Question 1 (Probability and Bayes Theorem)
Consider two events. Under event $E_1$, a student prefers to go to the beach for a vacation. Under event $E_2$, a student prefers to go to the mountains for a vacation. The probabilities of these events satisfy $\Pr(E_1) = 0.4$ and $\Pr(E_2) = 0.7$. Moreover, the probability that a student prefers neither the beach nor the mountains is $\Pr((E_1)^C, (E_2)^C) = 0.2$. Please compute the following probabilities. (Note that $(\cdot)^C$ denotes the complement of a set or event.)

(a) $\Pr((E_1)^C)$.

(b) $\Pr((E_2)^C)$.

c) $\Pr(E_1 \cup E_2)$
   \textbf{Solution:} This is equivalent to $1 - \Pr((E_1)^C, (E_2)^C)$, which is 0.8.

(d) $\Pr(E_1, E_2)$.
   \textbf{Solution:} This is equivalent to $\Pr(E_1) + \Pr(E_2) - \Pr(E_1 \cup E_2)$, which is 0.3.

(e) $\Pr((E_1)^C, E_2)$.
   \textbf{Solution:} This is equivalent to $\Pr(E_2/(E_1 \cap E_2))$, which is 0.4.

(f) $\Pr(E_1, (E_2)^C)$.
   \textbf{Solution:} This is equivalent to $\Pr(E_1/(E_1 \cap E_2))$, which is 0.1.
Question 2 (Model Complexity)

Consider a model for text, where the alphabet is comprised of $C$ characters, and each character is predicted by the previous 2 characters. That is, $X_n$ has a probability mass function, $P(X_n|X_{n-1}, X_{n-2})$. We want to learn these probabilities from the data. Please describe the model complexity for length-$N$ input strings as a function of $C$ and $N$.

Solution: For each pair of previous characters, $X_{n-2}$ and $X_{n-1}$, we have $C$ probabilities for the next character. However, because these probabilities sum to 1, there are $C - 1$ degrees of freedom, hence $C - 1$ parameters. For each of these $C - 1$ continuous valued parameters, the complexity is $\approx 0.5 \log_2(N)$ bits, hence $\approx \frac{C-1}{2} \log_2(N)$ bits per pair of previous characters.

Finally, there are $C^2$ pairs of previous characters, for each we pay a complexity penalty of $\approx \frac{C-1}{2} \log_2(N)$, and in total the model complexity is $\approx \frac{C^2(C-1)}{2} \log_2(N)$. (Note that each of these $C^2$ pairs appears roughly $N/C^2$ times, on average. Therefore, it might be more precise to use $N/C^2$ within the log. These sorts of higher order terms are unimportant when $N$ grows while $C$ is kept constant.)
Question 3 (Curve fitting.)
Consider a sequence of \( N \) measurements generated as follows,
\[
y = \sin(x) + \mathcal{N}(0, \sigma^2),
\]
where the noise is Gaussian with zero mean and variance \( \sigma^2 = 1 \). The inputs \( x \) and outputs \( y \) are each of length \( N \). Our goal in this question is to estimate the order of a reasonable Taylor approximation for the function, \( \sin(x) \).

For three signals of different lengths (\( N_1 = 10^2, N_2 = 10^4, N_3 = 10^6 \)) and several model orders, we fit coefficients to a polynomial approximation,
\[
y(x, (a_i)) = \sum_{i=0}^{\text{model order}} a_i x^i.
\]

Below, we list for several model orders the corresponding total squared error (TSE) for the three signal lengths (\( N_1, N_2, \) and \( N_3 \)),
\[
\text{TSE} = \sum_{n=1}^{N} (y_n - y(x_n, (a_i)))^2,
\]
where for each model order and signal length we optimized the \( (a_i) \) coefficients.

<table>
<thead>
<tr>
<th>Order</th>
<th>TSE ((N_1))</th>
<th>TSE ((N_2))</th>
<th>TSE ((N_3))</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>97.94</td>
<td>14205.78</td>
<td>1430238.64</td>
</tr>
<tr>
<td>1</td>
<td>81.73</td>
<td>10565.03</td>
<td>1062357.60</td>
</tr>
<tr>
<td>2</td>
<td>79.79</td>
<td>10564.51</td>
<td>1062351.54</td>
</tr>
<tr>
<td>3</td>
<td>74.77</td>
<td>10004.89</td>
<td>1000698.87</td>
</tr>
<tr>
<td>4</td>
<td>74.65</td>
<td>10004.34</td>
<td>1000697.44</td>
</tr>
<tr>
<td>5</td>
<td>73.97</td>
<td>9984.02</td>
<td>997574.81</td>
</tr>
<tr>
<td>6</td>
<td>73.47</td>
<td>9983.74</td>
<td>997518.10</td>
</tr>
<tr>
<td>7</td>
<td>73.10</td>
<td>9983.73</td>
<td>997514.20</td>
</tr>
<tr>
<td>8</td>
<td>71.49</td>
<td>9981.06</td>
<td>997514.20</td>
</tr>
<tr>
<td>9</td>
<td>71.44</td>
<td>9979.12</td>
<td>997506.23</td>
</tr>
</tbody>
</table>

For each signal length (\( N_1 = 10^2, N_2 = 10^4, N_3 = 10^6 \)), which model order seems best? Are your answers identical, different? What might that imply about the amount of data \( (N) \) needed to learn?

To solve this, we suggest that you minimize the length of a two part code, where the length of describing the data given the polynomial model obeys
\[
\text{len(data|model)} = -\sum_{n=1}^{N} \log_2(f(\text{unexplained}_n))
\]
\[
= -\sum_{n=1}^{N} \log_2 \left( \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{(y_n - y(x_n, (a_i)))^2}{2\sigma^2} \right\} \right)
\]
\[
= \frac{N}{2} \log_2(2\pi\sigma^2) + \log_2(e) \sum_{n=1}^{N} \frac{(y_n - y(x_n, (a_i)))^2}{2\sigma^2}
\]
\[
= \frac{N}{2} \log_2(2\pi\sigma^2) + \frac{\log_2(e)}{2\sigma^2} \text{TSE},
\]
and \( \frac{\log_2(e)}{2\sigma^2} \approx 0.72 \). (You may assume that each parameter requires \( \text{par}(N) = 2 + \frac{1}{2} \log_2(N) \) bits, and so \( \text{par}(N_1) \approx 5.3, \text{par}(N_2) \approx 8.6, \) and \( \text{par}(N_3) \approx 12.0 \) bits.) Make sure to justify your answer.
Solution: The total coding length is roughly

\[ \text{len}_1 + \text{len}_2 = \#\text{params} \cdot \text{par}(N) + \frac{N}{2} \log_2(2\pi\sigma^2) + \frac{\log_2(e)}{2\sigma^2} \text{TSE}, \]

where \( \frac{\log_2(e)}{2\sigma^2} \approx 0.72 \). Incrementing the model order makes sense if the reduction in TSE in Part 2 (multiplied by 0.72) exceeds the extra \( \text{par}(N) \) bits in Part 1.

For \( N_1 = 10^2 \) with \( \text{par}(N_1) = 5.3 \), moving from model order 0 (a constant approximation) to 1 (linear approximation) reduces the TSE by \( 97.94 - 81.73 = 16.21 \), hence the reduction in coding length is \( 16.21 \cdot 0.72 \approx 11.7 \) bits, which exceeds 5.3. However, all further increases in the model order are not helpful. Model order 1 is optimal.

For \( N_2 = 10^4 \) with \( \text{par}(N_2) = 8.6 \), moving from model order 4 to 5 reduces TSE by \( 10004.84 - 9984.02 \), and multiplying by 0.72 saves over 14 bits, which exceeds 8.6. However, model order 3 offers a TSE almost as good as model order 4, and shifting from 3 to 5 requires encoding 2 parameters, which requires \( 2 \cdot \text{par}(N_2) = 17.2 \), and model order 3 is optimal.

For \( N_3 = 10^6 \) with \( \text{par}(N_3) = 12.0 \) bits, moving from model order 6 to model order 7 reduces TSE by \( 997574.29 - 997518.10 = 56.19 \) bits, and model order 7 is optimal.