docs/assignments/assignment 1/all_splits_info_gain.py

```python
import pandas as pd
import numpy as np
from math import log2
import os
# Define the dataset
data = {
    'Instance': [1, 2, 3, 4, 5, 6, 7, 8, 9],
    'A1': ['T', 'T', 'T', 'F', 'F', 'F', 'F', 'T', 'F'],
    'A2': ['T', 'T', 'F', 'F', 'T', 'T', 'F', 'F', 'T'],
    'A3': [1.0, 6.0, 5.0, 4.0, 7.0, 3.0, 8.0, 7.0, 5.0],
    'Target Class': ['+', '+', '-', '+', '-', '-', '-', '+', '-']
}

df = pd.DataFrame(data)


def entropy(target_class):
    total_count = len(target_class)
    positive_count = (target_class == '+').sum()
    negative_count = (target_class == '-').sum()
    positive_prob = positive_count / total_count
    negative_prob = negative_count / total_count

    if positive_prob == 0 or negative_prob == 0:
        return 0

    return -(positive_prob * log2(positive_prob) + negative_prob * log2(negative_prob))


def information_gain(df, attribute, split_point, target_class):
    subset1 = df[df[attribute] <= split_point][target_class]
    subset2 = df[df[attribute] > split_point][target_class]

    total_count = len(df)
    count1 = len(subset1)
    count2 = len(subset2)

    weighted_entropy = (count1 / total_count) * entropy(subset1) + \
        (count2 / total_count) * entropy(subset2)
    original_entropy = entropy(df[target_class])

    return original_entropy - weighted_entropy


# Calculate the possible split points for A3
sorted_A3 = sorted(df['A3'].unique())
```

```
split_points = [(sorted_A3[i] + sorted_A3[i + 1]) /
                2 for i in range(len(sorted_A3) - 1)]

# Calculate the information gain for each split point
information_gains = [information_gain(
    df, 'A3', split_point, 'Target Class') for split_point in split_points]

# Find the best split point with the highest information gain
best_split_point = split_points[np.argmax(information_gains)]


# Save the output as a CSV table
output_data = {
    'Split Points': split_points,
    'Information Gains': information_gains
}
output_df = pd.DataFrame(output_data)

# Create the output directory if it doesn't exist
output_dir = './docs/assignments/assignment 1/output/all_splits_information_gain'
os.makedirs(output_dir, exist_ok=True)

# Save the DataFrame to a CSV file
output_df.to_csv(os.path.join(
    output_dir, 'all_splits_information_gain.csv'), index=False)

print(f"Split points: {split_points}")
print(f"Information gains: {information_gains}")
print(f"Best split point: {best_split_point}")
```