In [ ]:
```
import pandas as pd
import numpy as np
import jax.numpy as jnp
from jax.nn import softmax, one_hot

df = pd.read_csv("dumbal_orderly.csv")
```

In [ ]:
```
def predict (params, x):
    w = params["w"]
    weighted = jnp.multiply(x, w)
    utilities = jnp.sum(weighted, axis = 2)

    print("utilites shape: ", utilities.shape)

    T = params["T"]
    prediction = softmax(T * utilities)
    #print(prediction)

    return prediction

def compute_accuracy (params, x, y):
    prediction = predict(params, x)
    return jnp.mean(jnp.argmax(prediction, axis = -1) == jnp.argmax(y, axis
```

In [ ]:
```
a_feats = np.expand_dims(df[["a_cost", "a_rate"]].values, axis=1)
b_feats = np.expand_dims(df[["b_cost", "b_rate"]].values, axis=1)
d_feats = np.expand_dims(df[["d_cost", "d_rate"]].values, axis=1)

X = np.concatenate([a_feats, b_feats, d_feats], axis=1)
X = (X - X.mean()) / X.std()

y = df.choice.values
y = one_hot(y, X.shape[1])
```

In [ ]:
```
w1 = 8.21636933989816e-10
w2 = 1 - w1

print("The AGM model weights are ", w1, ", ", w2)

params = {'w': np.array([w1, w2]), 'T': 0.43136}
# measure the accuracy of the model
print("\n The AGM model accuracy is: ", compute_accuracy(params, X, y))
```

```
The AGM model weights are  8.21636933989816e-10 ,  0.999999999178363
utilites shape:  (204542, 3)

 The AGM model accuracy is:  0.4674639
```