Neural networks and random forests are examples of models.

* A typical **supervised learning** task is **classification.** The spam filter is a good example of this: it is trained with many example emails along with their class (spam or ham), and it must learn how to classify new emails
* Another typical task is to **predict** a target numeric value, such as the price of a car, given a set of features (mileage, age, brand, etc.). This sort of task is called **regression**.

\*\*\***some regression models can be used for classification as well, and vice versa**

* **Unsupervised learning -** hierarchical clustering algorithm, it may also subdivide each group into smaller groups
* simply comparing new data points to known data points, or instead by detecting patterns in the training data and building a predictive model, much like scientists do (**instance-based versus model-based learning**)
* **dimensionality reduction**, in which the goal is to simplify the data without losing too much information. One way to do this is to merge several correla‐ ted features into one. For example, a car’s mileage may be strongly correlated with its age, so the dimensionality reduction algorithm will merge them into one feature that represents the car’s wear and tear. This is called **feature extraction.**
* another common unsupervised task is **association rule** learning, in which the goal is to dig into large amounts of data and discover interesting relations between attributes. For example, suppose you own a supermarket. Running an association rule on your sales logs may reveal that people who purchase barbecue sauce and potato chips also tend to buy steak. Thus, you may want to place these items close to one another.
* **Semi-supervised learning -** a clustering algorithm may be used to group similar instances together, and then every unlabeled instance can be labeled with the most common label in its cluster. Once the whole dataset is labeled, it is possible to use any supervised learning algorithm.  
    
  **\*\*\* “unsupervised learning” is generally used when dealing with tasks like clustering, dimensionality reduction, or anomaly detection,  
  \*\*\*self-supervised learning focuses on the same tasks as supervised learning: mainly classification and regression.**
* **Reinforcement learning** is a very different beast. The learning system, called an agent in this context, can observe the environment, select and perform actions, and get rewards in return
* **feature engineering**, involves the following steps:

**• Feature selection** (selecting the most useful features to train on among existing features)

• **Feature extraction** (combining existing features to produce a more useful one—as we saw earlier, dimensionality reduction algorithms can help)

• Creating new features by gathering new data

* **Overfitting -** the model performs well on the training data, but it does not generalize well. Constraining a model to make it simpler and reduce the risk of overfitting is called **regularization**. amount of regularization to apply during learning can be controlled by a hyper‐ parameter. A **hyperparameter** is a parameter of a learning algorithm. If you set the regularization hyperparameter to a very large value, you will get an almost flat model (a slope close to zero); the learning algorithm will almost certainly not overfit the training data, but it will be less likely to find a good solution.
* **Underfitting in Machine Learning -** Underfitting occurs when a machine learning model is too simple to capture the underlying patterns in the data. As a result, the model performs poorly on both the training data and unseen (test) data. This typically happens when the model has **high bias** and lacks the capacity to learn the complexity of the data.
* **Holdout validation -** y hold out part of the training set to evaluate several candidate models. held-out set is called the validation set (or the devel‐ opment set, or dev set). More specifically, you train multiple models with various hyperparameters on the reduced training set (i.e., the full training set minus the validation set), and you select the model that performs best on the validation set. After this holdout validation process, you train the best model on the full training set (including the validation set), and this gives you the final model. Lastly, you evaluate this final model on the test set to get an estimate of the generalization error. perform repeated cross-validation, using many small validation sets. Each model is evaluated once per validation set after it is trained on the rest of the data. By averaging out all the evaluations of a model, you get a much more accurate measure of its performance.
* **Error Rate Identification –** RMSE(Root Mean Square Error) is more sensitive to outliers than the MAE(Mean Absolute Error). But when outliers are exponentially rare (like in a bell-shaped curve), the RMSE performs very well and is generally preferred.
* the test set generated using **stratified sampling** has income category proportions almost identical to those in the full dataset, whereas the test set generated using **purely random sampling** is skewed.
* **k-fold crossvalidation** means splitting the training set into k folds (in this case, three), then training the model k times, holding out a different fold each time for evaluation
* recall =
* Precision =
* F1 score is the harmonic mean of precision and recall.  
  F1 score =
* lowering the threshold increases recall and reduces precision.

To predict – **median Housing Prices in any district**

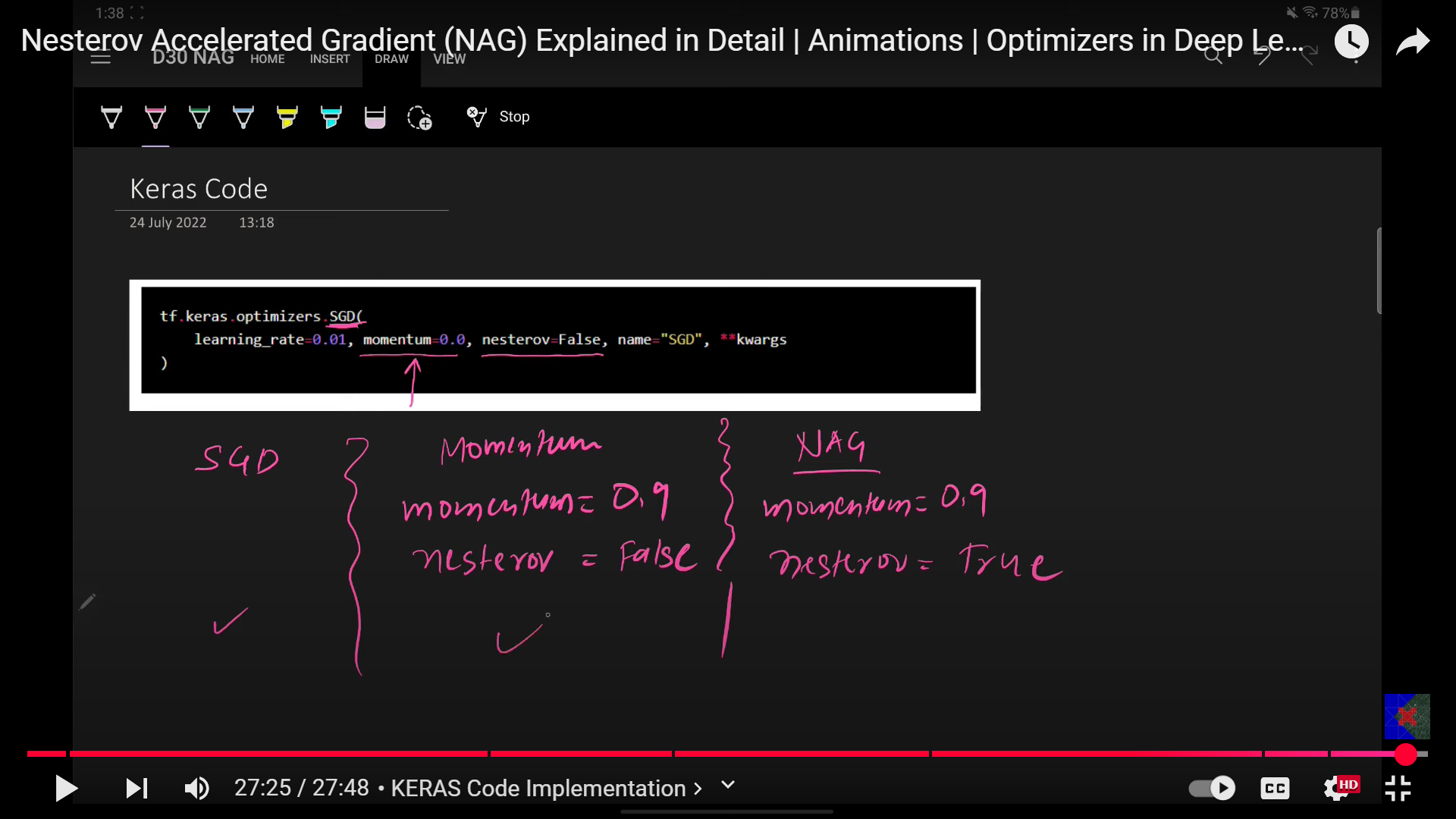
Population

median income

median housing price for each district  
\* We can use regression to solve this. Data is not going to change frequently so simple model will work fine and we don’t have large amt of data so we don’t require data to be partitioned to be used.

**Deep Learning**

**How to implement optimizers in NN?**

****