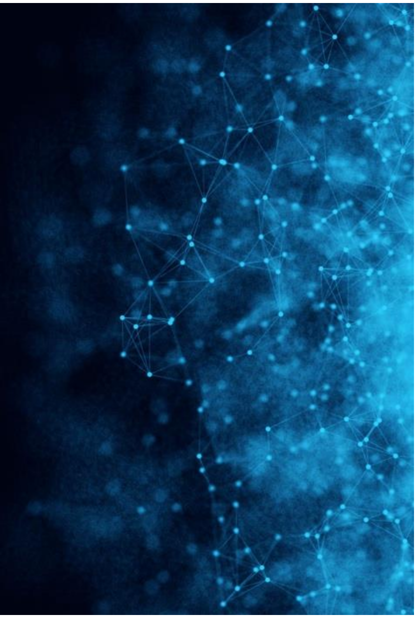


DA2011
Machine Learning I

Lecture 2

Dr. Deshanee Wickramarachchi
28th September 2025

Today you will learn…

- Supervised Learning algorithms for Regression
  - Linear Regression Model
    - Fit an OLS regression model
    - Metrics for model accuracy
    - Significance of the model and coefficients
  - Support Vector Regression
    - Hyperparameter tunning

Linear Models for Regression

- Recall that in prediction models, we believe that there is some relationship between the outcome and the predictors.

$$Y_i = f(X_i) + \varepsilon_i \quad \text{for } i = 1, \dots, n$$

- For linear models, the function $f(X_i)$ is a linear function.
- The general prediction formula would look like:

$$\hat{y} = \hat{\beta}_o + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_p x_p$$

- For simplicity, today we will focus on linear models with a single predictor.

Simple Linear Regression

- Mathematically we can write the relationship as:

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

- Here, β_0 and β_1 are two unknown constants, known as the *intercept* and *slope* coefficients in the model.
- We use training data to estimate the values of these two unknown constants so that the fitted line is as close as possible to all data points.
- Then the prediction model (fitted line) is:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

Fitting Regression Line

- How do we measure the closeness of data points to the fitted line?

- Minimizing the residual sum of squares.

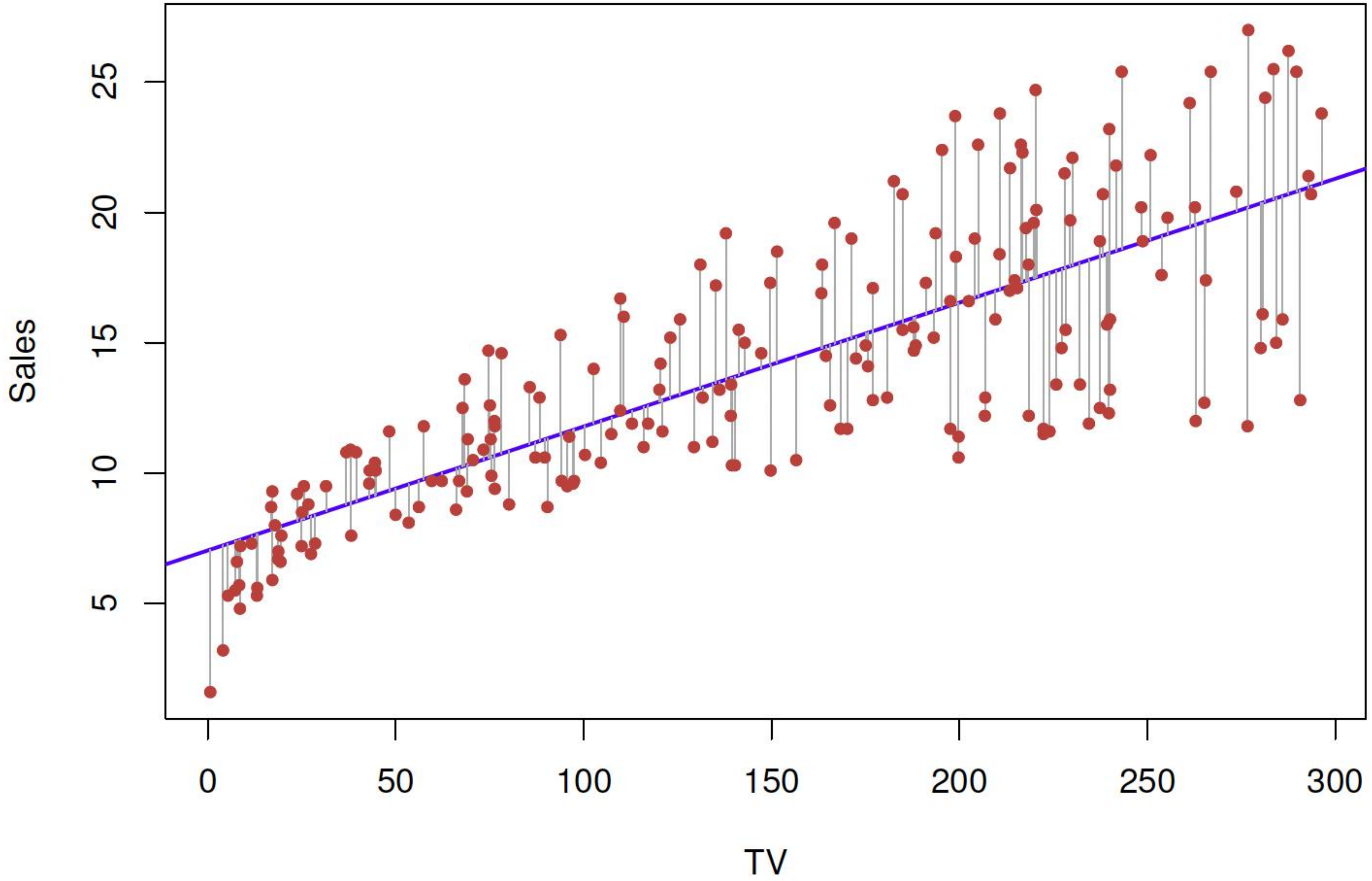


Figure 1: Relationship between TV advertising budget and sales

Ordinary Least Squares (OLS) Regression

- Let $\hat{y}_i$ denote the prediction for the i^{th} unit.
- Then the i^{th} residual is:

$$\begin{aligned} e_i &= y_i - \hat{y}_i \\ &= y_i - \hat{\beta}_0 + \hat{\beta}_1 x_i \end{aligned}$$

- The Residual Sum of Squares (RSS) is:

$$RSS = e_1^2 + e_2^2 + \cdots + e_n^2$$

- The least squares approach chooses $\hat{\beta}_0$ and $\hat{\beta}_1$ that minimizes RSS.

OLS Regression in Python

- `scikit-learn` library has the class `LinearRegression()`.
- User Guide:

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

- `LinearRegression().fit()` – to fit the linear model
- `model.predict()` – to predict using the `model`
- `model.score()` – Returns coefficient of determination (R^2).

Activitiy 1

1. Import the **advertising_TV** dataset into Google Colab.
2. Split the data into training and test set using 80%-20% ratio.
3. Draw a scatter plot for the training data.
4. Add test set data points to the same graph.
5. Fit OLS regression model.
6. Draw the fitted regression line on the scatter plot.
7. Find the R^2 of the fitted model.

Assessing Model Accuracy

Coefficient of Determination (R^2)

- RSS is measured in the units of Y.
- The R^2 statistic provides an alternative measure of fit.
- It is the proportion of variance that is explained by the model.
- R^2 is independent of the scale of Y.

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS}$$

where $TSS = \sum(y_i - \bar{y})^2$ is the Total Sum of Squares.

R^2 continued…

- TSS measures ____ .
- RSS measures ____ .
- An R^2 value that is close to 1 indicates ____________________________
  ____ .
- An R^2 value that is close to 0 indicates ____________________________
  ____ .
- However, there is no rule on what is a good R^2 value.

R^2 and Correlation

- R^2 is a measure of the linear relationship between X and Y.
- Therefore, it is closely related to the correlation coefficient.

Try It Yourself!

1. For the advertising data example, fit a simple linear regression model and find the R^2.
2. Find the correlation coefficient between X and Y.
3. Explore the relationship between the values obtained in parts 1 and 2.

Other Evaluation Metrics

Mean Absolute Error (MAE)

- This is the average of absolute differences between the actual and predicted outcome values.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- MAE tells you, on average, how far your predictions are from the actual values.
- MAE has the same units as the response variable.

Mean Squared Error (MSE) and RMSE

- MSE is the average of squared errors.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{RSS}{n}$$

- Units are the square of the outcome variable’s units.
- Therefore, sometimes we use the square root of it, which is referred to as the Root Mean Squared Error (RMSE).

$$RMSE = \sqrt{MSE}$$

`sklearn.metrics`

- This is the class that contains model evaluation metrics in `sklearn`.

https://scikit-learn.org/stable/api/sklearn.metrics.html#

- MAE

```
from sklearn.metrics import mean_absolute_error
```

- MSE

```
from sklearn.metrics import mean_squared_error
```

Inference on Coefficients

Significance of Regression Coefficients

- We assume the linear model:

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

- Assuming that the random errors are having zero mean:

$$E(Y) = \beta_0 + \beta_1 X$$

- β_o is the expected value of Y when $X = 0$.

- β_1 is the average change in Y for a one-unit increase in X.

Significance of Regression Coefficients

- Using a sample of data we try to estimate the unknown coefficients β_0 and β_1.
- Our estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ won't be exactly equal to the real β_0 and β_1.
- Therefore, using the mean and standard deviation of our estimators, we draw conclusions about the unknown true parameters.
- We are mainly interested in testing whether $\beta_1 = 0$ or not. Because we are mainly concerned about the relationship between X and Y.

Inference on $\boldsymbol{\beta_1}$

$$H_0: \beta_1 = 0 \text{ vs.}$$
$$H_1: \beta_1 \neq 0$$

- A non-zero β_1 coefficient implies that there is a significant linear relationship between X and Y.
- We can contruct a $(1-\alpha)100\%$ confidence interval for β_1 and test the above hypotheses.
- We need to use `statsmodels` library to check the significance of regression coefficients.

Try It Yourself!

1. For the advertising data example, check the significance of the coefficients.

Activitiy 2

- Use the California Housing Dataset to answer the following.

1. Use the below code to import the data:

```
from sklearn.datasets import fetch_california_housing
housing = fetch_california_housing()
```

1. housing.data contains the features and housing.target contains the target variable. Use **median income in block group, MedInc** as the only predictor and **median house value** as the outcome to fit a regression model.
2. Split the dataset into training and test tests using 70%-30% ratio and random state as your birth year.
3. Fit a simple linear regression model and assess the model accuracy.
4. Repeat the above steps 3 and 4 by splitting the data using 2025 as the random state.

When to use linear regression

- When the outcome and the predictor have a linear relationship.
- Interpretability and simplicity matter more than complexity.

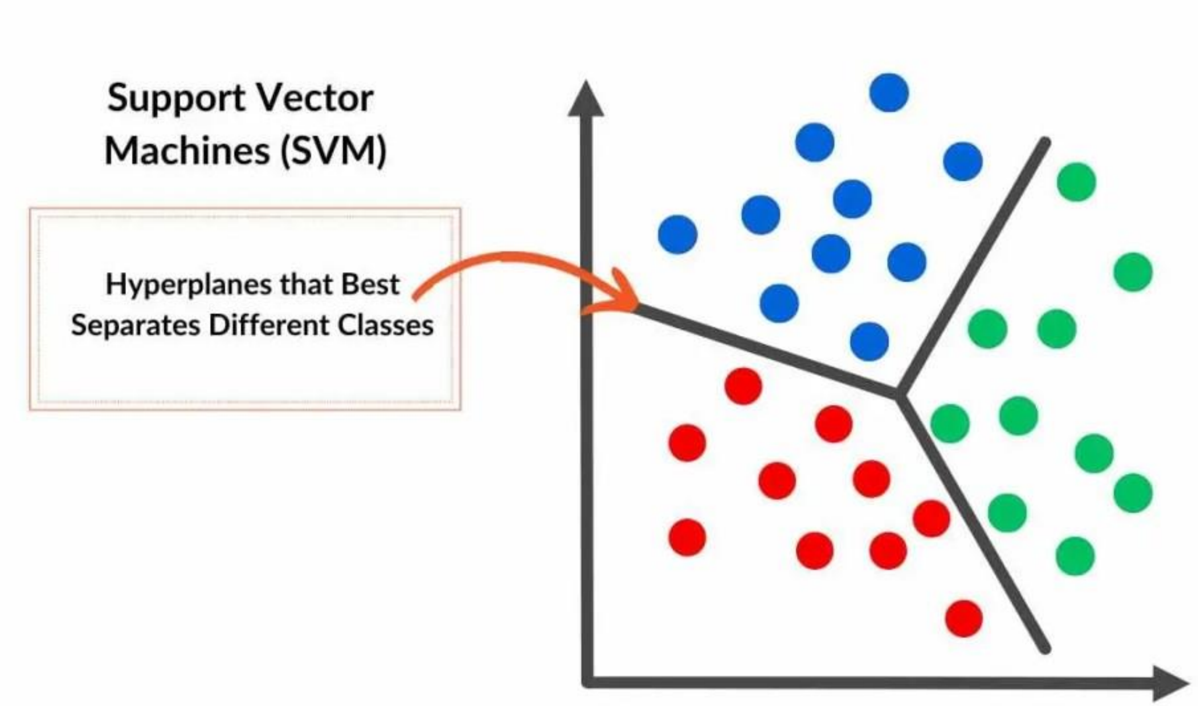
When NOT to use linear regression

- If the data shows strong non-linear patterns.
- When the data are high-dimensional and complex.
- When the data contains many extreme outliers that dominate the fit.

Support Vector Regression (SVR)

Support Vector Regression (SVR)

- SVR is a type of support vector machine (SVM).
- It tries to find a function that best predicts the continuous output value for a given input value.
- SVR is sensitive to feature scaling.
- **Kernels:** SVR can use different types of kernels, which are functions that determine the similarity between input vectors. SVR can use both linear and non-linear kernels.



Source: https://spotintelligence.com/2024/05/06/support-vector-machines-svm/

kernel : *{'linear', 'poly', 'rbf', 'sigmoid', 'precomputed'} or callable, default='rbf'*

Hyperparameter Tunning

- SVR has multiple parameters such as C, gamma and epsilon.
- Hyperparameter tunning involves finding the best combination of these parameters that optimizes model performance.
- Usually requires searching through a range of possible values for these parameters.

C : ***float, default=1.0***

Regularization parameter. The strength of the regularization is inversely proportional to C.

gamma : ***{'scale', 'auto'} or float, default='scale'***

Kernel coefficient for 'rbf', 'poly' and 'sigmoid'.

epsilon : ***float, default=0.1***

Thank you
See you next week!