**Question 1:**

The relationship between the weights of the logistic regression model for features n and n+1 depends on various factors, including the correlation between the duplicated feature and the impact of regularization.

1. **No Regularization:**
   * If there is no regularization (e.g., L1 or L2 regularization) applied during training, and the duplicated features *n+1* is identical to features *n*, the weights associated with these features (*wn* ​ and *w​​new,n+1*) are likely to be very close or equal.
2. **L1 Regularization:**
   * L1 regularization tends to encourage sparsity in the weights. If feature *n* and *n+1* are highly correlated, L1 regularization might choose to assign non-zero weights to only one of them. In this case, the weights *wn* ​ and *w​​new,n+1* could be non-zero, but the magnitudes may not be equal.
3. **L2 Regularization:**
   * L2 regularization penalizes large weights. If features *n* and *n+1* are highly correlated, the model may distribute the weight between them to avoid large individual weights. In this case, *wn* ​ and *w​​new,n+1*​​ could be non-zero, but the magnitudes may not be equal.
4. **Interaction Effects:**
   * If there are interaction effects between features, duplicating a feature may change the weight assigned to the original feature. The new weights *wn* ​ and *w​​new,n+1*​​ could be influenced by the interaction between *n* and *n+1*, and they may not be simply duplicated.

In summary, the relationship between *wn* ​ and *w​​new,n+1*​​ depends on the model's regularization, the correlation between the duplicated features, and potential interactions between features. Without more information about the specific dataset and the training process, it's challenging to provide a precise relationship between these weights.

**Question 2**

To determine the statistical significance of the click-through rates (CTR) for the different email templates, you can perform a hypothesis test. Typically, a chi-squared test or z-test for proportions is used for this type of comparison. In this case, given that you want to achieve 95% confidence in your conclusion, it's important to consider the confidence intervals for the CTR estimates.

Let's examine the options:

1. **"We have too little data to conclude that A is better or worse than any other template with 95% confidence."**
   * This statement is not accurate because we can compare the templates based on the provided CTRs and calculate confidence intervals.
2. **"E is better than A with over 95% confidence, B is worse than A with over 95% confidence. You need to run the test for longer to tell where C and D compare to A with 95% confidence."**
   * This statement is premature. While it's true that the CTR for E is higher than A, and the CTR for B is lower than A, we need to perform statistical tests to determine whether these differences are statistically significant. Additionally, we can already compare C and D based on the provided CTRs.
3. **"Both D and E are better than A with 95% confidence. Both B and C are worse than A with over 95% confidence."**
   * This statement is also premature. While it's true that D and E have higher CTRs than A, and B and C have lower CTRs than A, statistical tests are needed to confirm whether these differences are significant.

To make a more accurate conclusion, you would typically calculate confidence intervals for each CTR and perform hypothesis tests to compare them. Based on the results of these tests, you can determine whether there is a statistically significant difference in CTR between the templates. Without performing these tests, it is not possible to confidently state which template is better or worse with 95% confidence.

**Question 3**

In the context of logistic regression, especially with sparse features, the computational cost of each gradient descent iteration can be significantly reduced by taking advantage of the sparse nature of the data. Many modern machine learning libraries, such as scikit-learn, TensorFlow, and PyTorch, are optimized to handle sparse data efficiently.

Let's break down the computational cost of each gradient descent iteration:

1. **Forward Pass:**
   * Computing the linear combination of weights and features and applying the sigmoid function for each training example. The cost is proportional to the number of non-zero entries in the sparse feature vectors.
   * Cost: *O*(*k*⋅*m*), where *k* is the average number of non-zero entries per example, and *m* is the number of training examples.
2. **Backward Pass (Gradient Calculation):**
   * Computing the gradient of the logistic loss with respect to the weights. Again, the cost is proportional to the number of non-zero entries in the sparse feature vectors.
   * Cost: *O*(*k*⋅*m*).
3. **Weight Update:**
   * Updating the weights based on the computed gradients.
   * Cost: *O*(*n*), where *n* is the total number of features.

Considering both the forward and backward passes, the overall cost of each gradient descent iteration for logistic regression with sparse features is approximately *O*(*k*⋅*m*+*n*). Importantly, the *k*⋅*m* term dominates the cost because *k* is typically much smaller than *n* in the case of sparse data.

This computational cost is significantly more efficient than the cost for dense data, where the dominant term in the complexity is often *O*(*n*⋅*m*). Taking advantage of the sparsity in feature vectors allows for more scalable logistic regression training, especially when dealing with high-dimensional sparse data.

**Question 4**

Let's analyze each approach in terms of the potential impact on the accuracy of classifier V2:

1. **Run V1 Classifier on 1 Million Random Stories:**
   * **Potential Outcome:** This approach aims to select examples where V1's output is closest to the decision boundary. It may identify challenging examples that are difficult for the V1 classifier, potentially helping V2 generalize better on ambiguous cases.
   * **Potential Impact on V2 Accuracy:** Positive impact. The selected examples could provide valuable insights into cases where the model struggles, potentially leading to improved performance on similar examples in the future.
2. **Get 10k Random Labeled Stories:**
   * **Potential Outcome:** Randomly selecting labeled stories from the 1000 news sources may provide a diverse set of examples. However, it does not specifically focus on cases where the V1 classifier struggled or where the decision boundary is ambiguous.
   * **Potential Impact on V2 Accuracy:** Neutral. While it contributes to a diverse training set, it may not specifically address the weaknesses of the V1 classifier.
3. **Pick a Random Sample of 1 Million Stories, Selecting Wrong and Farthest from Decision Boundary:**
   * **Potential Outcome:** This approach focuses on selecting examples where the V1 classifier was both wrong and farthest away from the decision boundary. This could highlight cases where the V1 model was confidently wrong, potentially indicating areas for improvement.
   * **Potential Impact on V2 Accuracy:** Positive impact. By focusing on cases where V1 was confidently wrong, this approach may address specific weaknesses and contribute to improving accuracy.

**Ranking Based on Potential Impact:**

1. **Approach 3:** This is likely to have the most positive impact on V2 accuracy, as it specifically targets cases where the V1 classifier was wrong and far from the decision boundary.
2. **Approach 1:** This approach may also contribute positively by providing challenging examples close to the decision boundary, helping V2 generalize better.
3. **Approach 2:** While it adds diversity to the training set, it may not specifically address the weaknesses of the V1 classifier.

In summary, focusing on examples where the V1 classifier was confidently wrong and far from the decision boundary (Approach 3) is likely to have the most significant positive impact on the accuracy of classifier V2.

**Question 5:**

Let's calculate the estimates for the probability *p* using the three methods described:

**1. Maximum Likelihood Estimate (MLE):**

The MLE for *p* is obtained by maximizing the likelihood function, which is the probability of observing the data given the parameter *p*.

MLE: *p* ^MLE=MLE: *p*^​MLE​=*k/n*

**2. Bayesian Estimate:**

Assuming a uniform prior on *p* from 0 to 1, the posterior distribution is a Beta distribution. The expected value (mean) of the posterior distribution is the Bayesian estimate for *p*.

Bayesian Estimate: *p*^Bayesian= *k* +1 / *n* +2

**3. Maximum a Posteriori (MAP) Estimate:**

Assuming a uniform prior on *p* from 0 to 1, the posterior distribution is again a Beta distribution. The mode of the Beta distribution is given by (*k*,*n*−*k*), so the MAP estimate is the mode.

MAP Estimate: *p*^​MAP​=*nk*​

**Summary:**

1. **Maximum Likelihood Estimate (MLE):**

*p*^​MLE​=*nk*​

1. **Bayesian Estimate:**

*p*^​Bayesian​=*n*+2*k*+1​

1. **Maximum a Posteriori (MAP) Estimate:**

*p*^​MAP​=*nk*​

It should be noted that, the MAP estimate and the MLE are the same because the uniform prior used in the Bayesian approach does not influence the mode of the posterior distribution. The Bayesian estimate introduces a slight regularization by adding 1 to both the number of heads and the total number of tosses.