```python
import tensorflow as tf
import pandas as pd
import numpy as np

x = pd.read_csv('FB_predictions/x_train.csv')
y = pd.read_csv('FB_predictions/y_train(landslide_rate).csv')
df = pd.read_csv('FB_predictions/df.csv')

print("null values at x: " + str(x.isnull().sum()) +
      "null values at y: " + str(y.isnull().sum()) + "\n")
print(x)
print(y)

xData = []
yData = y.values

for i, rows in x.iterrows():
    xData.append([rows])

print(xData)
print(yData)

model = tf.keras.models.Sequential([
    tf.keras.layers.Dense(64, activation='tanh'),
    tf.keras.layers.Dense(128, activation='tanh'),
    tf.keras.layers.Dense(256, activation='tanh'),
    tf.keras.layers.Dense(512, activation='tanh'),
    tf.keras.layers.Dense(256, activation='tanh'),
    tf.keras.layers.Dense(128, activation='tanh'),
    tf.keras.layers.Dense(64, activation='tanh'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
model.fit(np.array(xData), np.array(yData), epochs=5000)

# makes x values for prediction
team_wins = []

for j in range(146):
    team_win = pd.DataFrame(columns=df.columns)
    for i in range(1355):
        if df.iloc[i, j + 12] == 1:
            team_win.loc[len(team_win.index)] = df.iloc[i]
    team_wins.append(team_win)

team_predictions_winner = []
for i in range(146):
    team_predictions_winner.append(model.predict(team_wins[i].drop('landslide', axis=1)))

team_loses = []

for j in range(146):
    team_lose = pd.DataFrame(columns=df.columns)
    for i in range(1355):
        if df.iloc[i, j + 158] == 1:
            team_lose.loc[len(team_lose.index)] = df.iloc[i]
    team_loses.append(team_lose)

team_predictions_loser = []
for i in range(146):
    team_predictions_loser.append(model.predict(team_loses[i].drop('landslide', axis=1)))


def calc_landslide_rate(team_predictions):
    landslide_rate = []
    for i in range(len(team_predictions)):
        avg = 0
        for j in range(len(team_predictions[i])):
            print(team_predictions[i][j])
            avg += team_predictions[i][j]
        avg /= len(team_predictions[i])
        landslide_rate.append(avg[0])
    return landslide_rate
```

```
landslide_rate_win = calc_landslide_rate(team_predictions_winner)
landslide_rate_lose = calc_landslide_rate(team_predictions_loser)

print(landslide_rate_win)
print(landslide_rate_lose)
landslideLogDf = pd.DataFrame({'prob_landslide_winning': landslide_rate_win,
                               'prob_landslide_losing': landslide_rate_lose},
                              columns=['prob_landslide_winning', 'prob_landslide_losing'])

landslideLogDf.to_csv('FB_predictions/DL_football_prediction_winner.csv', index=False)
```