1. Short questions.

- Explain why replacing the linear dot product with a non-linear kernel function (e.g., the quadratic kernel) in SVM allows it to learn a non-linear decision boundary.

Using a non-linear kernel function is equivalent to transforming the data into a nonlinear space then taking dot product in the transformed space. A linear boundary learned in this transformed space is nonlinear in the original space.

- Consider the following strategy for selecting $k$ for $k$-means. Run $k$-means using $k=2, 3, \cdots, k_{\text{max}}$, choose the $k$ value that gives the lowest Sum of Squared Error (SSE). What is wrong with this strategy? This strategy will always select $k_{\text{max}}$ because SSE is expected to decrease as we increase $k$.

- In perceptron, let $w_t$ denote the current weight vector and it misclassifies $(x_t, y_t)$. What is the update rule for creating the new weight $w_{t+1}$? Explain why this update rule can correct for the mistake on $(x_t, y_t)$.

The update rule is $w' \leftarrow w + y_t x_t$. $y_t w^T x_t = y_t w^T + y_t^2 x_t^T x_t \geq y_t w^T$, which corrects $w$ toward making $(x_t, y_t)$ correct.
Please indicate for each action below whether it increases, decreases, or does not impact overfitting.

* For KNN, change $k$ from 5 to 1
  Increasing overfitting
* Change parameter $c$ for soft-margin SVM from 1000 to 0.1
  Decreasing overfitting
* Change from a linear Kernel to a 3-rd order polynomial kernel in SVM.
  Increasing overfitting

2. True or false questions.

  – (True or False) Logistic regression learns a non-linear decision boundary because it assumes that $p(y|X) = \frac{1}{1+e^{-w \cdot X}}$, which is a nonlinear function of $X$.
    F
  – (True or False) When learning a linear decision boundary with the perceptron algorithm, it is guaranteed to converge within a finite number of steps.
    F (T only if the data is linear separable.)
  – (True or False) When applying $k$-means algorithm to cluster a given set of data, it is guaranteed to converge within a finite number of steps.
    T
  – (True or False) If random variable A and B are independent from one another, they must be conditionally independent given C, where C can be any arbitrary random variable.
    F
We want to predict the stock price of an S&P 500 company. According to the recent KD Nuggets news, the perfect set of variables are: Butter production in Bangladesh (B), U.S. and Bangladesh Cheese production (C), and the Sheep population in U.S. and Bangladesh (S). We collected the following training data.

<table>
<thead>
<tr>
<th>B</th>
<th>C</th>
<th>S</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>high</td>
<td>low</td>
<td>low</td>
<td>up</td>
</tr>
<tr>
<td>high</td>
<td>high</td>
<td>low</td>
<td>up</td>
</tr>
<tr>
<td>high</td>
<td>low</td>
<td>high</td>
<td>down</td>
</tr>
<tr>
<td>low</td>
<td>low</td>
<td>low</td>
<td>down</td>
</tr>
<tr>
<td>low</td>
<td>high</td>
<td>high</td>
<td>up</td>
</tr>
</tbody>
</table>

3 Please use the Naive Bayes classifier to make prediction for (B=low, C=low, S=high). Please show your steps, which should clearly state all the distributions you use for your Naive Bayes classifier. A simple answer of predicting up or down will not be given points.

\[
P(\text{up}) = 0.6; \quad P(\text{down}) = 0.4
\]

\[
P(B = \text{low}|\text{up}) = 1/3
\]

\[
P(B = \text{low}|\text{down}) = 1/2
\]

\[
p(C = \text{low}|\text{up}) = 1/3
\]

\[
p(C = \text{low}|\text{down}) = 1
\]

\[
p(S = \text{low}|\text{up}) = 1/3
\]

\[
P(S = \text{high}|\text{down}) = 1/2
\]

\[
P(\text{up}|B = \text{low}, C = \text{low}, S = \text{high}) \propto P(B = \text{low}|\text{up})P(C = \text{low}|\text{up})P(S = \text{high}|\text{up})P(\text{up}) = 1/3 \times 1/3 \times 1/3 \times 0.6
\]

\[
P(\text{down}|B = \text{low}, C = \text{low}, S = \text{high}) \propto P(B = \text{low}|\text{down})P(C = \text{low}|\text{down})P(S = \text{high}|\text{down})P(\text{down}) = 1/2 \times 1 \times 1/2 \times 0.4
\]

\[
P(\text{down}|B = \text{low}, C = \text{low}, S = \text{high}) > P(\text{up}|B = \text{low}, C = \text{low}, S = \text{high}). \text{ Thus predict down.}
\]

– Suppose now we learned that the price always goes down when B = S and goes up when B ≠ S. Does the Naive Bayes assumption hold and why?

No. The features B and S are not conditional independent of each other given price.

– Build a depth-one decision tree (a.k.a a decision stump) using error-rate as the uncertainty measure. Please show your steps and clearly mark out the class labels for the leaf nodes of your decision tree.

Feature B and S will each create 2 errors, whereas feature C will only introduce 1 error. Thus the decision stump will test on feature C, and predict down if C=low, and predict up otherwise.

– Can the top-down greedy algorithm learn a decision tree to correctly represent the class concept described in (b)? Why?

No, because the greedy top down algorithm will not likely select either B or S as the root test since individually each of these features is not indicative of the class label. Without some level of look ahead, the greedy algorithm will likely be stuck with a suboptimal tree.