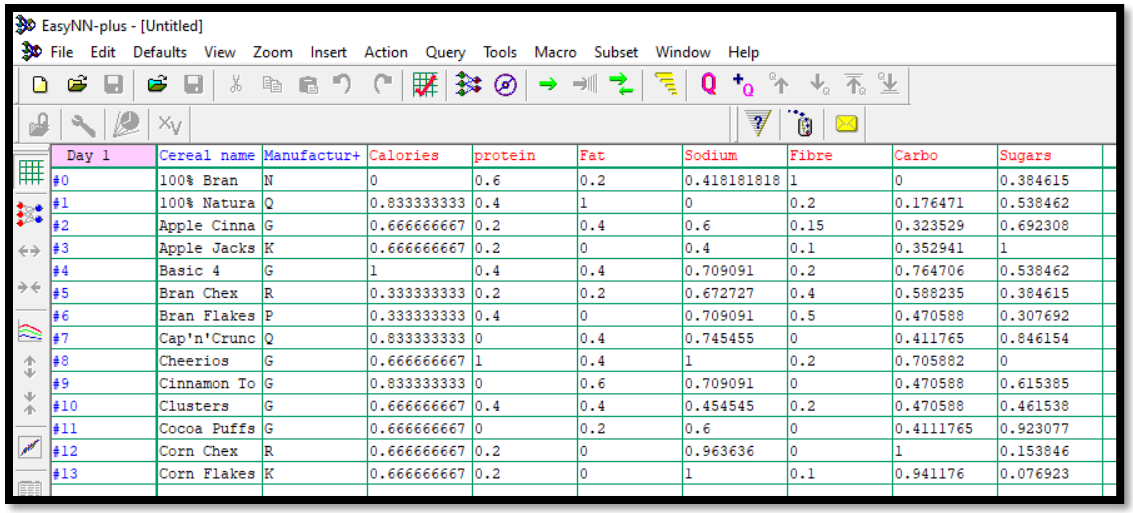
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.models import load_model

# Step 1: Load the dataset
# Assuming you have your dataset saved as a CSV file. Adjust the path accordingly.
# Example DataFrame based on your earlier dataset
data = {
    'Cereal_name': ['100% Bran', '100% Natural Bran', 'Apple Cinnamon Cheerios', 'Apple Jacks',
                    'Basic 4', 'Bran Chex', 'Bran Flakes', 'Cap\'n\'Crunch', 'Cheerios',
                    'Cinnamon Toast Crunch', 'Clusters', 'Cocoa Puffs', 'Corn Chex', 'Corn Flakes'],
    'Manufact': ['N', 'Q', 'G', 'K', 'G', 'R', 'P', 'Q', 'G', 'G', 'G', 'G', 'R', 'K'],
    'Calories': [0, 0.833333, 0.666667, 0.666667, 1, 0.333333, 0.333333, 0.833333, 0.666667, 0.833333, 0.666667, 0.666667, 0.666667, 0.666667],
    'Protein': [0.6, 0.4, 0.2, 0.2, 0.4, 0.2, 0.4, 0, 1, 0, 0.4, 0, 0.2, 0.2],
    'Fat': [0.2, 1, 0.4, 0, 0.4, 0.2, 0, 0.4, 0.2, 0.6, 0.4, 0.2, 0, 0],
    'Sodium': [0.418182, 0.054545, 0.327273, 0.2, 0.709091, 0.672727, 0.709091, 0.745455, 1, 0.709091, 0.454545, 0.327273, 0.963636, 1],
    'Fibre': [1, 0.2, 0.15, 0.1, 0.2, 0.4, 0.5, 0, 0.2, 0, 0.2, 0, 0, 0],
    'Carbo': [0, 0.176471, 0.323529, 0.352941, 0.764706, 0.588235, 0.470588, 0.411765, 0.705882, 0.470588, 0.470588, 0.411765, 0.941176, 0.941176],
    'Sugars': [0.384615, 0.538462, 0.692308, 0.923077, 0.538462, 0.384615, 0.307692, 0.846154, 0.076923, 0.615385, 0.538462, 0.692308, 0.153846, 0.076923] }

df = pd.DataFrame(data)
# Step 2: Preprocess the data
# Convert categorical 'Manufact' column to numerical using Label Encoding
label_encoder = LabelEncoder()
df['Manufact'] = label_encoder.fit_transform(df['Manufact'])

# Separate features and target (we'll predict the 'Sugars' column)
X = df.drop(columns=['Cereal_name', 'Sugars'])
y = df['Sugars']

# Step 3: Split data into training and test sets (80% training, 20% testing)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 4: Create the ANN model
model = Sequential()
model.add(Dense(units=64, activation='relu', input_dim=X_train.shape[1]))
model.add(Dense(units=32, activation='relu'))
model.add(Dense(units=1))  # For regression output

# Step 5: Compile the model
model.compile(optimizer='adam', loss='mean_squared_error', metrics=['mae'])


# Step 6: Train the model
history = model.fit(X_train, y_train, epochs=100, batch_size=10, validation_split=0.2)

# Step 7: Evaluate the model
loss, mae = model.evaluate(X_test, y_test)
print(f"Test Mean Absolute Error: {mae}")

# Step 8: Save the trained model
model.save('cereal_sugar_prediction_model.h5')
print("Model saved as 'cereal_sugar_prediction_model.h5'")

# Step 9: Load the model (optional)
loaded_model = load_model('cereal_sugar_prediction_model.h5')

# Step 10: Query the network (predicting sugars for a new cereal)
# Create a sample input (normalized values)
sample_input = np.array([[0.666667, 1, 0.2, 0.7, 0.4, 0.7]])  # Replace with actual sample values

# Predict the sugar content using the trained model
predicted_sugars = loaded_model.predict(sample_input)
print(f"Predicted Sugars for sample input: {predicted_sugars[0][0]}")
```



Day 1	Cereal name	Manufactur+	Calories	protein	Fat	Sodium	Fibre	Carbo	Sugars
#0	100% Bran	N	0	0.6	0.2	0.418181818	1	0	0.384615
#1	100% Natura	Q	0.833333333	0.4	1	0	0.2	0.176471	0.538462
#2	Apple Cinna	G	0.666666667	0.2	0.4	0.6	0.15	0.323529	0.692308
#3	Apple Jacks	K	0.666666667	0.2	0	0.4	0.1	0.352941	1
#4	Basic 4	G	1	0.4	0.4	0.709091	0.2	0.764706	0.538462
#5	Bran Chex	R	0.333333333	0.2	0.2	0.672727	0.4	0.588235	0.384615
#6	Bran Flakes	P	0.333333333	0.4	0	0.709091	0.5	0.470588	0.307692
#7	Cap'n'Crunc	Q	0.833333333	0	0.4	0.745455	0	0.411765	0.846154
#8	Cheerios	G	0.666666667	1	0.4	1	0.2	0.705882	0
#9	Cinnamon To	G	0.833333333	0	0.6	0.709091	0	0.470588	0.615385
#10	Clusters	G	0.666666667	0.4	0.4	0.454545	0.2	0.470588	0.461538
#11	Cocoa Puffs	G	0.666666667	0	0.2	0.6	0	0.4111765	0.923077
#12	Corn Chex	R	0.666666667	0.2	0	0.963636	0	1	0.153846
#13	Corn Flakes	K	0.666666667	0.2	0	1	0.1	0.941176	0.076923

	Cereal nar	Manufacti	Calories	protein	Fat	Sodium	Fibre	Carbo	Sugars
1									
2	100% Bran	N	70	4	1	130	10	5	6
3	100% Natu	Q	120	3	5	15	2	8	8
4	Apple Cin	G	110	2	2	180	1.5	10.5	10
5	Apple Jacl	K	110	2	0	125	1	11	14
6	Basic 4	G	130	3	2	210	2	18	8
7	Bran Chex	R	90	2	1	200	4	15	6
8	Bran Flake	P	90	3	0	210	5	13	5
9	Cap'n'Crur	Q	120	1	2	220	0	12	12
10	Cheerios	G	110	6	2	290	2	17	1
11	Cinnamon	G	120	1	3	210	0	13	9
12	Clusters	G	110	3	2	140	2	13	7
13	Cocoa Puf	G	110	1	1	180	0	12	13
14	Corn Chex	R	110	2	0	280	0	22	3
15	Corn Flake	K	100	2	0	290	1	21	2
16									