1.

1. The reason it is better to binarize features such as age is that linearizing the data into two fields introduces a bias about the data that may be incorrect. It is not likely the relationship between age and income is so easily captured by a linear mapping.

2. You get 230 binarized features from the 9 fields (2 numerical, 7 categorical) as well as 1 additional field from the bias dimension for a total of **231** features.

2.

1.

```
epoch 1 updates 1257 (25.14% ) dev_error 22.4% (+:31.6% )
epoch 2 updates 1221 (24.42% ) dev_error 20.8% (+:30.6% )
epoch 3 updates 1177 (23.54% ) dev_error 18.0% (+:23.8% )
epoch 4 updates 1170 (23.4% ) dev_error 19.7% (+:15.1% )
epoch 5 updates 1172 (23.44% ) dev_error 18.7% (+:21.3% )
```

My best error rate was 18% was on epoch 3.

2.

```
epoch 1 updates 1257 (25.14% ) dev_error 15.0% (+:18.6% )
epoch 2 updates 1221 (24.42% ) dev_error 15.1% (+:19.3% )
epoch 3 updates 1177 (23.54% ) dev_error 14.8% (+:20.0% )
epoch 4 updates 1170 (23.4% ) dev_error 14.7% (+:19.3% )
epoch 5 updates 1172 (23.44% ) dev_error 14.8% (+:20.0% )
```

My best error on dev rate is 14.7% on my 4th epoch

3. Comparing the results between the non averaged and the averaged clearly shows that the averaged not only gets better error, but also does so faster (and I didn't even use the clever speed up trick).
4. The results make a lot of sense. In the positive group we are seeing doctorate degrees and professional school and in negative we are seeing farm-fishing (which is likely a low paying job) and people who have only completed education through middle school.

Positive
(2, Doctorate) 8.949
(3, Married-civ-spouse) 7.5747
(2, Prof-school) 6.8508
(8, Iran) 6.0962
(7, 65) 5.1061
Negative
(0, 28) -7.50345
(2, 7th-8th) -7.08815
(2, bias) -7.31665

(4, Farming-fishing) -7.03415
(0, 26) -6.2085

5. The weights are negative despite men having a higher percentage of positive income because the data itself is biased towards income <50k to begin with. Both men and women are more likely to make less than 50k despite men being more likely than women.

(6, Male) -3.99015
(6, Female) -3.3265

6. The bias weight is -7.31665 . For the same reason as above, we expect that the bias would be negative as the data itself is biased towards negative outcomes.

7. The update % is not equivalent to the training error as the training error is the final outcome after performing an epoch (or multiple epochs) while the update is only telling you how many times it updated whilst still training.

3.

1. The perceptron has several major advantages over the k-NN:

- It has significantly faster evaluation times than the k-NN which has to perform distance calculations and sorting/partitioning to lookup values. This allows the perceptron to be used on data with significantly more features and larger datasets.
- The perceptron is actually able to learn feature significance (to an extent) by updating weights. The k-NN does not actually perform any learning.
- For the reason above, the accuracy of the perceptron is notably better than the k-NN.

2. I did a very simple comparison of training times (treating the feature mapping process as being the "training" for the k-NN) as evaluation times, as well as the best error that I achieved for both approaches:

For the **k-NN:**
Time to create mapping is 0.036032676696777344s
Best error rate for k-NN is 17.34% taking 7.337460279464722s

For the **Averaged Perceptron:**
Time to create mapping and train is 0.11910796165466309s
Best error rate for k-NN is 14.8% taking 0.0s

As can be seen by looking at the results, the training time for the perceptron is slightly higher, however the evaluation time is orders of magnitude faster than the k-NN. Furthermore, the best error rate using a fairly basic averaged perceptron is already decently better than the results from the k-NN.

4.

1. Both the basic (vanilla) and averaged perceptrons performance dropped significantly when ordering the data such that either the positives or negatives came first. This appears to have been especially problematic for the basic perceptron, especially when encountering all positive data points at the start.

Averaged Perceptron: Positives first

```
epoch 1 dev_error 15.0% (+:18.6% )  → dev_error 23.6% (+:0.2% )
epoch 2 dev_error 15.1% (+:19.3% )  → dev_error 23.3% (+:0.3% )
epoch 3 dev_error 14.8% (+:20.0% )  → dev_error 23.0% (+:0.6% )
epoch 4 dev_error 14.7% (+:19.3% )  → dev_error 23.0% (+:1.0% )
epoch 5 dev_error 14.8% (+:20.0% )  → dev_error 22.7% (+:1.5% )
```

Averaged Perceptron: Negatives first

```
epoch 1 dev_error 15.0% (+:18.6% )  → dev_error 26.1% (+:2.5% )
epoch 2 dev_error 15.1% (+:19.3% )  → dev_error 23.8% (+:0.4% )
epoch 3 dev_error 14.8% (+:20.0% )  → dev_error 23.8% (+:0.4% )
epoch 4 dev_error 14.7% (+:19.3% )  → dev_error 23.6% (+:0.2% )
epoch 5 dev_error 14.8% (+:20.0% )  → dev_error 23.8% (+:0.4% )
```

Basic Perceptron: Positives first

```
epoch 1 dev_error 22.4% (+:31.6% )  → dev_error 23.6% (+:0.0% )
epoch 2 dev_error 20.8% (+:30.6% )  → dev_error 23.6% (+:0.0% )
epoch 3 dev_error 18.0% (+:23.8% )  → dev_error 23.5% (+:0.1% )
epoch 4 dev_error 19.7% (+:15.1% )  → dev_error 23.6% (+:0.0% )
epoch 5 dev_error 18.7% (+:21.3% )  → dev_error 23.6% (+:0.0% )
```

Basic Perceptron: Positives first

```
epoch 1 dev_error 22.4% (+:31.6% )  → dev_error 76.4% (+:100.0% )
epoch 2 dev_error 20.8% (+:30.6% )  → dev_error 76.4% (+:100.0% )
epoch 3 dev_error 18.0% (+:23.8% )  → dev_error 76.4% (+:100.0% )
epoch 4 dev_error 19.7% (+:15.1% )  → dev_error 76.4% (+:100.0% )
epoch 5 dev_error 18.7% (+:21.3% )  → dev_error 76.4% (+:100.0% )
```

2. Feature engineering:

a) Adding the numerical fields in addition to the binarized age and hour fields did not improve the results of either perceptron.

Basic Perceptron

```
epoch 1 updates 1222 (24.44% ) dev_error 23.2% (+:27.2% )
epoch 2 updates 1134 (22.68% ) dev_error 18.9% (+:19.3% )
epoch 3 updates 1127 (22.54% ) dev_error 18.9% (+:23.5% )
epoch 4 updates 1137 (22.74% ) dev_error 19.9% (+:24.1% )
epoch 5 updates 1130 (22.6% ) dev_error 20.9% (+:21.5% )
```

Averaged Perceptron

```
epoch 1 updates 1222 (24.44% ) dev_error 15.7% (+:18.3% )
epoch 2 updates 1134 (22.68% ) dev_error 15.6% (+:19.6% )
epoch 3 updates 1127 (22.54% ) dev_error 14.7% (+:20.1% )
epoch 4 updates 1137 (22.74% ) dev_error 15.1% (+:19.9% )
epoch 5 updates 1130 (22.6% ) dev_error 15.2% (+:20.2% )
```

b) …
c) …
d) …

3. …

Debriefing:

1. I spent maybe 8 hours on this assignment (however I didn’t exactly finish).
2. I would say it was not very hard thus far, I just was unable to spend the time due to research commitments.
3. I worked on it alone.
4. I feel like I am 80% there.
5. My apologies for the late submission, I appreciate the one late assignment policy, it allowed me to get what I needed done with research and still have enough time to get a reasonable amount of this assignment done.