ID3 code :-

```
import pandas as pd
data = {
'Outlook': ['Sunny', 'Sunny', 'Overcast', 'Rainy', ...],
'Temperature': ['Hot', 'Hot', 'Hot', 'Mild', ...],
'Humidity': ['High', 'High', 'High', 'High', 'Normal', ...],
'Windy': ['Weak', 'Strong', 'Weak', 'Weak', 'Strong', ...],
'PlayTennis': ['No', 'No', 'Yes', 'Yes', 'Yes', 'No', 'Yes', ...]
}
df = pd.DataFrame(data)

import math
def entropy(data):
    total = len(data)
    counts = data.value_counts()
    entropy_value = 0
    for count in counts:
        probability = count / total
        entropy_value -= probability * math.log2(prob
    return entropy_value

def info_gain(data, feature, target):
    total_entropy = entropy(data[target])
    feature_values = data[feature].unique()
    weighted_entropy = 0
    for value in feature_values:
        subset = data[data[feature] == value]
        weighted_entropy += (len(subset) / len(data)) *
                            entropy(subset[target])
    return total_entropy - weighted_entropy
```

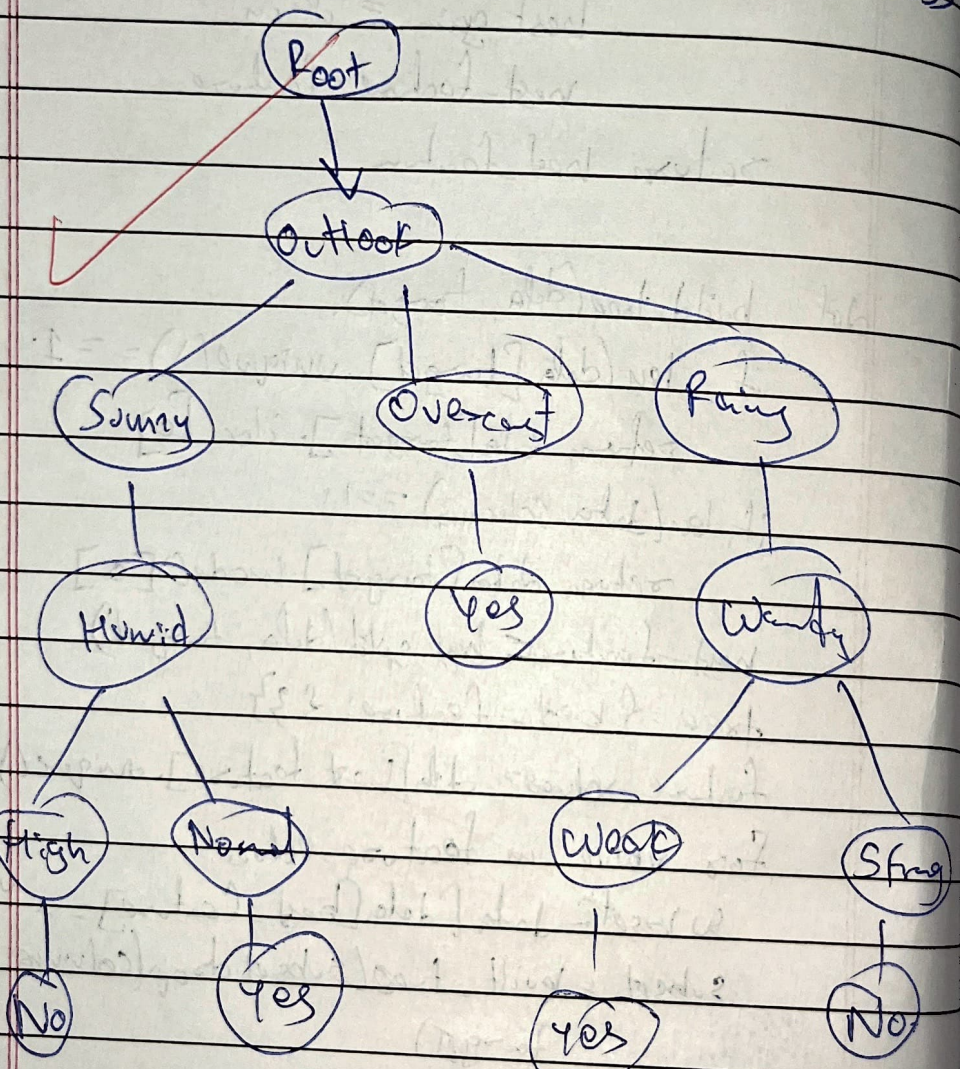
```
def best_split(data, target):
    features = data.drop(columns=[target]).columns
    best_feature = None
    best_gain = -1
    for feature in features:
        gain = information_gain(data, feature, target)
        if gain > best_gain:
            best_gain = gain
            best_feature = feature
    return best_feature

def build_tree(data, target):
    if len(data[target].unique()) == 1:
        return data[target].iloc[0]
    if len(data.columns) == 1:
        return data[target].mode()[0]
    best_feature = best_split(data, target)
    tree = {best_feature: {}}
    feature_values = data[best_feature].unique()
    for value in feature_values:
        subset = data[data[best_feature] == value]
        subset = build_tree(subset.drop(columns=[best_feature]),
                            target)
        tree[best_feature][value] = subset
    return tree

tree = build_tree(df, target='PlayTennis')
print(tree)
```

Output

{'Outlook': {'Sunny': {'Humidity': {'High': 'No', 'Normal': 'yes'}}, 'Overcast': 'yes', 'Rainy': {'Windy': {'weak': 'yes', 'strong': 'No'}}}}



California Housing Prices

① import pandas as pd
housing = pd.read_csv("housing.csv")
housing.head()

② housing.info()

③ housing["ocean_proximity"].value_counts()

④ housing.describe()

⑤ import matplotlib.pyplot as plt
housing["median_house_value"].hist(bins=50, figsize=(10,6))

Random splits ⑥
train_set, test_set = train_test_split(housing, test_size=0.2, random_state=42)

⑦ from sklearn.model_selection import train_test_split

X = housing.drop("median_house_value", axis=1)
y = housing["median_house_value"]
housing["income_cat"] = pd.cut(housing["median_house_value"],
bins=[0, 100000, 200000, 300000, np.inf],
labels=[1, 2, 3, 4, 5])

X_train, X_test, y_train, y_test = train_test_split
(X, y, test_size=0.2, random_state=42,
stratify = housing["income_cat"]).

(8)
```
train_set = X_train.copy()
train_set["median_house_value"] = y_train
train_set.plot(kind="scatter", x="longitude",
               y="latitude")
plt.legend()
```