Weekly Report March 30

This week we were able to find a suitable solution for our problem using Decision Tree Regression, This allowed us to accurately predict the output variable. However, the level of accuracy we had was suspiciously accurate. The highly level of accuracy usually indicates Overfitting however, despite cross validation the data seemed to work well with the test and validation data. However, Due to the simplistic nature of the data and the problem we might be running into such issues, Because we have a model that can with 99% accuracy predict all outputs. So we are attempting to complicate the data set a little bit to test our model further for this we have decide to introduce a few more features.

Below is our code for error statistics and model:

import pandas as pd

from sklearn.tree import DecisionTreeRegressor

import numpy as np

import cv2

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, r2\_score

import matplotlib.pyplot as plt

df = pd.read\_excel('C:\\Users\\Kabir\\Downloads\\TestDS.xlsx')

print(df)

actual\_angles = df['Actual/Real Angle']

reported\_angles = df['Reported Angle by cv2.minAreaRect()']

point\_A\_X = df['Point\_A\_X']

point\_A\_Y = df['Point\_A\_Y']

point\_B\_X = df['Point\_B\_X']

point\_B\_Y = df['Point\_B\_Y']

point\_C\_X = df['Point\_C\_X']

point\_C\_Y = df['Point\_C\_Y']

point\_D\_X = df['Point\_D\_X']

point\_D\_Y = df['Point\_D\_Y']

print(reported\_angles)

X = np.column\_stack((reported\_angles, point\_A\_X, point\_A\_Y, point\_B\_X, point\_B\_Y, point\_C\_X, point\_C\_Y, point\_D\_X, point\_D\_Y))

y = actual\_angles

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = DecisionTreeRegressor()

model.fit(X\_train, y\_train)

y\_pred\_test = model.predict(X\_test)

mae\_test = mean\_absolute\_error(y\_test, y\_pred\_test)

mse\_test = mean\_squared\_error(y\_test, y\_pred\_test)

rmse\_test = np.sqrt(mse\_test)

r2\_test = r2\_score(y\_test, y\_pred\_test)

print("Mean Absolute Error (MAE) on testing set:", mae\_test)

print("Mean Squared Error (MSE) on testing set:", mse\_test)

print("Root Mean Squared Error (RMSE) on testing set:", rmse\_test)

print("Coefficient of determination (R^2) on testing set:", r2\_test)

# Visualize predictions

plt.scatter(y\_test, y\_pred\_test)

plt.xlabel("Actual Output Angle")

plt.ylabel("Predicted Output Angle")

plt.title("Actual vs Predicted Output Angles (Testing Set)")

plt.show()