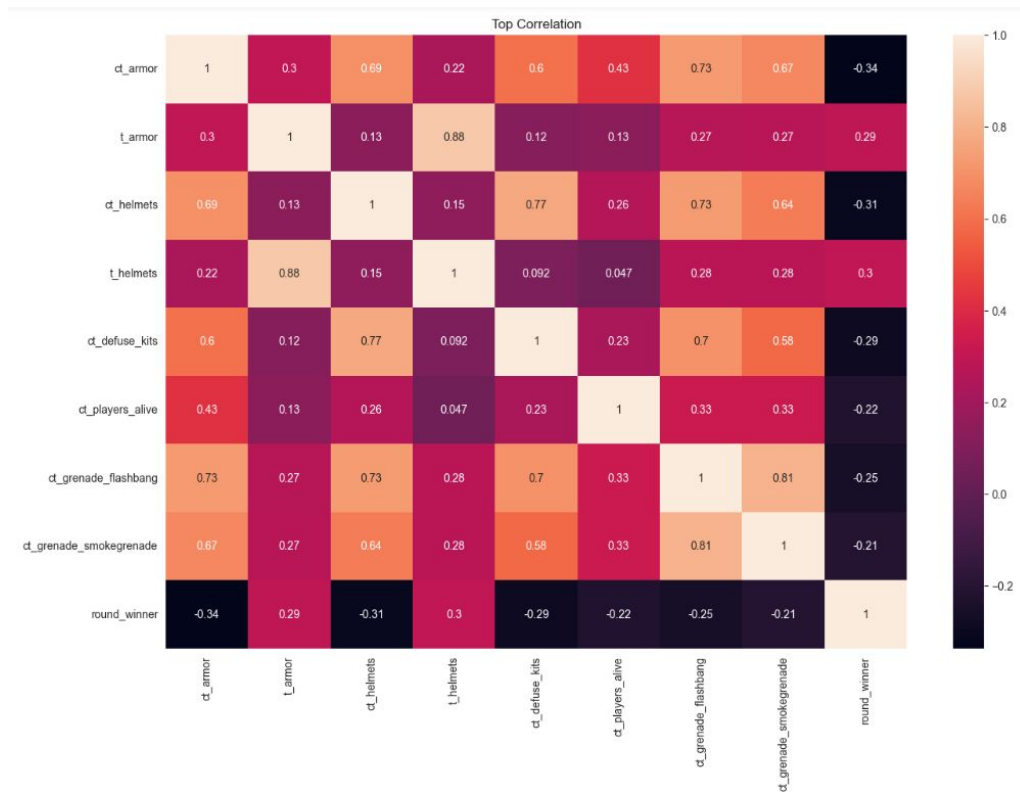
# РОЗПОДІЛ КІЛЬКОСТІ НАБОРІВ ДЛЯ ЗНЕШКОДЖЕННЯ ВИБУХІВКИ В ЗАЛЕЖНОСТІ ВІД ФАКТУ ПЕРЕМОГИ І ПОРАЗКИ СТОРОНИ СТ



# РОЗПОДІЛ КІЛЬКОСТІ БРОНІ В ЗАЛЕЖНОСТІ ВІД ФАКТУ РЕЗУЛЬТАТУ РАУНДУ



# МАТРИЦЯ ТОП КОРЕЛЯЦІЇ ДЛЯ ХАРАКТЕРИСТИК





# ОБГРУНТУВАННЯ ОБРАНИХ МЕТОДІВ

Вибір методу Random Forest був зроблений із розрахунку на те, що даний алгоритм ефективно працює із невеликою кількістю класів і великою кількістю характеристик та знаходить найбільш інформативні з них для класифікації, що відповідає нашому випадку, оскільки у нас є багато характеристик в наборі даних, таких як кількість гравців, їхній стан здоров'я, наявність зброї тощо, а також лише два класи СТ та Т.

# ОБГРУНТУВАННЯ ОБРАНИХ МЕТОДІВ

Вибір методу Decision Tree був зроблений із розрахунку, що він є простим і інтерпретованим алгоритмом, який може допомогти зрозуміти, які ознаки є найбільш важливими для класифікації. У нашому випадку такими ознаками є використання різного спорядження, факт про те, чи встановлено бомбу, кількість здоров'я та броні у гравців тощо. Дерево рішень може допомогти виявити ключові фактори, які впливають на переможця раунду.

# ОБГРУНТУВАННЯ ОБРАНИХ МЕТОДІВ

Вибір методу K-Nearest Neighbors був зроблений із розрахунку, що даний алгоритм є простим алгоритмом, який використовує найближчих сусідів для класифікації нових прикладів. У нашому випадку, можна сподіватися, що стиль гри і тактика команди, що програє або перемагає, можуть мати подібні риси та характеристики. Використання k-Найближчих сусідів може допомогти знайти схожі групи прикладів і виробити прогнози на основі їхнього класу.

# МОДЕЛЬ RANDOM FOREST

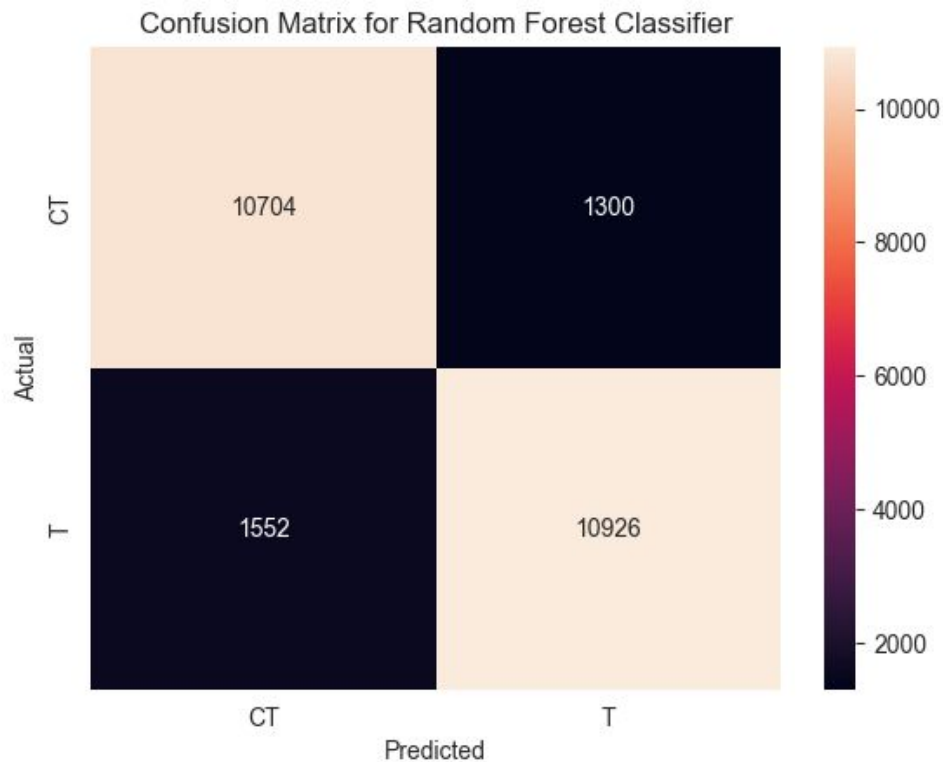
```
: rf_classifier = RandomForestClassifier()
rf_grid_search = GridSearchCV(rf_classifier, param_grid={'n_estimators': [10, 50, 100, 200, 500]}, cv=5, scoring='accuracy')
rf_grid_search.fit(X_train, y_train)

rf_model = rf_grid_search.best_estimator_
Y_pred = rf_model.predict(X_test)

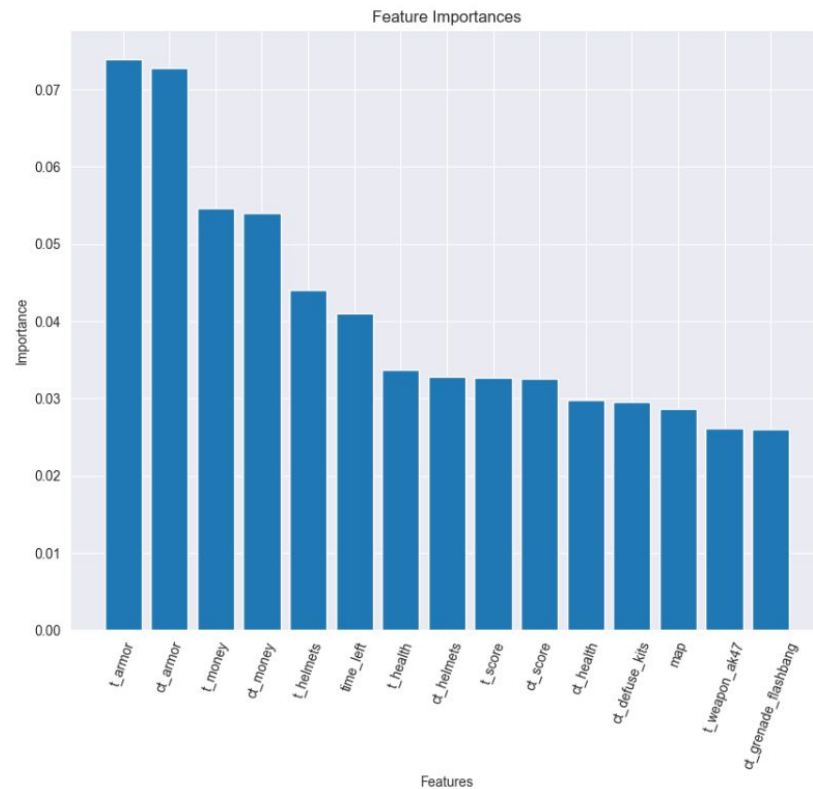
print("Random Forest Classifier")
print('MSE: %.2f' % mean_squared_error(y_test, Y_pred))
print('R2 score: %.2f' % r2_score(y_test, Y_pred))
print('Accuracy score: %.2f' % accuracy_score(y_test, Y_pred))
```

```
Random Forest Classifier
MSE: 0.12
R2 score: 0.53
Accuracy score: 0.88
```

# МОДЕЛЬ RANDOM FOREST



# МОДЕЛЬ RANDOM FOREST



# МОДЕЛЬ DECISION TREE

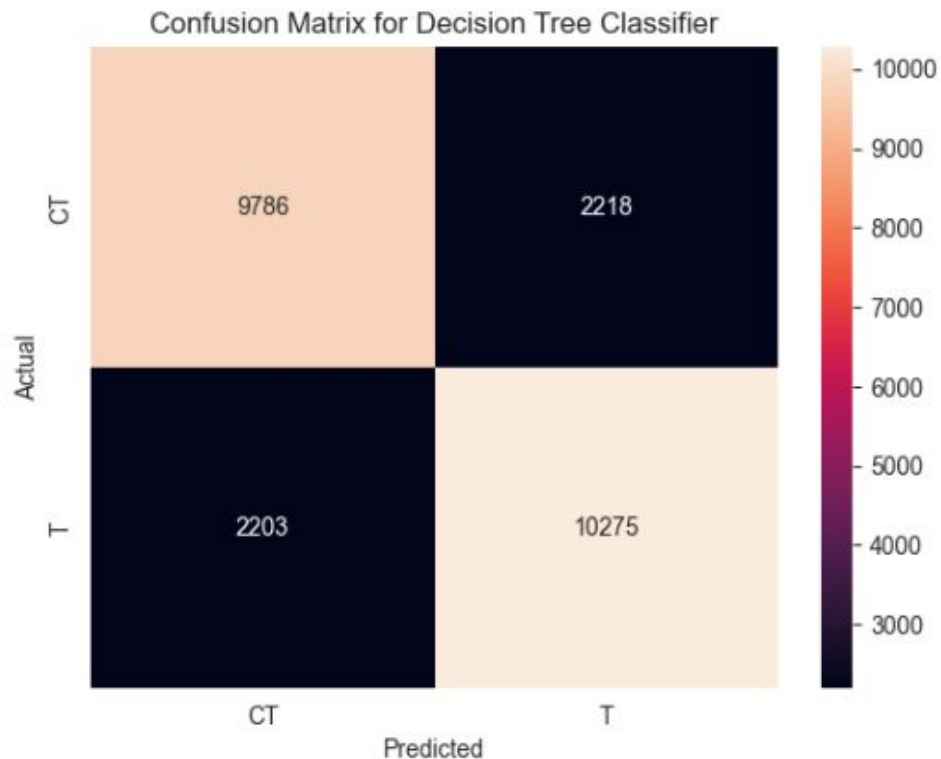
```
dt_classifier = DecisionTreeClassifier()
dt_grid_search = GridSearchCV(dt_classifier, {'max_depth': [i for i in range(40, 50)]}, cv=5, scoring='accuracy')
dt_grid_search.fit(X_train, y_train)

dt_model = dt_grid_search.best_estimator_
Y_pred = dt_model.predict(X_test)

print("Decision Tree Classifier")
print('MSE: %.2f' % mean_squared_error(y_test, Y_pred))
print('R2 score: %.2f' % r2_score(y_test, Y_pred))
print('Accuracy score: %.2f' % accuracy_score(y_test, Y_pred))
```

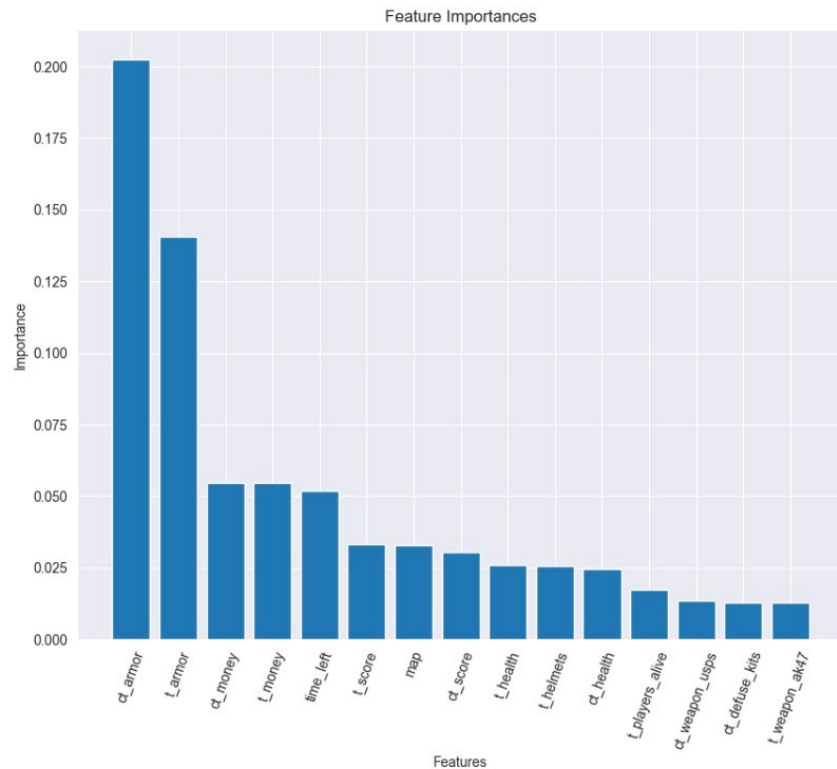
Decision Tree Classifier  
MSE: 0.18  
R2 score: 0.28  
Accuracy score: 0.82

# МОДЕЛЬ DECISION TREE

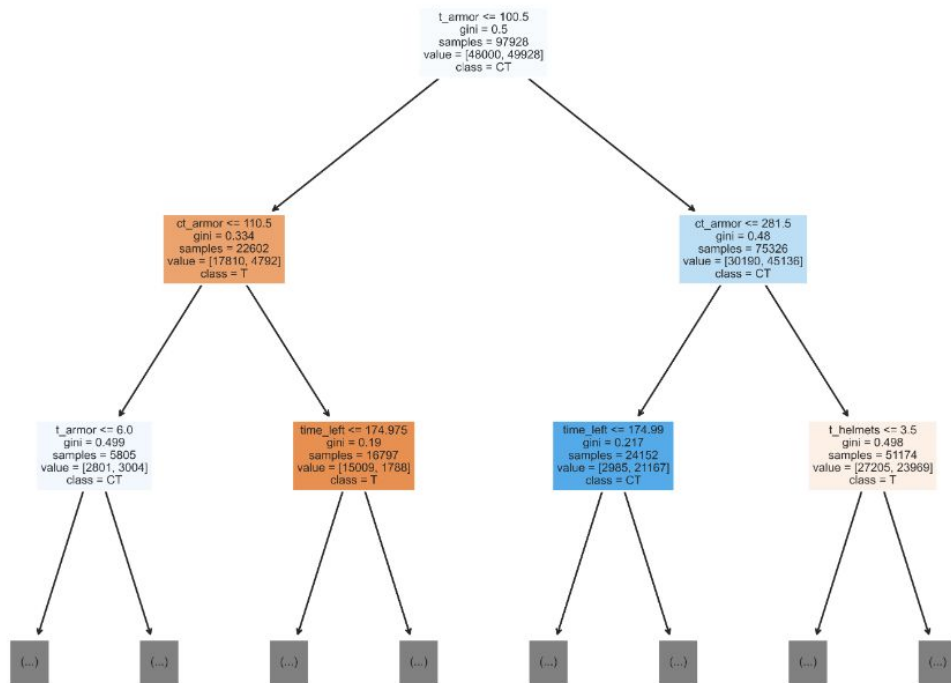




# МОДЕЛЬ DECISION TREE



# МОДЕЛЬ DECISION TREE



# МОДЕЛЬ K-NEAREST NEIGHBORS

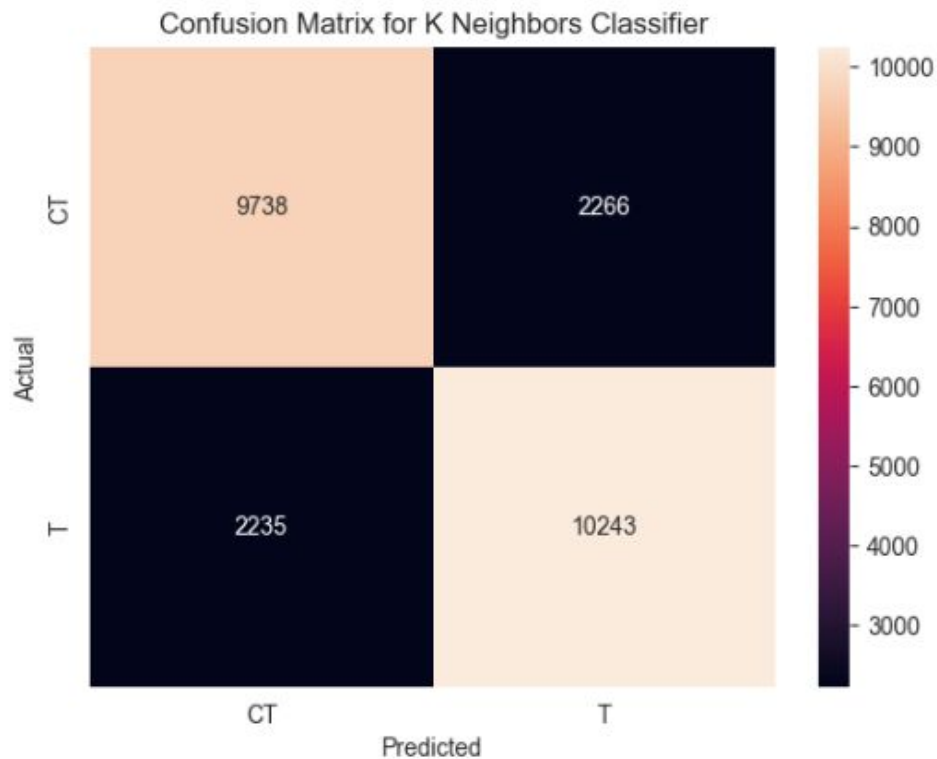
```
: kn_model_classifier = KNeighborsClassifier()
kn_model_grid_search = GridSearchCV(kn_model_classifier, {'n_neighbors': [1, 5, 10, 15]}, cv=5, scoring='accuracy')
kn_model_grid_search.fit(X_train, y_train)

kn_model = kn_model_grid_search.best_estimator_
Y_pred = kn_model.predict(X_test)

print("K Neighbors Classifier")
print('MSE: %.2f' % mean_squared_error(y_test, Y_pred))
print('R2 score: %.2f' % r2_score(y_test, Y_pred))
print('Accuracy score: %.2f' % accuracy_score(y_test, Y_pred))
```

```
K Neighbors Classifier
MSE: 0.18
R2 score: 0.26
Accuracy score: 0.82
```

# МОДЕЛЬ K-NEAREST NEIGHBORS



# ПОРІВНЯННЯ МОДЕЛЕЙ

score/model	Random Forest	Decision Tree	K-Nearest Neighbors
MSE	0.12	0.18	0.18
R2	0.53	0.28	0.26
Accuracy	0.88	0.82	0.82

# ПОРІВНЯННЯ МОДЕЛЕЙ

score/model	Random Forest	Decision Tree	K-Nearest Neighbors
G	0.116494	0.180582	0.183849

# ВИСНОВОК

Отже, було проаналізовано предметну область, проведено попередню роботу з даними, проведено інтелектуальний аналіз даних, побудовано та проаналізовано моделі для прогнозування переможця у раунді на основі показників стану ігрового поля у грі Counter-Strike, також було визначено найкращу модель Random Forest за характеристиками MSE, Accuracy score та оцінкою G.

ДЯКУЮ ЗА УВАГУ!!!