AGENDA

1. Feature Engineering and Feature Selection
2. Introduction to Machine Learning & Linear Regression Analysis
3. Model Evaluation

Data Science Academy

I

Feature Engineering and Feature Selection

Data Science Academy

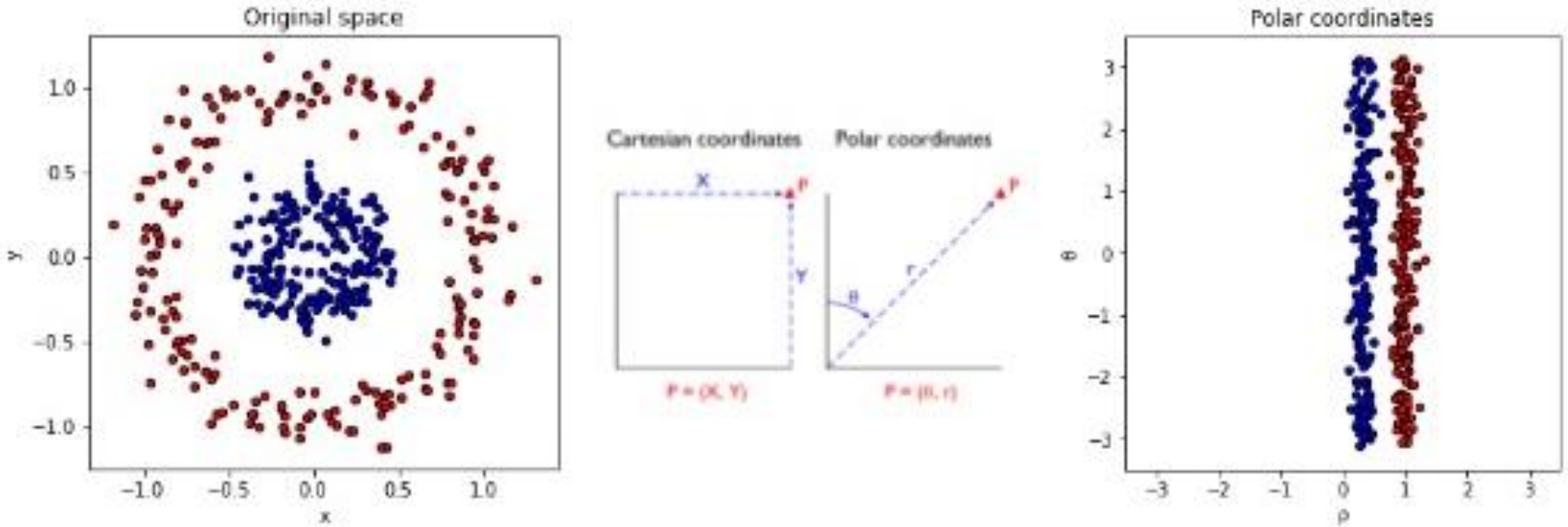
Feature Engineering

- Pre-processing refers to the transformations applied to our data before feeding it to the algorithm.
- Data Preprocessing is a technique that is used to convert the raw data into a clean data set.
- Whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.

Bütün hüquqlar qorunur.

Feature Engineering

- Feature engineering is the process of extracting useful features from raw data using math, statistics and domain knowledge

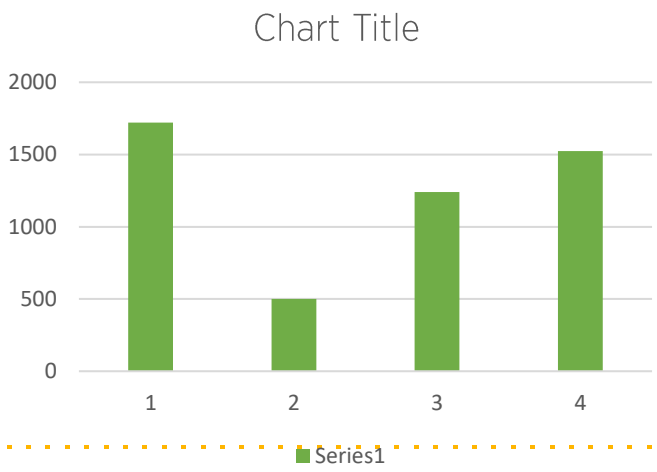


Bütün hüquqlar qorunur.

Feature Engineering

- To eleminate outliers refer to
  - Domain knowledge: price per m^2>1000
  - Visualization:
  - Math/Statistics

m^2	Price
90	155000
20	10000
75	93000
120	183000



m^2	Price	Price per m^2
90	155000	1722
20	10000	500
75	93000	1240
120	183000	1525

Bütün hüquqlar qorunur.

Feature Engineering

- Binarize Data (Make Binary)
- We can transform our data using a binary threshold. All values above the threshold are marked 1 and all equal to or below are marked as 0.
- This is called binarizing your data or threshold your data. It can be useful when you have probabilities that you want to make crisp values. It is also useful when feature engineering and you want to add new features that indicate something meaningful.
- We can create new binary attributes in Python using scikit-learn with the Binarizer class.
- We can see that all values equal or less than 0 are marked 0 and all of those above 0 are marked 1

Bütün hüquqlar qorunur.

Binarize Data

In [ ]:

```
age = data_set.iloc[:, 1].values
salary = data_set.iloc[:, 2].values
from sklearn.preprocessing import Binarizer

x = age
x = x.reshape(1, -1)
y = salary
y = y.reshape(1, -1)

# For age, let threshold be 35
# For salary, let threshold be 61000
binarizer_1 = Binarizer(35)
binarizer_2 = Binarizer(61000)

# Transformed feature
Binarized_age = binarizer_1.fit_transform(x)  # [[1 0 0 1 1 0 1 1 1 1]]
Binarized_salary = binarizer_2.fit_transform(y) # [[1 0 0 0 0 0 0 1 1 1]]

data_set ['Bin.Age>35'] = Binarized_age[0]
data_set ['Bin.S>6100'] = Binarized_salary[0]
```


Bütün hüquqlar qorunur.

Binarize Data

Country	Age	Salary
France	44	72000
Spain	27	48000
Germany	30	54000
Spain	38	61000
Germany	40	1000
France	35	58000
Spain	78	52000
France	48	79000
Germany	50	83000
France	37	67000

Country	Age	Salary	Bin. Age>35	Bin. S>6100
France	44	72000	1	1
Spain	27	48000	0	0
Germany	30	54000	0	0
Spain	38	61000	1	0
Germany	40	1000	1	0
France	35	58000	0	0
Spain	78	52000	1	0
France	48	79000	1	1
Germany	50	83000	1	1
France	37	67000	1	1

Bütün hüquqlar qorunur.

Feature Engineering

- One-Hot Encoding
- The first step for us is known as one-hot encoding of the data. This process takes categorical variables, such as days of the week and converts it to a numerical representation without an arbitrary ordering.

Week	Mon	Tue	Wed	Thu	Fri
Mon	1	0	0	0	0
Tue	0	1	0	0	0
Wed	0	0	1	0	0
Thu	0	0	0	1	0
Fri	0	0	0	0	1

Bütün hüquqlar qorunur.

Feature Engineering

- Label Encoding
- Refers to converting the labels into numeric form so as to convert it into the machine-readable form.
- Machine learning algorithms can then decide in a better way on how those labels must be operated.

SAFETY LEVEL (TEXT)	SAFETY LEVEL (NUMERICAL)
None	0
Low	1
Medium	2
High	3
Ver-High	4

Bütün hüquqlar qorunur.

Role of feature selection

- How to select features and what are Benefits of performing feature selection before modeling your data?

  - Reduces Overfitting: Less redundant data means less opportunity to make decisions based on noise.
  - Improves Accuracy: Less misleading data means modeling accuracy improves.
  - Reduces Training Time: fewer data points reduce algorithm complexity and algorithms train faster.

Bütün hüquqlar qorunur.

Feature Selection Methods

- Feature selection techniques that are easy to use and also gives good results:
  - Univariate Selection
  - Feature Importance
  - Correlation Matrix with Heatmap

Bütün hüquqlar qorunur.

Univariate Selection

- Statistical tests can be used to select those features that have the strongest relationship with the output variable.
- The scikit-learn library provides the SelectKBest class that can be used with a suite of different statistical tests to select a specific number of features.

Bütün hüquqlar qorunur.

Feature Importance

- You can get the feature importance of each feature of your dataset by using the feature importance property of the model.
- Feature importance gives you a score for each feature of your data, the higher the score more important or relevant is the feature towards your output variable.
- Feature importance is an inbuilt class that comes with Tree Based Classifier.

Bütün hüquqlar qorunur.

Correlation Matrix with Heatmap

- Correlation states how the features are related to each other or the target variable.
- Correlation can be positive (increase in one value of feature increases the value of the target variable) or negative (increase in one value of feature decreases the value of the target variable)
- Heatmap makes it easy to identify which features are most related to the target variable.

Bütün hüquqlar qorunur.

II
Introduction to Machine Learning & Linear Regression Analysis (Simple and Multiple)

What is Machine Learning?

- Machine learning is a method of data analysis that automates analytical model building.
- Using algorithms that iteratively learn from data, machine learning allows computers to find hidden insights without being explicitly programmed where to look.

Bütün hüquqlar qorunur.

Applications of Machine Learning

- Facebook face recognition
- Voice recognition and text
- Netflix recommendations
- Cancer prediction and prognosis
- and so on...

Bütün hüquqlar qorunur.

Types of Machine Learning

- There are 3 main types of Machine Learning, these types are:
  - Supervised Learning
  - Unsupervised Learning
  - Reinforcement Learning

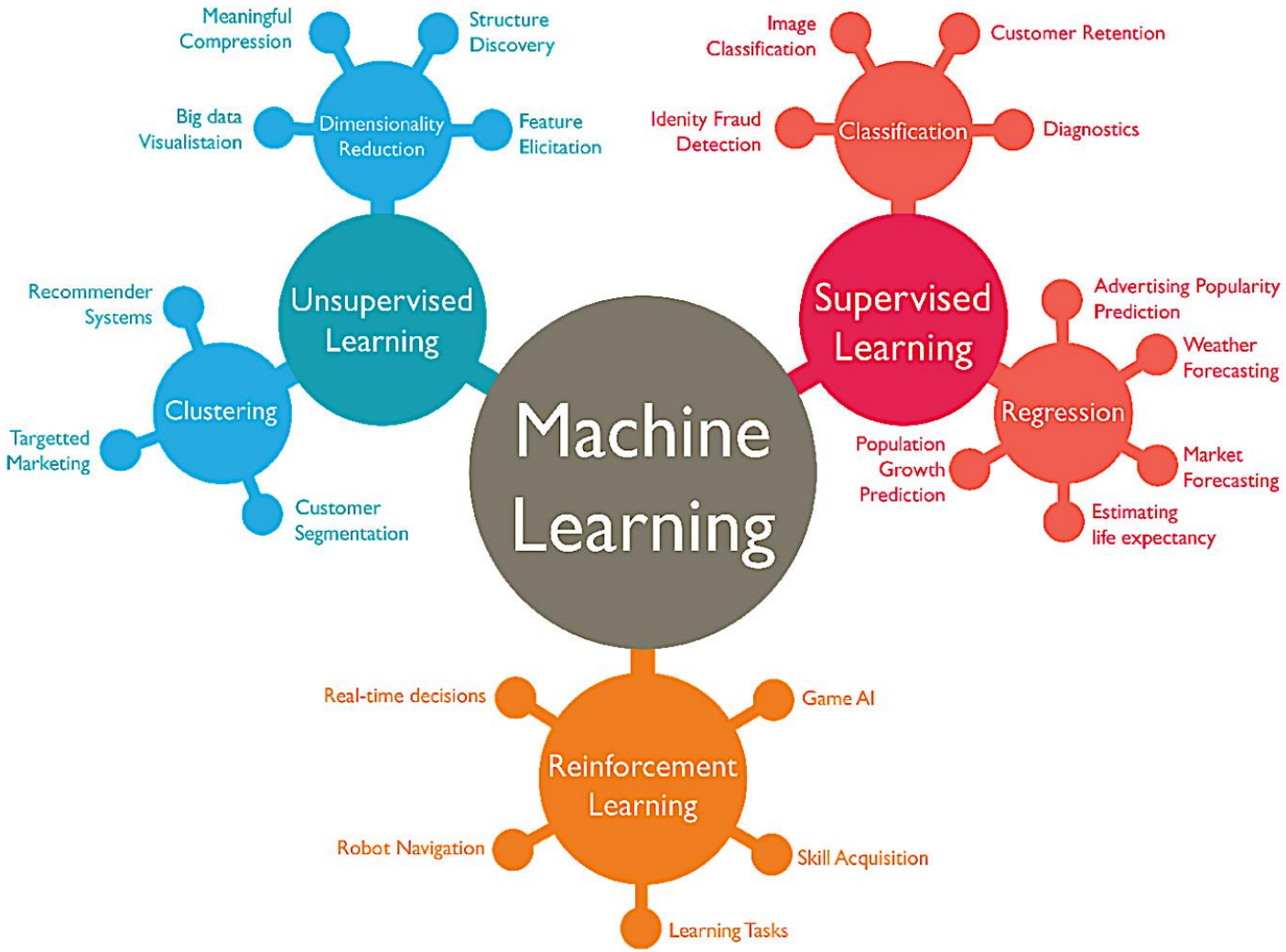
Bütün hüquqlar qorunur.

Machine Learning Algorithms

- Supervised Learning
  - Regressions – Linear Regression, Support Vector Regressor, Decision Trees and Random Forests
  - Classifiers – Logistic Regression, Support Vector Machines, KNN, Naïve Bayes, Decision Trees and Random Forests
- Unsupervised Learning
  - Clustering – K-Means and many more
- Reinforcement Learning
  - Model-free RL
  - Model-based RL

Bütün hüquqlar qorunur.

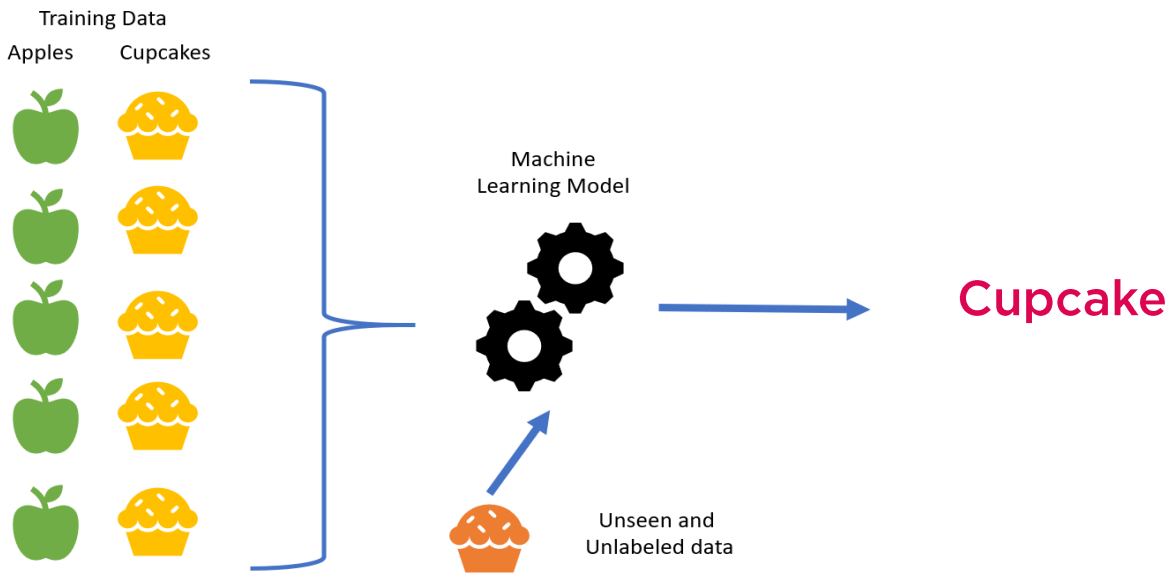
Types of Machine Learning



Bütün hüquqlar qorunur.

Supervised Learning

- Supervised learning is by far the most popular form of AI and ML used today.
- We take a set of **labeled data** (called a dataset) and we feed it into some ML learning algorithm that then creates a model to fit this data to some outputs;
- E.g., We have Apples and Cupcakes in our dataset. Our algorithm learns the specifications of both the classes and now can predict the class of a new coming object



Bütün hüquqlar qorunur.

Supervised Learning In Business

- In the Business world a lot of Machine Learning Models fall under this type. Some examples are:
  - Predicting which customers are most likely to leave our business (churn)
  - Predicting who might default on a loan
  - Predicting prices based on a set of information, e.g. predicting Airbnb prices based on location apartment size, number of persons it can accommodate etc.

Bütün hüquqlar qorunur.

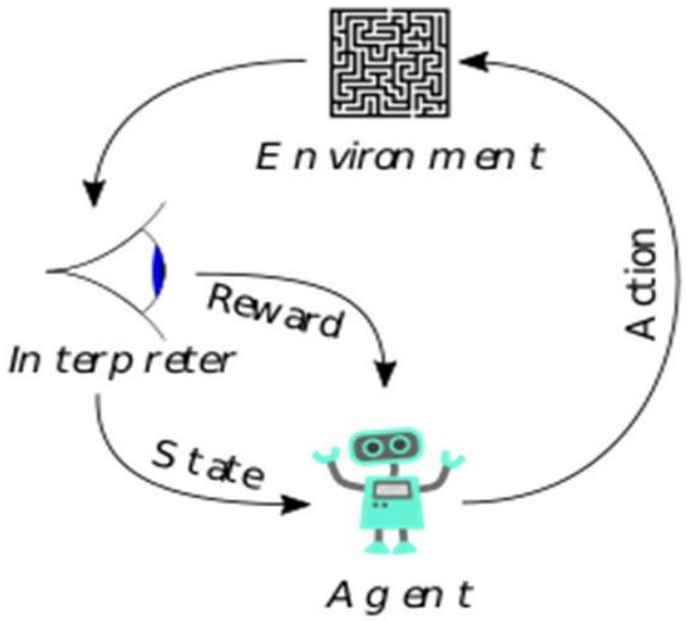
Unsupervised Learning

- Unsupervised learning is concerned with finding interesting clusters of input data. It does so **without any help of data labeling.**
- It does this by creating interesting transformations of the input data
- It is very important in data analytics when trying to understand data
- Examples in the Business world
  - Customer Segmentation

Bütün hüquqlar qorunur.

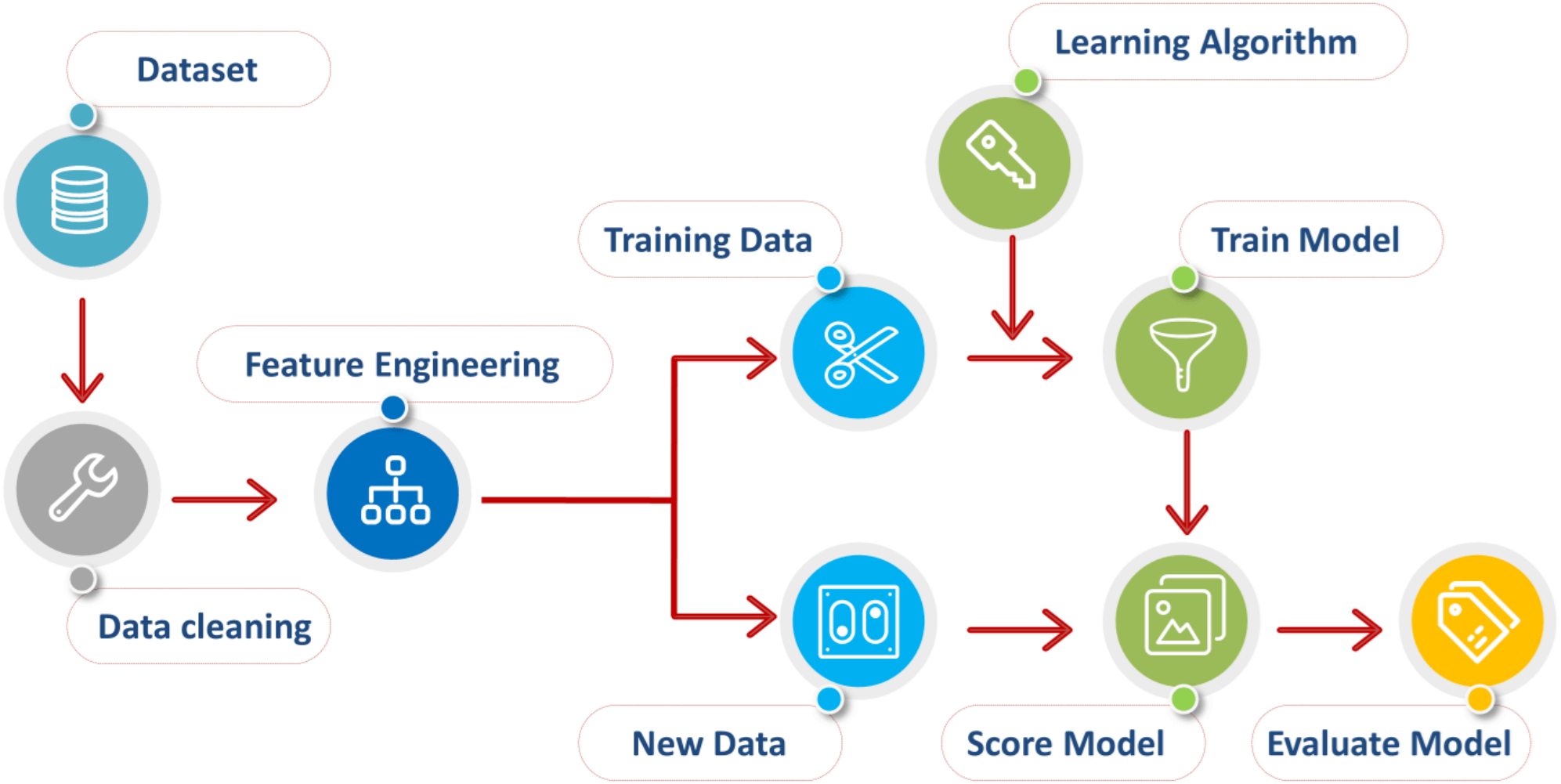
Reinforcement Learning

- Reinforcement learning is a type of learning where an agent learns by receiving rewards and penalties.
- Unlike Supervised Learning, it isn't given the correct label or answer. It is learning based on experience.
- Usual applications are AI playing games(e.g., DeepMind's Go AI)



Bütün hüquqlar qorunur.

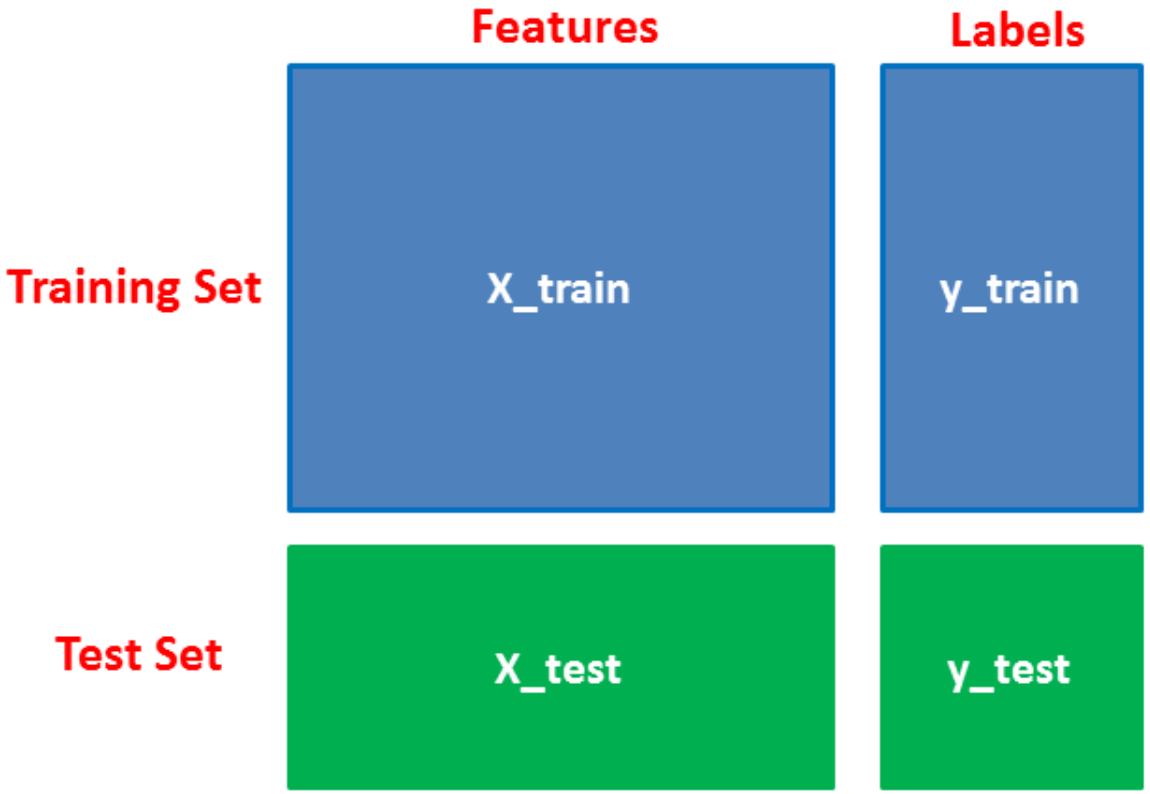
Machine Learning Process



Bütün hüquqlar qorunur.

Evaluation matrix

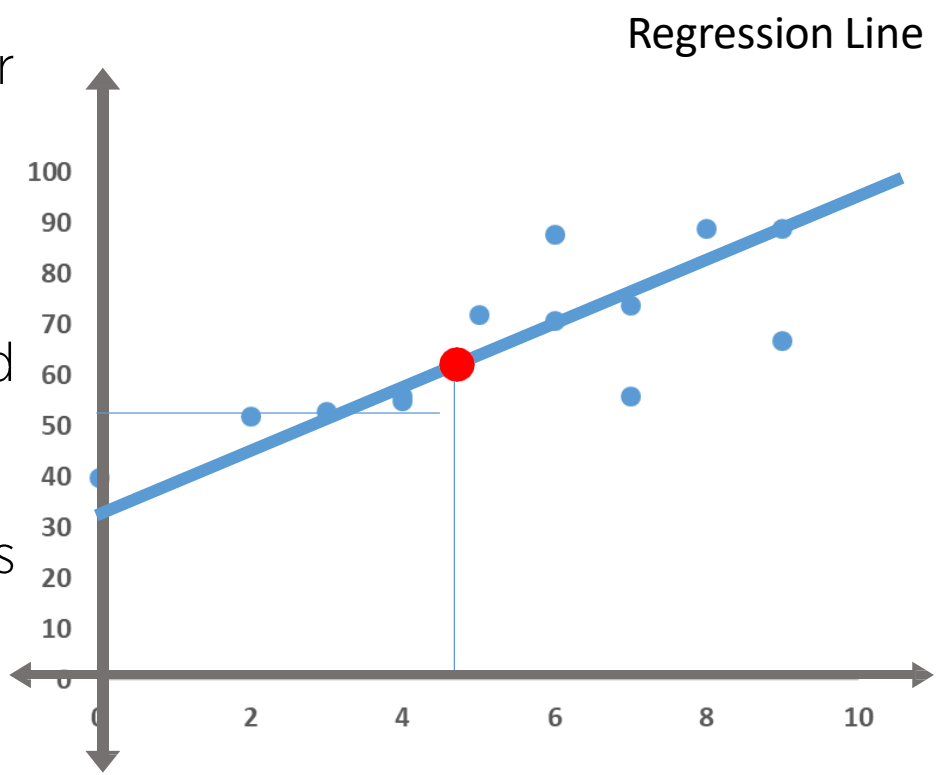
- Regression
  - R^2
  - RMSE
- Classification
  - Precision
  - Recall
  - F-1 Score



Bütün hüquqlar qorunur.

Regression Analysis

- Regression Analysis is a statistical process for estimating the relationships among variables
- The predictor is a continuous variable
- Relationship between a dependent variable and one or more independent variables (or 'predictors')
- Can also be used to infer causal relationships between dependent and independent variables.



Dependent variable

Population Y intercept

Population Slope Coefficient

Independent variable

Random Error term

$$Y_i = \underbrace{\beta_0 + \beta_1 X_i}_{\text{Linear component}} + \underbrace{\varepsilon_i}_{\text{Random Error component}}$$

Bütün hüquqlar qorunur.

Simple and Multiple Linear Regression

- Simple Regression :
  - Only one Dependent
  - Only one Independent

$$y = b_0 + b_1 x_1$$

- Multiple Linear Regression :
  - Only one Dependent
  - Many Independent

predictor, ‘x-variable’. independent variable. explanatory variable

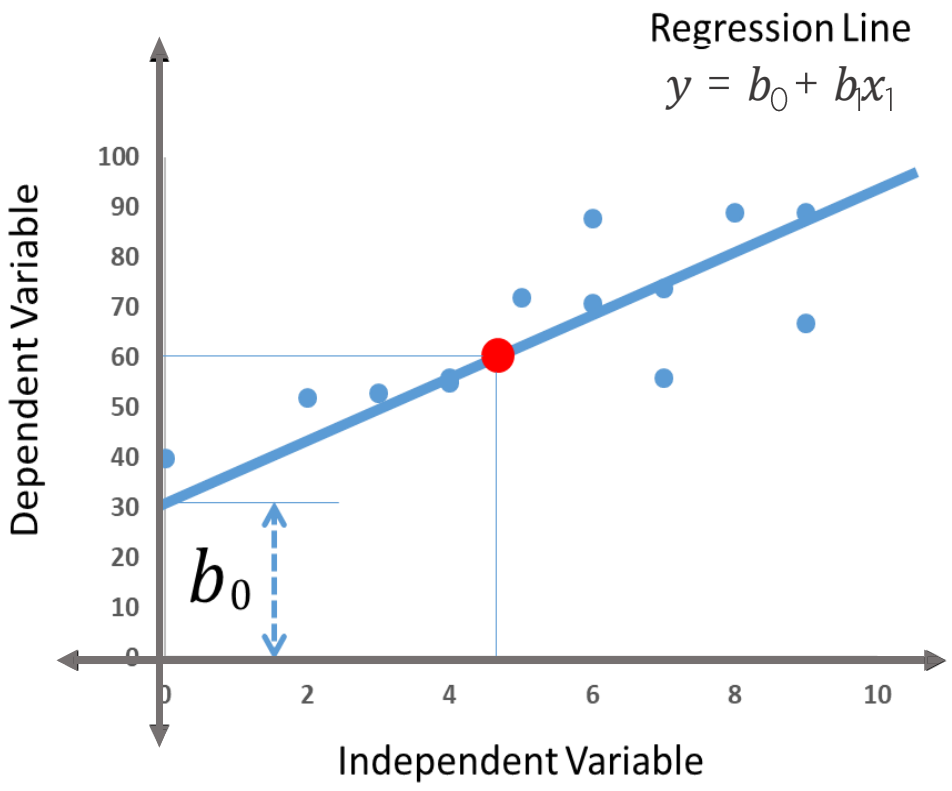
coefficient

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \varepsilon$$

response, dependent variable, observation, ‘y-variable’

linear predictor

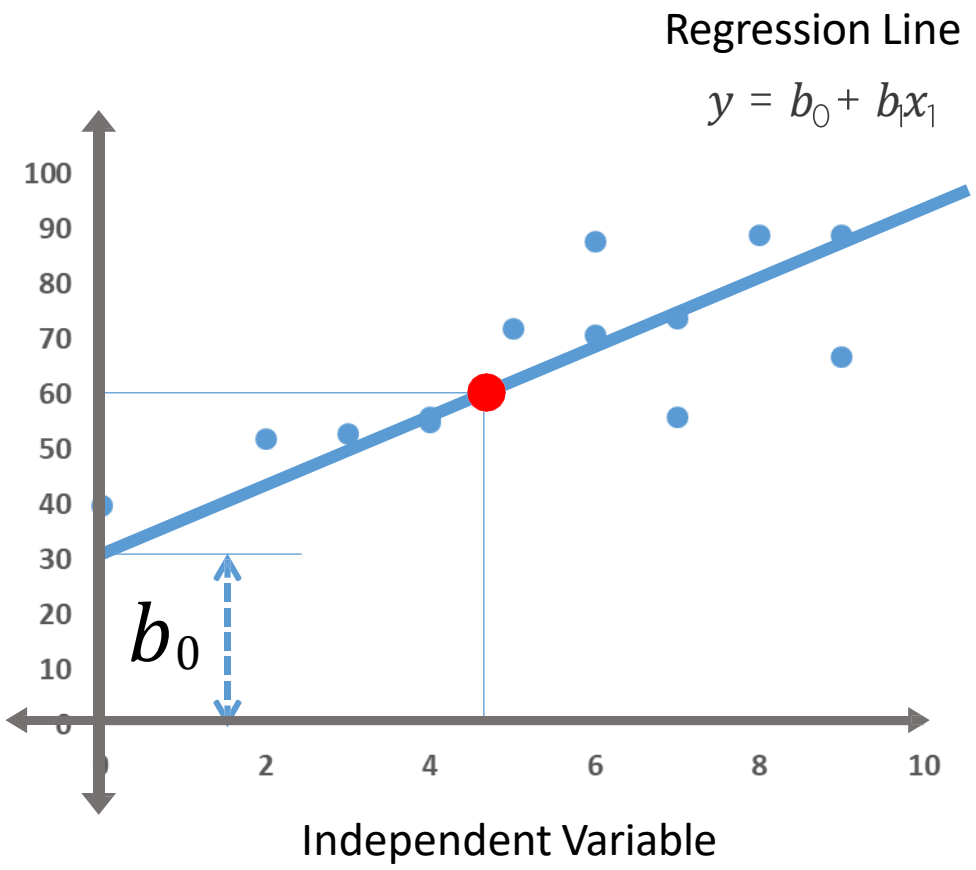
random error, “noise”



Bütün hüquqlar qorunur.

Simple Linear Regression

- A Linear Regression is a statistical approach to modeling the relationship between a dependent variable (y) and one or more independent variables (x).
- Basically, we want to know the regression equation that can be used to make predictions.
- Our model uses linear predictor functions whose parameters (similar to the k and b in 'y=kx+b') are estimated using the training data.
- Linear Regressions are one of the most popular ML algorithms due to their simplicity and ease of implementation.



Bütün hüquqlar qorunur.

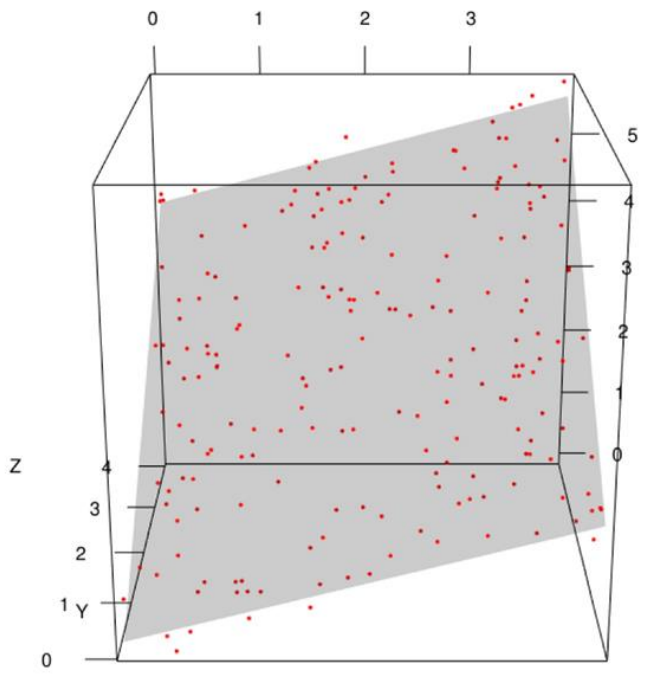
Multiple linear regression

- For a simple regression linear model, a **straight line** expresses y as a function of x.

- Now we have an additional dimension (z). What will dimension is added to a line? **It becomes a plane.**

Let's denote z as x_2, so the formula becomes:

$$y = k_1 x + k_2 x_2 + b_0$$



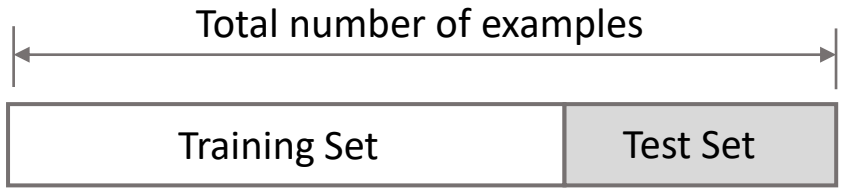
Bütün hüquqlar qorunur.

Train Test Splits

- Split arrays or matrices into random train and test subsets
- Training set contains a known output, and the model learns on this data in order to be generalized to other data later on.
- Test dataset in order to test our model's prediction on this subset.

In [ ]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=None)
```



Bütün hüquqlar qorunur.

Scikit Learn

- Scikit-learn is probably the most useful library for machine learning in Python. It is built on NumPy, SciPy and matplotlib and supports various models including regressions, classification and clustering

Used for modeling instrument such as:

- Feature extraction: for defining attributes in image and text data.
- Feature selection: for identifying meaningful attributes from which to create supervised models.
- Supervised Models: a vast array not limited to generalized linear models, discriminate analysis, naive bayes, lazy methods, neural networks, support vector machines and decision trees.

Bütün hüquqlar qorunur.

Train and Test Splitting in Linear Regression

- After EDA, we split 70%-80% of the data to the training set while 30%-20% of the data to test set using below code.

- The test_size variable is where we actually specify the proportion of the test set.

In [ ]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=None)
```


Bütün hüquqlar qorunur.

Fitting in Linear regression

- Time is to train our algorithm. For that, we need to import LinearRegression class, instantiate it, and call the *fit()* method along with our training data.

In [ ]:

```
regressor = LinearRegression()
regressor.fit(X_train, y_train) #training the algorithm
```


Bütün hüquqlar qorunur.

Intercept and Coefficient in Linear regression

- Linear regression model basically finds the best value for the intercept and slope, which results in a line that best fits the data.

- To see the value of the intercept and slope calculated by the linear regression algorithm for our dataset, execute the following code:

In [ ]:

```
#To retrieve the intercept:
print(regressor.intercept_)

#For retrieving the slope:
print(regressor.coef_)
```


Bütün hüquqlar qorunur.

Prediction in Linear regression

- We will use our test data and see how accurately our algorithm predicts the percentage score.
- To make predictions on the test data, execute the following script:

In [ ]:

```
y_pred = regressor.predict(X_test)
```


Bütün hüquqlar qorunur.

III
Model Evaluation

Regression Model Evaluation

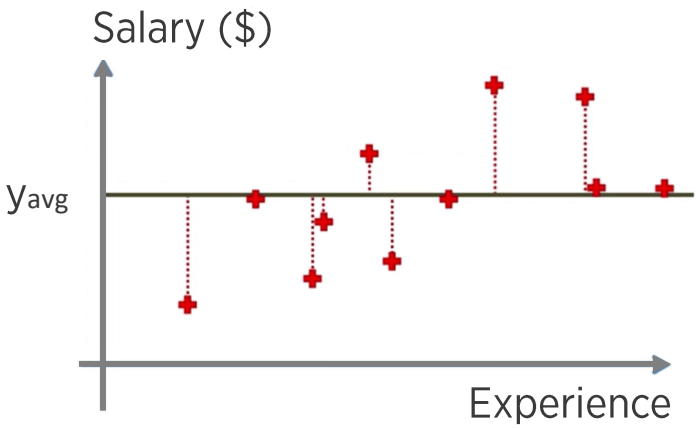
- Evaluation methods:
  - R^2
  - Adjusted R^2
  - Mean Absolute Error
  - Mean Squared Error
  - Root Mean Squared Error
  - Cross Validation

Bütün hüquqlar qorunur.

R^2

- R-squared is a statistical measure of how close the data are to the fitted regression line.
- It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression.
- R-squared is always between 0 and 100%:
- 0% indicates that the model explains none of the variability of the response data around its mean.
- 100% indicates that the model explains all the variability of the response data around its mean.
- The higher the R-squared, the better the model fits your data.

Simple Linear Regression:



$$SS_{res} = SUM(y_i - \hat{y_i})^2$$

$$SS_{tot} = SUM(y_i - y_{avg})^2$$

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

Bütün hüquqlar qorunur.

Adjusted R^2

- The **adjusted R-squared** is a modified version of **R-squared** that has been **adjusted** for the number of predictors in the model.
- The **adjusted R-squared** increases only if the new term improves the model more than would be expected by chance.
- R2 always increases when the number of variables increases.
- If you add a useless variable to you model, your R2 will still increase.
- To balance it, you should always compare models with different number of independent variables with adjusted R2.

$$Adjusted\ R^2 = 1 - \frac{(n-1)}{[n-(k+1)]}(1-R^2)$$

Formula 9-6

Where n= sample size
k=number of independent (x) variables

Bütün hüquqlar qorunur.

Mean Absolute Error

- The absolute error is the absolute value of the difference between the forecasted value and the actual value.
- The **Mean Absolute Error(MAE)** is the <u>average</u> of all absolute errors.
- MAE tells us how big of an error we can expect from the forecast on average.

$$s = \sqrt{\frac{\sum(x_i - \bar{x})^2}{N - 1}}$$

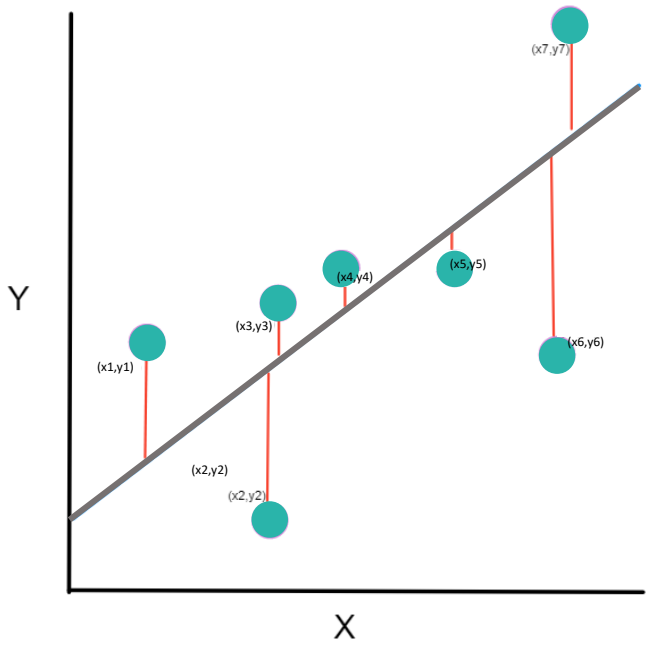
- Σ is the summation symbol
- x_i is the ith value
- $\bar{x}$ is the sample mean
- N is the sample size

Bütün hüquqlar qorunur.

Mean Squared Error

- The **mean squared error (MSE)** of an estimator measures the average of the squares of the errors.
- MSE is a risk function, corresponding to the expected value of the squared error loss.

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



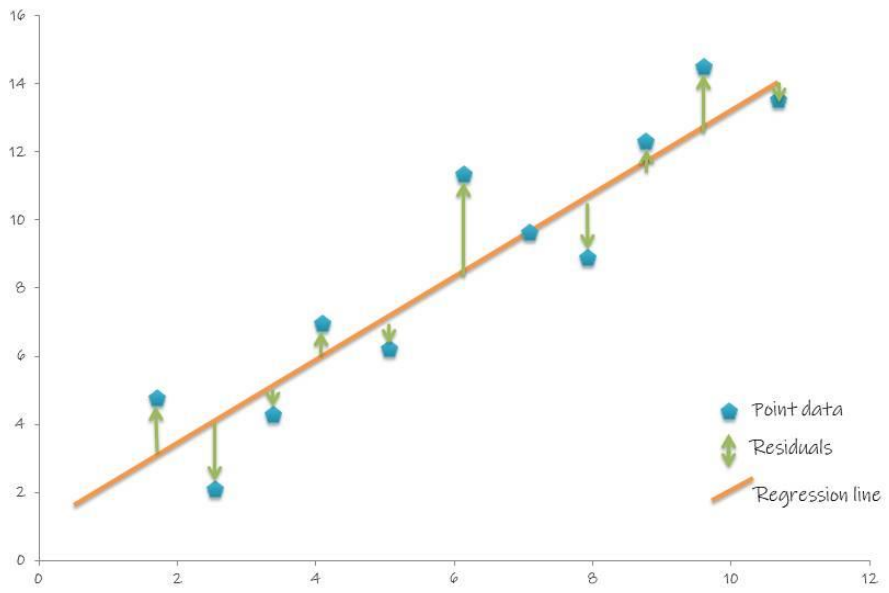
Bütün hüquqlar qorunur.

Root Mean Squared Error

- Root mean squared error (RMSE) is the square root of the mean of the square of all of the error
- The parameter indicates the standard deviation of the residuals or how far the points are from the regression or modelled line

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f_i - o_i)^2}$$

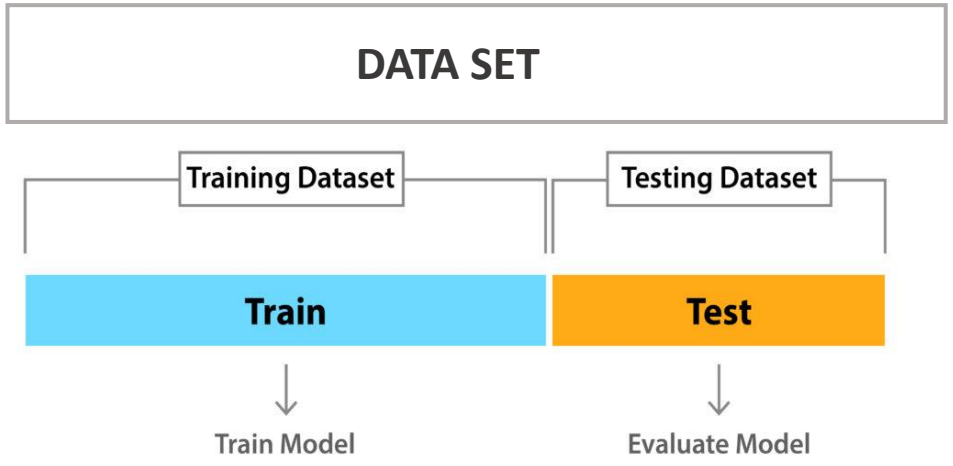
- n: number of samples
- f: forecasts
- o: observed values

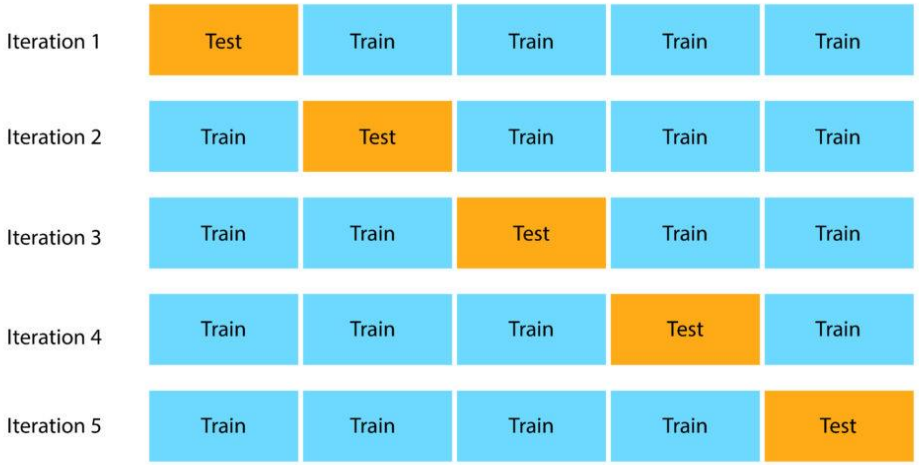


Bütün hüquqlar qorunur.

Cross Validation

- **Cross validation** is a statistical method used to estimate the performance (or accuracy) of machine learning models.
- Types of Cross Validation:
  - Holdout method
  - K fold cross validation





Bütün hüquqlar qorunur.

Evaulation of Model in Linear Regression

- This step is particularly important to compare how well different algorithms perform on a particular dataset. For regression algorithms, these evaluation metrics are commonly used:

In [ ]:

```
R2 = metrics.r2_score(y_test, y_pred)
n = X_test.shape[0] #sample size
p = X_test.shape[1] #number of predictors
print('R^2:' , R2)
print('Adjusted R^2 :'  , 1 - metrics.r2_score(y_test, y_pred))
print('Mean Absolute Error:' , metrics.mean_absolute_error(y_test, y_pred))
print('Mean Squared Error:' , metrics.mean_squared_error(y_test, y_pred))
print('Root Mean Squared Error:' , np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```


Bütün hüquqlar qorunur.

Model Formulation Example

- Fernando wants to get best car with best price.
- Fernando reaches out to his friend for more data.
- He asks him to provide more data on other characteristics of the cars.

make	fuelType	nDoors	driveWheels	engineSize	horsePower	peakRpm	cityMpg	price
alfa-romero	gas	two	rwd	130	111	5000	21	13495
alfa-romero	gas	two	rwd	130	111	5000	21	16500
alfa-romero	gas	two	rwd	152	154	5000	19	16500
bmw	gas	four	rwd	164	121	4250	21	21105
bmw	gas	four	rwd	164	121	4250	20	24565
bmw	gas	four	rwd	209	182	5400	16	30760
bmw	gas	two	rwd	209	182	5400	16	41315
bmw	gas	four	rwd	209	182	5400	15	36880

Bütün hüquqlar qorunur.

Model Formulation Example

- The multiple regression model that he formulates is:
- Estimate price as a function of engineSize, horsePower, peakRPM, length, width and height.

price = f(engineSize, horsePower, peakRPM, length, width, height)

In [ ]:

```
price = β0 + β1.engine size + β2.horsepower + β3. peak RPM + β4.length+ β5.width +
β6.height
```

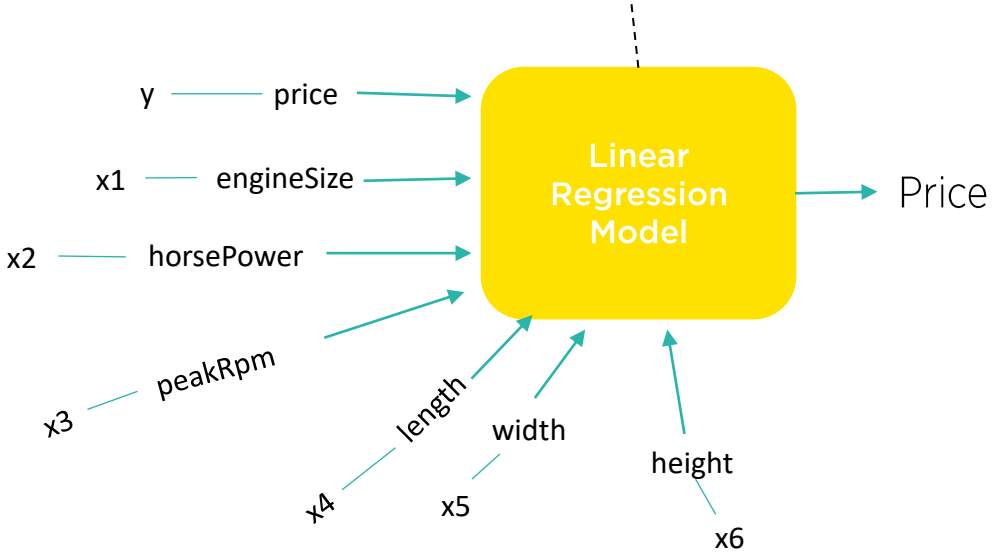
Model Formulation Example

- Fernando inputs these data into his statisticalpackage.
- The package computes parameters. The output is following:

In [ ]:

```
price = -85090 + 102.85 * engineSize + 43.79 * horsePower + 1.52 * peakRpm - 37.91 * length + 908.12 * width+ 364.33 * height
```

- Estimates β0 β1 β2 β3 β4 β5 β6
- Creates model performance metrics



	Estimate
(Intercept)	-85086.39
engineSize	102.85
horsePower	43.79
peakRpm	1.52
length	37.91
width	908.12
height	364.33

Bütün hüquqlarqorunur.

Model Formulation Example

- The interpretation of multiple model provides impact of each independent variable on the dependent variable (target).
- Remember, the equation provides an estimation of average price. Each coefficient is interpreted with all other predictors held constant.

Bütün hüquqlarqorunur.

Interpret the coefficients

- **Engine Size:** With all other predictors held constant, if the engine size is increased by one unit, the average price increases by $102.85
- **Horse Power:** With all other predictors held constant, if the horsepower is increased by one unit, the average price increases by $43.79
- **Peak Rpm:** With all other predictors held constant, if the peak RPM is increased by one unit, the average price increases by $1.52
- **Length:** With all other predictors held constant, if the length is increased by one unit, the average price decreases by $37.91
- **Width:** With all other predictors held constant, if the width is increased by one unit, the average price increases by $908.12
- **Height:** With all other predictors held constant, if the height is increased by one unit, the average price increases by $364.33

Bütün hüquqlarqorunur.

Model Summary

- Coefficients
  - All coefficients are greater than zero. This implies that all variables have an impact on the average price.
- p-value:
  - The probability of observing the p- value purely by chance is quite low for all of the variables except for length. The p-value for length is 0.4854. This implies that probability that the observed t-stat is by chance is 48.54%. This number is quite high

$\beta_0 \ldots \beta_6$ t-stat p-value

```
Coefficients:
             Estimate Std. Error t value     Pr(>|t|)
(Intercept) -85086.39   15265.49   -5.57 0.00000012266 ***
engineSize     102.85      15.38    6.69 0.00000000049 ***
horsePower      43.79      16.41    2.67       0.0085 **
peakRpm          1.52       0.72    2.11       0.0367 *
length         -37.91      54.19   -0.70       0.4854
width          908.12     282.27    3.22       0.0016 **
height         364.33     153.36    2.38       0.0189 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3300 on 141 degrees of freedom
  (2 observations deleted due to missingness)
Multiple R-squared:  0.818,     Adjusted R-squared:  0.811
F-statistic:  106 on 6 and 141 DF,  p-value: <0.0000000000000002
```

coefficient of determiniation (R^2) Adjusted coefficient of determiniation (R^2)

Bütün hüquqlarqorunur.

Model Summary

- Recall the discussion of how R-squared help to explain the variations in the model. When more variables are added to the model, the r-square will not decrease. It only increases. However, there has to be a balance. Adjusted R-squared strives to keep that balance.

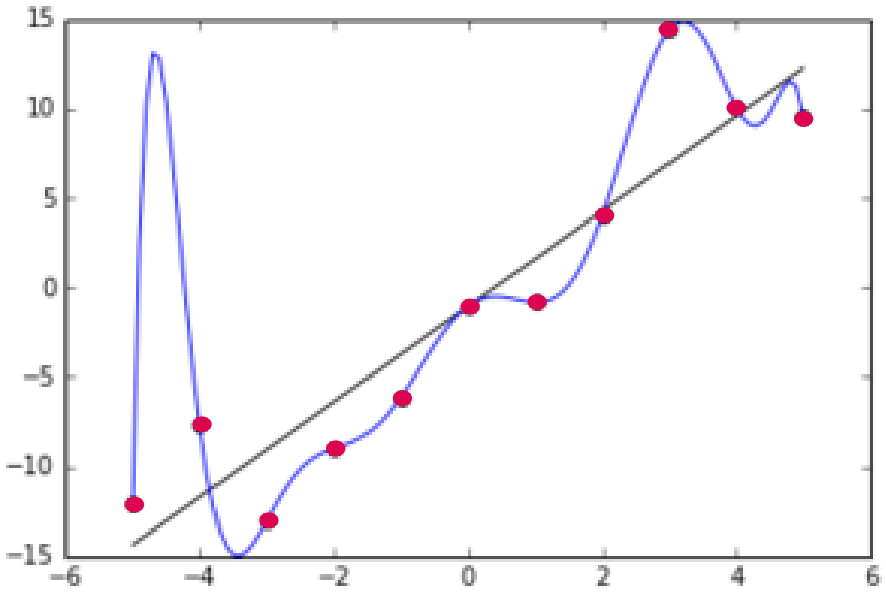
- The adjusted R-squared is a modified version of R-squared that has been adjusted for the number of predictors in the model. The adjusted R-squared compensates for the addition of variables and only increases if the new term enhances the model.

- Adjusted R-squared: The r-squared value is 0.811. This implies that the model can explain 81.1% of variations seen in training data. It is better than the previous model (75.03%).

Bütün hüquqlarqorunur.

Overfitting

- Overfitting means that the model we trained has trained “too well” and fit too closely to the training dataset.
- Techniques to reduce overfitting :
  - Increase training data.
  - Reduce model complexity.
  - Early stopping during the training phase (have an eye over the loss over the training period as soon as loss begins to increase stop training).
  - Ridge Regularization and Lasso Regularization
  - Use dropout for neural networks to tackle overfitting.



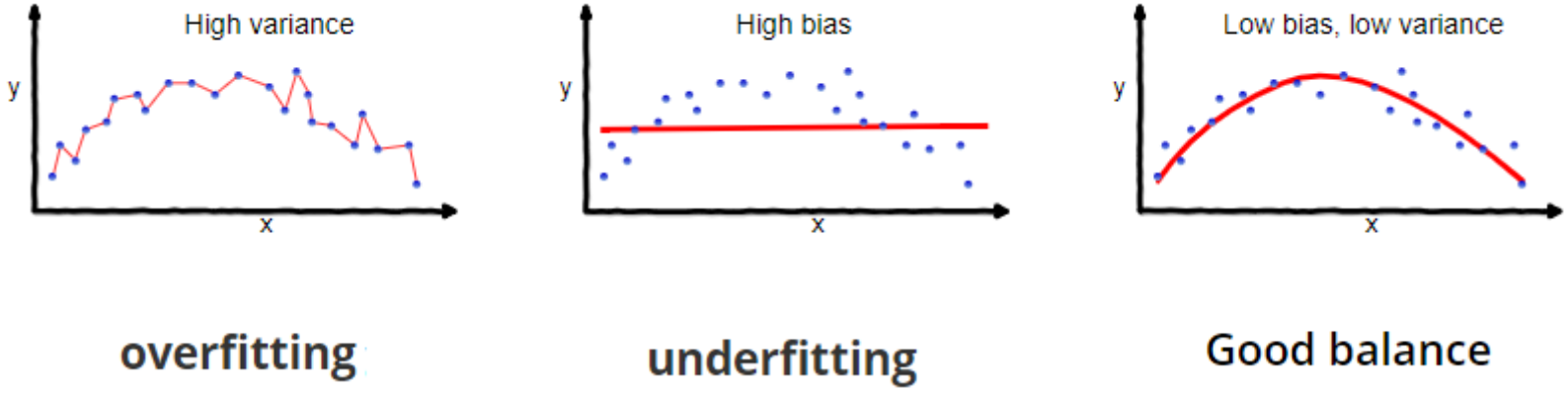
Bütün hüquqlar qorunur.

Underfitting

- Underfitting can happen when the model is too simple and means that the model does not fit the training data.

- Techniques to reduce underfitting:

  - Increase model complexity
  - Increase number of features, performing feature engineering
  - Remove noise from the data.
  - Increase the number of epochs or increase the duration of training to get better results.

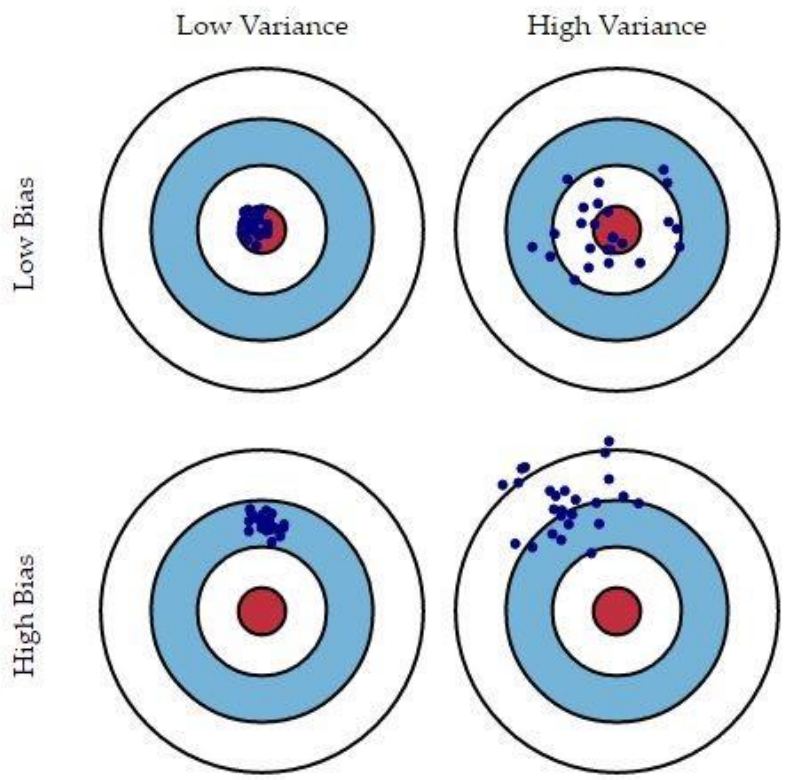


Bütün hüquqlar qorunur.

Bias & Variance

- Bias is the difference between the average prediction of our model and the correct value which we are trying to predict. Model with *high bias* pays very little attention to the training data and oversimplifies the model. It always leads to high error on training and test data.

- Variance is the variability of model prediction for a given data point or a value which tells us spread of our data. Model with high variance pays a lot of attention to training data and does not generalize on the data which it hasn't seen before. As a result, such models perform very well on training data but has high error rates on test data.



Bütün hüquqlar qorunur.

Thank You !