#Code cell <l4v-WiJNLCPT>

# %% [code]

# %% [code]

import pandas as pd

import matplotlib.pyplot as plt

# Arrays from the table

average\_mass = [1.678, 1.586, 1.658, 1.617, 1.590, 1.655, 1.582, 1.675, 1.526, 1.697]

ann\_average\_mass = [1.66, 1.58, 1.66, 1.62, 1.61, 1.60, 1.53, 1.66, 1.46, 1.64]

rf\_average\_mass = [1.65, 1.60, 1.65, 1.62, 1.60, 1.64, 1.58, 1.66, 1.55, 1.67]

ada\_boost\_average\_mass = [1.667, 1.586, 1.662, 1.618, 1.604, 1.654, 1.582, 1.657, 1.526, 1.683]

# Creating DataFrame

data = {

'Density': average\_mass,

'ANN Density\*\*': ann\_average\_mass,

'RF Density\*\*': rf\_average\_mass,

'ADA Boost Density\*\*': ada\_boost\_average\_mass

}

df\_average\_mass = pd.DataFrame(data)

# Plotting

plt.figure(figsize=(14, 7))

plt.plot(df\_average\_mass['Density'], label='Actual Index of Refraction', marker='o', linewidth=3, markersize=8, color='black')

plt.plot(df\_average\_mass['ANN Density\*\*'], label='ANN Index of Refraction', marker='x', linestyle='--', color='red')

plt.plot(df\_average\_mass['RF Density\*\*'], label='RF Index of Refraction', marker='s', linestyle='--' , color='blue')

plt.plot(df\_average\_mass['ADA Boost Density\*\*'], label='ADA Boost Index of Refraction', marker='d', linestyle='--', color='orange')

# Adding annotations to highlight differences with adjusted positions

for i, (actual, ann, rf, ada) in enumerate(zip(df\_average\_mass['Density'], df\_average\_mass['ANN Density\*\*'], df\_average\_mass['RF Density\*\*'], df\_average\_mass['ADA Boost Density\*\*'])):

plt.annotate(f'{actual}', (i, actual), textcoords="offset points", xytext=(0,10), ha='center', fontsize=8, color='black')

plt.annotate(f'{ann}', (i, ann), textcoords="offset points", xytext=(15,-10), ha='center', fontsize=8, color='blue')

plt.annotate(f'{rf}', (i, rf), textcoords="offset points", xytext=(-15,-10), ha='center', fontsize=8, color='green')

plt.annotate(f'{ada}', (i, ada), textcoords="offset points", xytext=(15,-20), ha='center', fontsize=8, color='red')

plt.xlabel('Sample Index')

plt.ylabel('Index of Refraction')

# plt.title('Comparison of Actual and Predicted Average Mass')

plt.legend()

plt.grid(True)

plt.ylim(1.4,2) # Adjusting y-axis limits to focus on the range of values

plt.show()

AI

<https://app.stealthwriter.ai/humanizer>