1. i. False, but its output is quantitative.
   ii. True, $\Phi$ is monotonically increasing and $\beta^T X$ is a linear function of $\beta$. This means that the decision boundary $\Phi(\beta^T x) = 0.2 \iff \beta^T x = \Phi^{-1}(0.2)$ is linear.
   iii. True, this is what boosting does for example.
   iv. True, a binary splitting cannot produce open corners like the lower right corner of the upper right rectangle.
   v. False, the term $\beta_0 \beta_1$ is not linear in the parameters.
   vi. False, $k$-NN (as any other method) does not overfit by adding more data. Adding more data reduces the error in average, when keeping $k$ constant.
   vii. False, least squares is a solution to maximum likelihood in some special cases.
   viii. True.
   ix. True, AdaBoost can use any base classifier.
   x. True, since the $k$-NN classifier is based on computing distances between inputs, and these distances are sensitive to the scaling of the input variables.

2. (a) Linear regression since this is a regression problem and not a classification problem (the output, price, is quantitative).
   (b) • id - neither. The price does not depend on the particular id of the wine.
   • grape - input. The price could depend on the grape used.
   • alcohol - input. Unknown if necessary but straightforward to implement and try out.
- year - input. The price is probably dependent on the year of the wine.
- region - neither or input. The region of the wine could be a useful input for predicting the price. However, since this variable has so many categories, it could be difficult to extract any useful information from it without using a very large training data set. One could consider some kind of partitioning as mentioned in the d) part, but with a better heuristic.
- proline - input. Unknown if necessary but straightforward to implement and try out.
- timestamp - neither. The price does not depend on when the wine was added to the database.
- price - output.

(c) grape - qualitative. The different grapes does not have any apparent order
- alcohol - quantitative. Is of ordinal nature
- year - quantitative or qualitative. The year could be viewed as having an ordinal nature (“the older the better”), however it could also be argued that there are certain years that are particularly good for producing wines (without any ordinal nature) and from this viewpoint the variable is qualitative.
- region - qualitative. The different regions does not have any apparent order.
- proline - quantitative. Is of ordinal nature.
- price - quantitative. As argued in a).

(d) It is intractable to enumerate all unique subsets \( I \subset \mathbb{R} \) since there are \( 2^{78} - 2 \) proper subsets (\(-2\) coming from the exclusion of \( I = \emptyset \) and \( I = \mathbb{R} \)). Even if we could process the subsets in 1GHz it would take 10 million years.

3. (a) The LDA classifier is linear and will therefore separate the two classes with a straight line in 2D. Looking at the scatter plots in Figure 2 it is clear that dataset i and iii can be separated using a straight line, but dataset ii can not. The QDA classifier is non-linear and will hence result in a non-linear (quadratic) decision boundary. Looking at the scatter plots in Figure 2 it is clear that in all three plots the two classes can be separated by a quadratic curve.
(b) Both LDA and QDA assume the inputs corresponding to each class have a Gaussian distribution with some mean value $\mu_k$ for each class $k$. The difference between LDA and QDA is that LDA assumes that the covariance matrix $\Sigma$ is the same for all classes $k$ whereas QDA assumes that each class $k$ has a unique covariance matrix $\Sigma_k$.

Looking at dataset $i$ it is clear that there does not seem to be any correlation (pos or neg) in any of the classes and the spread seems to be the same. Hence the assumptions of LDA (same covariance matrix for both classes) are fulfilled by dataset $i$. (Since QDA is a generalization of LDA, we could also argue that dataset $i$ satisfies the assumptions of QDA as well.)

Looking at dataset $iii$ it is clear that there seems to be correlation in both classes. Since the correlation is positive for $Y = 0$ and negative for $Y = 1$ the two classes have different covariance matrices. Hence the assumptions of QDA (but not LDA) are fulfilled by dataset $iii$.

Looking at dataset $ii$, class $Y = 0$ seems to have a Gaussian distribution whereas the class $Y = 1$ does not have a Gaussian distribution. Hence dataset $ii$ does not fulfill the assumptions of either LDA or QDA.

(Note that even though dataset $ii$ does not fulfill the assumptions of QDA, a QDA decision boundary can still separate the classes in this case! The same holds for LDA and dataset $iii$. This shows that the LDA/QDA classifier can perform well in practice, even if the assumptions are not satisfied.)

(c) Note first that none of the figures have periodic sample paths. Consequently, model M3 (which has a periodic kernel) is the model which is not represented among the four. For comparison, sample paths from model M3 are shown in the figure below.

3
Next, we note that the noise variance $\sigma^2$ decides how concentrated the posterior distribution will be around the observed data points. Specifically, setting $\sigma^2 = 0$ is the same as saying that the function $f$ is observed exactly, i.e. $y_i = f(x_i)$. This forces the sample paths from the posterior distribution to pass exactly through the measured data. Finally, we note that the length scale $\ell$ determines how quickly the correlation between different points in the $x$-space decays. This is also visible from the plots of the kernels $k(r) = k(|x - x'|)$, corresponding to the covariance between two input points at a distance of $r$ from each other. Based on these insights, we can draw the following conclusions:

**Figure ii:** All sample paths pass exactly through the measured data points. Hence this plot must correspond to model M5, which is the only model with $\sigma^2 = 0$. It can also be observed that the sampled paths are a bit “ragged” which is typical for a Matérn kernel.

**Figure iii:** All sample paths seem to be almost independent of the measured data points (at least compared to the other three figures). Hence this plot must correspond to the distribution with the highest noise variance. M2 has the highest noise variance ($\sigma^2 = 3^2$ compared to $\sigma^2 = 0.3^2$ and $\sigma^2 = 0$) so this must be samples from M2. It can also be observed that the sampled paths are very smooth, hence they should belong to a squared-exponential kernel.

**Figure i:** There are strong correlations between points quite far apart (compare to Figure vi where only points very close to each other are noticeably correlated). Hence this plot must correspond
to M4 which has a large length scale $\ell = 5$ (i.e., the plot of $k(r)$ decays slowly to zero for M4). Note also that the paths are a bit ragged and should therefore belong to a Matérn kernel.

**Figure iv:** There is very little correlation between points far apart which corresponds to a kernel with a small length scale. Note also that the paths are smooth and should therefore belong to a squared-exponential kernel. Hence this plot must correspond to M1.

4. (a) The model $Y = \beta_0 + \beta_1 X + \varepsilon$ for training points $T = \{(x_1, y_1), (x_2, y_2)\} = \{(-1, 2), (1, 1)\}$ gives

$$
\begin{pmatrix}
y_1 \\
y_2
\end{pmatrix} =
\begin{pmatrix}
\beta_0 + \beta_1 x_1 + \varepsilon_1 \\
\beta_0 + \beta_1 x_2 + \varepsilon_2
\end{pmatrix} =
\begin{pmatrix}
1 & x_1 \\
1 & x_2
\end{pmatrix}
\begin{pmatrix}
\beta_0 \\
\beta_1
\end{pmatrix} +
\begin{pmatrix}
\varepsilon_0 \\
\varepsilon_1
\end{pmatrix}
$$

$$
\iff \ y = X\beta + \varepsilon
$$

where (for the given training points)

$$
X = \begin{pmatrix}
1 & -1 \\
1 & 1
\end{pmatrix}, \ X^T = \begin{pmatrix}
1 & 1 \\
-1 & 1
\end{pmatrix}, \ X^T X = \begin{pmatrix}
2 & 0 \\
0 & 2
\end{pmatrix}
$$

$$
(X^T X)^{-1} = \frac{1}{2} \begin{pmatrix}
1 & 0 \\
0 & 1
\end{pmatrix}, \ y = \begin{pmatrix}
2 \\
1
\end{pmatrix}
$$

$\hat{\beta}_{LS}$ is then given by

$$
\hat{\beta}_{LS} = (X^T X)^{-1} X^T y = \begin{pmatrix}
3/2 \\
-1/2
\end{pmatrix}.
$$

(This problem can also be solved by noting that the least-squares estimate of $\beta$ must correspond to the straight line which connects the two training data points.)

(b) Same $X$ and $y$ as in a) for the given training points gives

$$
\hat{\beta}_{RR} = (X^T X + \lambda I)^{-1} X^T y = \left( \begin{pmatrix}
2 & 0 \\
0 & 2
\end{pmatrix} + \begin{pmatrix}
\lambda & 0 \\
0 & \lambda
\end{pmatrix} \right)^{-1} \begin{pmatrix}
1 & 1 \\
-1 & 1
\end{pmatrix} \begin{pmatrix}
2 \\
1
\end{pmatrix}
$$

$$
= \left( \begin{pmatrix}
2 + \lambda & 0 \\
0 & 2 + \lambda
\end{pmatrix} \right)^{-1} \begin{pmatrix}
3 \\
-1
\end{pmatrix} = \frac{1}{2+\lambda} \begin{pmatrix}
3 \\
-1
\end{pmatrix}
$$

(c) The hint (LASSO estimate of $\beta_j$ is either 0 or the same sign as $\hat{\beta}_j$ for LS) gives that for LASSO $\hat{\beta}_0$ is either zero or positive and $\hat{\beta}_1$
is either zero or negative. LASSO aims to minimize the following sum (for given training data)

\[ S = \sum_{i=1}^{2} (y_i - (\beta_0 + \beta_1 x_i))^2 + \lambda \sum_{j=0}^{1} |\beta_j| \]

\[ = (2 - (\beta_0 - \beta_1))^2 + (1 - (\beta_0 + \beta_1))^2 + \lambda |\beta_0| + \lambda |\beta_1| \]

\[ = [\text{Use hint, } \beta_0 \geq 0 \text{ and } \beta_1 \leq 0] = \]

\[ = 5 - 6\beta_0 + 2\beta_1 + 2\beta_0^2 + 2\beta_1^2 + \lambda \beta_0 - \lambda \beta_1, \quad (\star) \]

where the last equality holds for \( \lambda \) sufficiently small, so that both \( \beta_0 \) and \( \beta_1 \) are non-zero. Now, minimize the sum with respect to \( \beta \) gives

\[ \frac{\partial S}{\partial \beta_0} = -6 + 4\beta_0 + \lambda = 0 \quad \Rightarrow \quad \beta_0 = \begin{cases} 6 - \lambda & \text{if } \lambda < 6 \\ 0 & \text{if } \lambda \geq 6 \end{cases} \]

\[ \frac{\partial S}{\partial \beta_1} = 2 + 4\beta_1 - \lambda = 0 \quad \Rightarrow \quad \beta_1 = \begin{cases} -2 + \lambda & \text{if } \lambda < 2 \\ 0 & \text{if } \lambda \geq 2 \end{cases} \]

(technically, the equation \((\star)\) only holds for \( \lambda \leq 2 \), since from the solution for \( \beta_1 \) there is a breakpoint at \( \lambda = 2 \). However, since there are no cross terms between \( \beta_0 \) and \( \beta_1 \) in \( S \), this does not affect the solution for \( \beta_0 \). In other words, for the interval \( \lambda = [2, 6] \) we have another expression for \( S \) obtained by setting \( \beta_0 = 0 \) in \((\star)\), which results in the same solution for \( \beta_1 \) as above.)

5. (a) Parametric models are parameterized using a fixed-dimensional vector of parameters. Learning a parametric model amounts to estimating the values of these parameters. These estimates are subsequently used to define the prediction model for a parametric model. Nonparametric models, on the other hand, use (in some way) the training data as parameters, i.e. the prediction model for a parametric model is expressed directly in terms of the training data. Nonparametric models can thus be viewed as having infinitely many parameters. The flexibility of a nonparametric model thus increases as we use more training data.

(b) From the problem we have that the probability for \( Y = 1 \) (according to the model) is the logistic function of \( \beta^T X \) i.e.

\[ \Pr(Y = 1 \mid \beta, X) = \frac{e^{\beta^T X}}{1 + e^{\beta^T X}}. \]
Since the probability of \( Y = 0 \) is the complement to the probability of \( Y = 1 \) we have,

\[
\Pr (Y = 0 \mid \beta, X) = 1 - \Pr (Y = 1 \mid \beta, X) = 1 - \frac{e^{\beta^T X}}{1 + e^{\beta^T X}} = \frac{1}{1 + e^{\beta^T X}}.
\]

Since the first element of \( \beta \) is an intercept we insert a 1 first in \( x \) which leads to the estimate,

\[
\hat{\beta}^T \left( \begin{array}{c} 1 \\ x^* \end{array} \right) = \left( \begin{array}{ccc} 1 & -1 & 2 \\ 2 & 1 & -1 \end{array} \right) = 2,
\]

and finally

\[
\Pr \left( Y = 0 \mid \beta, X = \left( \begin{array}{c} 1 \\ x^* \end{array} \right) \right) = \frac{1}{1 + e^2} \approx 12\%.
\]

(c) Since all data points are independent we can write the full probability as a product i.e.,

\[
\Pr(Y_1 = y_1, \ldots, Y_N = y_N \mid X_1 = x_1, \ldots, X_N = x_N, \beta)
= \prod_{i=1}^N \Pr(Y = y_i \mid X = x_i, \beta).
\]

\( \Pr(Y = y_i \mid X = x_i, \beta) \) can be expressed in multiple ways by splitting the two outcomes of \( Y \). This example will use,

\[
\Pr(Y = y_i \mid X = x_i, \beta) = p(x_i, \beta)^{y_i} (1 - p(x_i, \beta))^{1-y_i},
\]

where \( p(x, \beta) \) is taken from the problem formulation (this is indeed the standard approach for deriving the likelihood for a logistic regression, see e.g. ESL(p120)). However you could also consider

\[
\Pr(Y = y_i \mid X = x_i, \beta) = y_i p(x_i, \beta) + (1 - y_i) (1 - p(x_i, \beta)),
\]

or more explicitly

\[
\Pr(Y = y_i \mid X = x_i, \beta) = I(y_i = 1) p(x_i, \beta) + I(y_i = 0) (1 - p(x_i, \beta)),
\]

where \( I(\cdot) \) is the indicator function. Using the first expression of these three we get,

\[
\Pr(Y_1 = y_1, \ldots, Y_N = y_N \mid X_1 = x_1, \ldots, X_N = x_N, \beta)
= \prod_{i=1}^N p(x_i, \beta)^{y_i} (1 - p(x_i, \beta))^{1-y_i}.
\]
(d) Recall that for the logistic regression model we have $p(x, \beta) = \frac{e^{\beta^T x}}{1 + e^{\beta^T x}}$ and $1 - p(x, \beta) = \frac{1}{1 + e^{\beta^T x}}$. Using this we get

$$\log \Pr(Y_1 = y_1, \ldots, Y_N = y_N \mid X_1 = x_1, \ldots, X_N = x_N, \beta)$$

$$= \log \prod_{i=1}^{N} p(x_i, \beta)^{y_i} (1 - p(x_i, \beta))^{1 - y_i}$$

$$= \sum_{i=1}^{N} y_i \log p(x_i, \beta) + (1 - y_i) \log (1 - p(x_i, \beta))$$

$$= \sum_{i=1}^{N} y_i \beta^T x_i - y_i \log(1 + e^{\beta^T x_i}) - (1 - y_i) \log(1 + e^{\beta^T x_i})$$

$$= \sum_{i=1}^{N} y_i \beta^T x_i - \log(1 + e^{\beta^T x_i}).$$