SVR:

Support Vector Regression (SVR) is a machine learning algorithm for regression problems, aiming to minimize the generalization error bound. (cite Basak)
SVR is an extension of Support Vector Machines (SVM); therefore, it is necessary to explain SVM first.
Typically used in classification, SVM seeks to identify a hyperplane that effectively separates data points into distinct classes, maximizing the margin between the hyperplane and the nearest data points of each class while minimizing classification errors. When new data points need to be classified, the hyperplane acts as the decision boundary.(cite website https://www.analyticsvidhya.com/blog/2020/03/support-vector-regression-tutorial-for-machine-learning/)

BILD 0

Figure 0.0 shows the assignment to a category divided by the hyperplane. This data is called linearly separable.

BILD 1

Sometimes, the data is more complex, and a simple line for class separation is not possible. Like in Figure 0.1, the classes can be separated using a new dimension: the z-axis. After this transformation, the data is linearly separable in three dimensions.

BILD 2

After transforming the graph back into 2D space, the hyperplane looks like a circle in Figure 0.2

In practical scenarios, when dealing with datasets containing numerous features, applying transformations involving multiple polynomial combinations of these features can result in unnecessary computational expenses. This is because operations with the higher-dimensional vectors in the transformed feature space are required to train a support vector classifier and optimize the objective function.(https://towardsdatascience.com/the-kernel-trick-c98cdbcaeb3f).
The solution to this problem is the so-called "kernel trick". It is a technique that allows you to avoid the need to explicitly convert the data into a higher-dimensional space. Instead, kernel functions focus on comparing the original data observations based on their similarities.

Kernel function takes x,z as inputs vectors in the original space and returns a scalar that represents the inner product of vectors in a potentially much higher-dimensional space.
If we have data x,z el X and a map: phi : X-> Rn then k(x,z) = <phi(x),phi(z)> is a kernel function (https://towardsdatascience.com/the-kernel-trick-c98cdbcaeb3f)

\text{If we have data } \mathbf{x}, \mathbf{z} \in \mathcal{X} \text{ and a map } \phi: \mathcal{X} \rightarrow \mathbb{R}^N \text{ then} is a kernel function.

k(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle

BILD FORMULA

So this way, it computes these inner products without ever having to map the data points into the higher dimensions, which does not require a high computational effort. Naturally, all these calculations happen inside the kernel functions when implementing it. (cite ski kit learn). Linear Kernel is the simplest kernel form, used when the data is linearly separable. In SVR, it effectively means no transformation is needed, and the original feature space is used. The Equation looks like k(x,z)=xTz.
Other popular kernels are polynomial, radial basis function (RBF), and sigmoid. (cite paper https://dergipark.org.tr/tr/download/article-file/1047384).

In SVR, on the other hand, the hyperplane represents the regression line that minimizes the error between the predicted and actual output values while maximizing the margin. The hyperplane is determined by support vectors, which are the data points closest to the hyperplane and significantly influence the model, while other data points have little or no impact. (cite https://medium.com/@vk.viswa/support-vector-regression-unleashing-the-power-of-non-linear-predictive-modeling-d4495836884).

So, in this case, the output is not a categorical label; it is a continuous value representing the predicted target variable for each data point. (cite website https://www.analyticsvidhya.com/blog/2020/03/support-vector-regression-tutorial-for-machine-learning/)
In emotion recognition tasks, arousal and valence are essential components typically evaluated on continuous scales (Citron, Weekes, & Ferstl, 2012 https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4098114/). Therefore, SVR is well-suited for such scenarios as it involves estimating a position along a continuous range, making it a more appropriate option.

BILD (https://medium.com/it-paragon/support-vector-machine-regression-cf65348b6345)

Figure 0.4 shows Linear SVR and Non-linear SVR. Each graph represents the concept of fitting an SVR model to data points marked with triangles and squares. The goal is to find the function that best fits the data while allowing for some errors within a certain margin. The following parameters are displayed:

1. **Epsilon (ε)**: the width of the margin around the predicted function. Data points within this margin are considered acceptable.
2. **Slack Variables (ξ)**: the distances by which the data points exceed the epsilon margin. The goal is to minimize the slack variables, which means minimizing the errors outside the margin.

Bild ueberschrift: **Linear SVR** and **Non-linear SVR after a transformation of the features into a higher-dimensional space via a kernel function**

In practice, Grid Search Cross-Validation (GridSearchCV) inside SVR can be used in emotion recognition. This method works through multiple combinations of parameter tunes to determine which tune gives the best performance.

Each combination of C and kernel will be tried in a defined parameter grid. The regularization parameter C determines the trade-off between achieving a low training error and a low testing error (overfitting). The kernel parameter selects the type of kernel to be used in the algorithm. Once GridSearchCV has tested all parameter combinations, it will identify and return the model with the best performance metrics. This model is then used to predict arousal or valence on the development and test datasets.
(https://scikit-learn.org/stable/modules/grid_search.html)