ML 1 Assignments

Daniel Rodinger

19. September 2024

1 Assignment 2

1.1 Regression

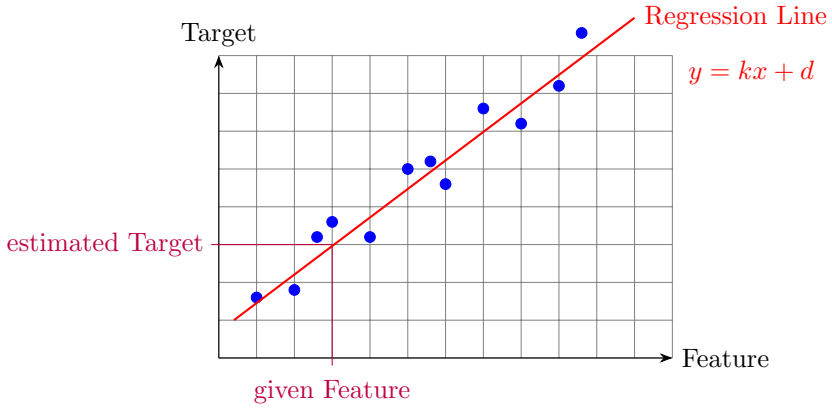


Figure 1: This figure illustrates a regression model applied to a dataset. The blue data points represent observations, such as the relationship between hours studied (Feature) and test scores (Target). The red regression line models this relationship, showing how test scores generally increase with more hours studied. The purple lines demonstrate an example where a specific number of study hours is the input. The expected test score can be estimated by projecting onto the regression line.

1.2 Classification

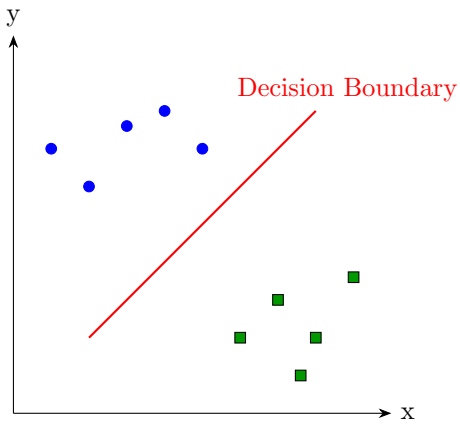


Figure 2: This figure depicts a classification problem involving two classes. Blue circles represent one class, such as emails labeled "Spam," and green squares represent another class, like "Not Spam." The red decision boundary separates the two classes based on features. The model uses this boundary to classify new emails as either Spam or Not Spam.

1.3 Features

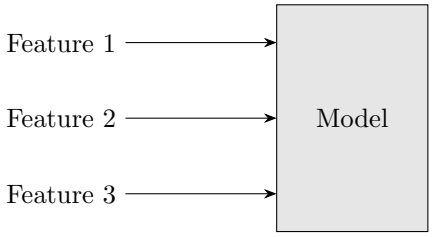


Figure 3: This diagram shows three features being input into a machine learning model. For example, in predicting house prices, Feature 1 could be the size of the house in square meters, Feature 2 the number of bedrooms, and Feature 3 the location. These features are fed into the model.

1.4 Targets

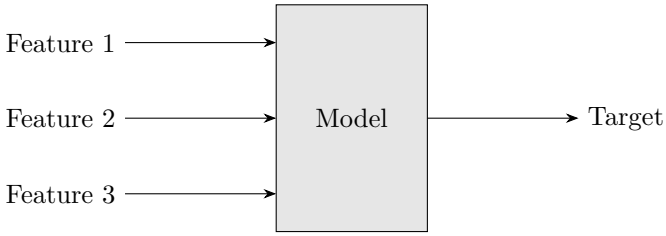


Figure 4: Extending figure 3, this figure illustrates the model producing a target output from the input features. After inputting the features, the model outputs a target value, such as the estimated price of the house in Euros.

1.5 Supervised Machine Learning Workflow

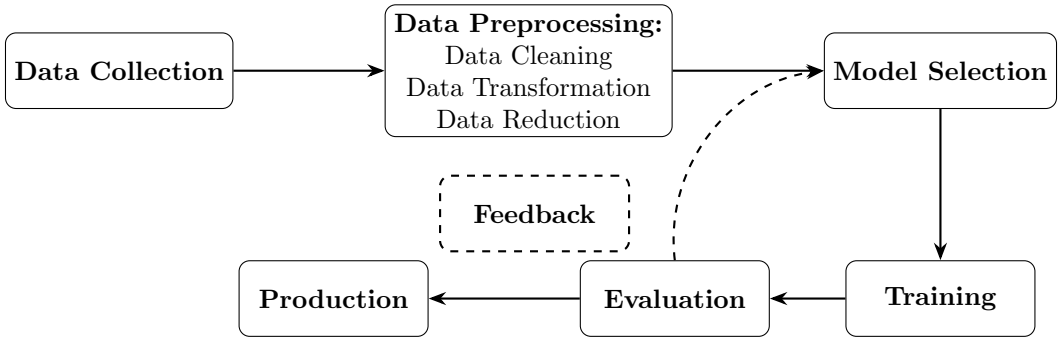


Figure 5: This flowchart represents the supervised machine learning workflow. The dashed feedback loop indicates that if the evaluation shows low accuracy, the model selection or training process may be revisited to improve performance.

The steps are:

1. **Data Collection**: Gather images labeled with the objects they contain (e.g., cats and dogs).
2. **Data Preprocessing**: Resize images and normalize pixel values.
3. **Model Selection**: Choose an appropriate model, like a convolutional neural network (CNN).

4. **Training**: Train the CNN using the labeled images.
5. **Evaluation**: Assess the model's accuracy in classifying new images.
6. **Prediction**: Use the trained model to classify unlabeled images as either cats or dogs.

2 Assignment 3

2.1 Train-Test Split

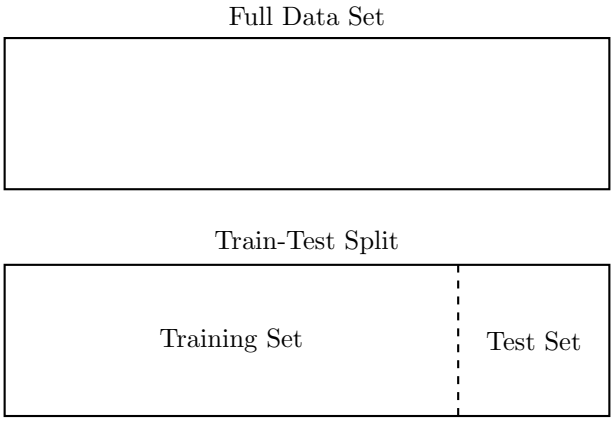


Figure 6: The Train-Test split is applied on data to split the test in training data and test data that is unseen to the model to validate it. A typical split ratio is 80% training data and 20% test data.

It is important to shuffle the data before processing because of many reasons. Some of the most imortant are:

- **Break order based patterns:** shuffling prevents models from learning unintended dependencies on data order (e.g. is often gathered in certain groups).
- **Reduce bias and overfitting:** ensure diverse data represantation in every batch of training data.
- **Generalization and effectiveness:** to ensure the model is able to learn meaningful patterns and perform well on unseen data .

2.2 Mean Absolute Error

The formula for Mean Absolute Error (MAE) is given by:

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

Where:

- n is the total number of observations.
- y_i is the actual value for the i-th observation.
- $\hat{y}_i$ is the predicted value for the i-th observation.
- $|y_i - \hat{y}_i|$ is the absolute difference between the actual and predicted values.

MAE measures the average magnitude of errors between the predicted values and actual values, treating all errors equally. For example if a model wants to predict the location of an object in an image. The actual coordinates of the top-left corner are $(50, 100)$, while the predicted coordinates are $(55, 90)$.
To calculate the Mean Absolute Error (MAE):

- **Actual Coordinates**: $(50, 100)$
- **Predicted Coordinates**: $(55, 90)$

- Absolute error for the x-coordinate: $|50 - 55| = 5$
- Absolute error for the y-coordinate: $|100 - 90| = 10$

The MAE for this prediction is:

$$MAE = \frac{5 + 10}{2} = 7.5$$

This means the average error in object location prediction is 7.5 pixels.
While MAE treats all errors equally by calculating the average of absolute differences and therefore makes it less sensitive to outliers, the Root Mean Squared Error (RMSE) squares the errors before averaging, giving more weight to larger errors. This means RMSE is more sensitive to outliers, as large deviations are amplified.

2.3 Accuracy

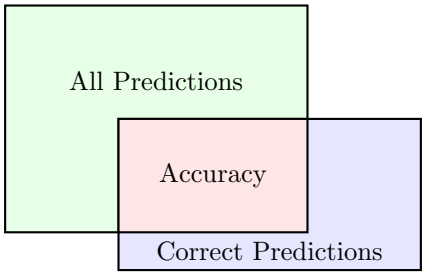


Figure 7: This diagram shows the overlap between all predictions (green) and correct predictions (blue), with the overlapping area corresponding to the accuracy. The bigger the overlap is, the higher the model's accuracy. Very sensitive to class imbalance (more positives than negatives).

$$Accuracy = \frac{Correct\,Predictions}{All\,Predictions} = \frac{TP + TN}{TP + TN + FP + FN}$$

2.4 Confusion Matrix (Snippet)

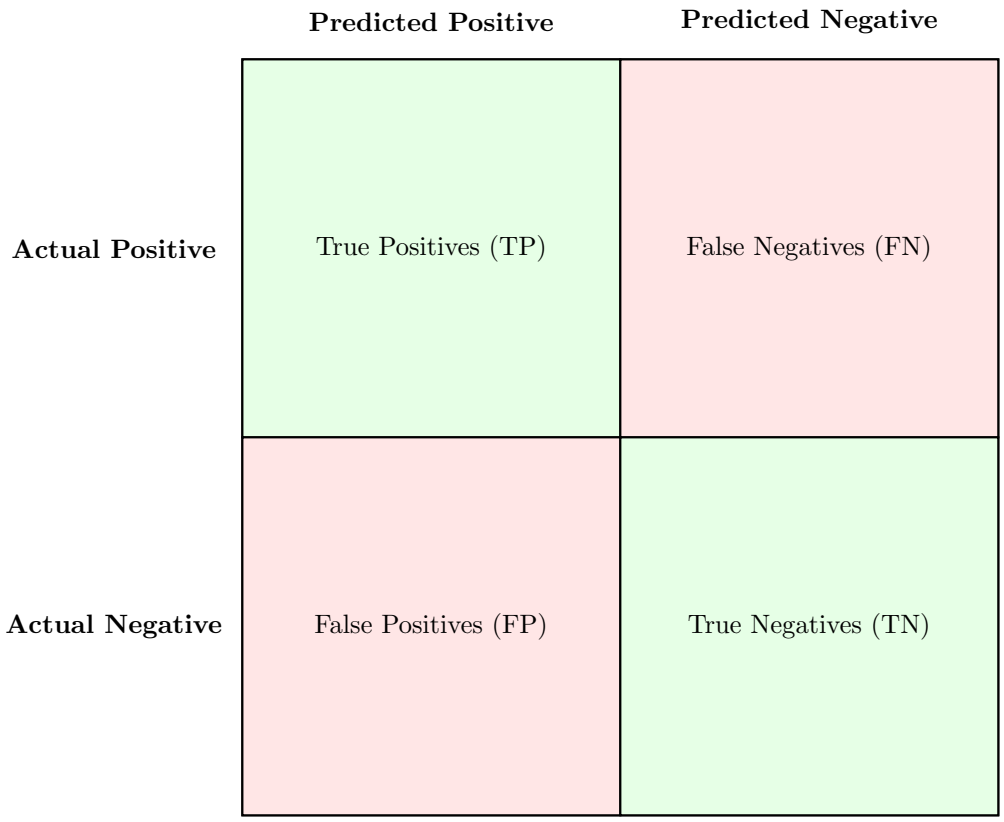


Figure 8: Confusion Matrix is a table to evaluate the performance of a model by comparing predicted labels with actual labels. The rows are actual and the columns are predicted labels

The steps are:

1. **True Positives**: Correctly predicted positive cases (e.g., cars that are labeled as cars).
2. **False Positives**: Incorrectly predicted positive cases (e.g., buses predicted as cars).
3. **True Negatives**: Correctly predicted negative cases (e.g., buses predicted as not cars).
4. **False Negatives**: Incorrectly predicted negative cases (e.g., cars predicted as not cars).

2.4.1 True Positive Rate (Sensitivity)

The True-Positive Rate (TPR) measures the correctly identified labels. TPR is crucial when missing positive cases is costly (e.g. stop sign detection for self driving car). Not sensitive to True Negatives. It is defined by:

$$TPR = \frac{TP}{TP + FN}$$

2.4.2 True Negative Rate (Specicivity)

The True-Negative Rate measures the proportion of actual negatives correctly identified. A high TNR is crucial when avoiding false positives is crucial (e.g. marking credit card transactions as non fraud). It is defined by:

$$TNR = \frac{TN}{TN + FP}$$

3 Assignment 4

Explain “hyper parameters“ and “K nearest neighbours“ in your own words.

3.1 Hyperparameters

((a)) Stride example 1

((b)) Stride example 2

Figure 9: Examples of stride operations.

Hyper parameters are external configuration variables to manage machine learning training. They are determined through tests and are not obtained from the actual data. They determine how a model learns and are therefore critical for fine-tuning. Examples include:

- Stride: Step size of filter
- Filter Size: Size of sliding window
- Learning rate: Step size towards minimum of loss function in one iteration
- Batch size: Number of training samples used in one iteration

3.2 K-Nearest-Neighbours

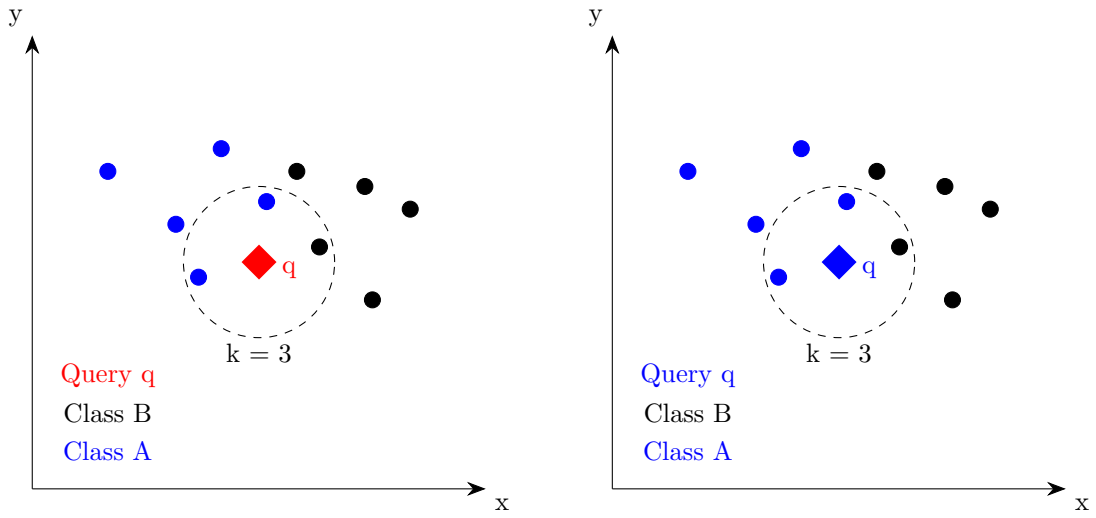


Figure 10: Query point q before and after performing kNN

K-nearest-neighbours (KNN) is a machine learning algorithm. it classifies points based on majority vote of it's k nearest neighbours. K and the distance metric are hyper-parameters the output depends heavily on it. KNN works works by calculating the distance from each point in the dataset to the query point (lazy learning).

- k: The value of k significantly affects performance. A small K is sensitive to noise, while a large K might over-generalize.
- Distance Metric: is used to calculate how close two points are (e.g Manhattan, Euclidian)

4 Assignment 5

4.1 Missing Data

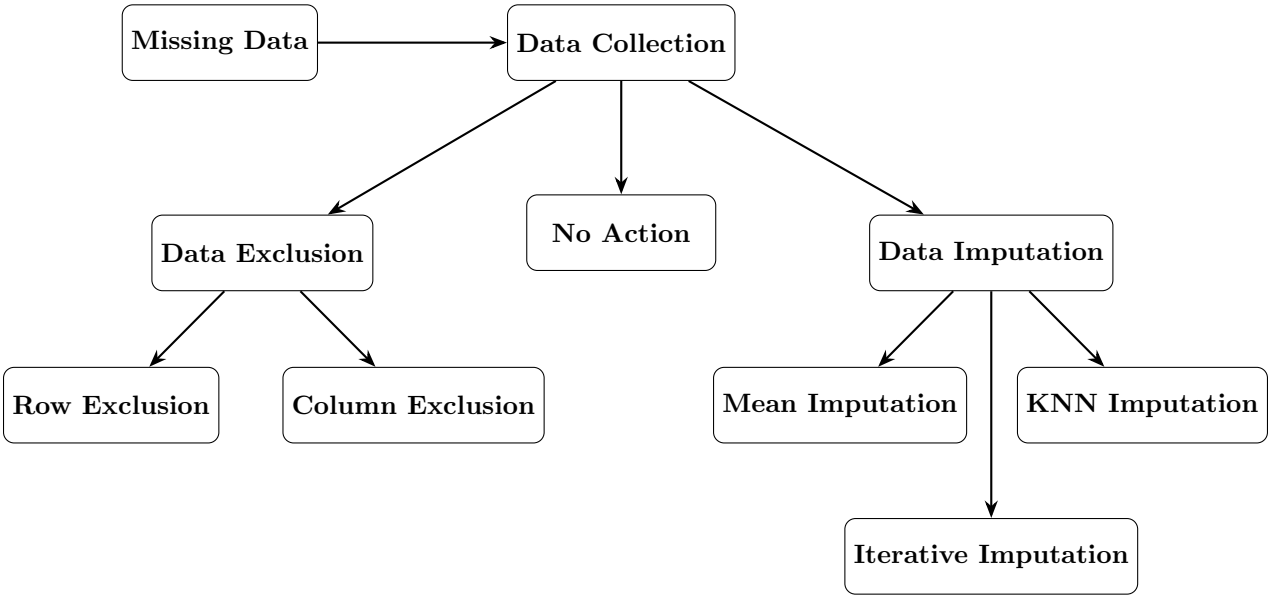


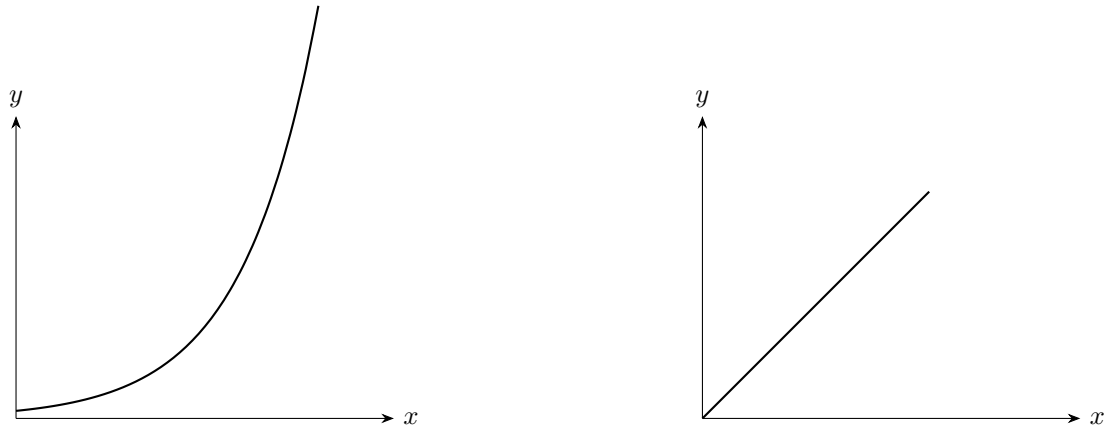
Figure 11: Methods for Handling Missing Data

When dealing with missing data there are several strategies for different scenarios. They can broadly be categorized into collection, exclusion, imputation:

- **Data Collection**: best way to adress missing data is to collect it through further data gathering
- **Data Exclusion**: if collection more data is not feasable either **rows** or **cols** can be dropped. But this can introduce bias.
- **Data Imputation**: missing values can be filled by:
  - **Mean Imputation**: fill with mean value of the column.
  - **Iterative Imputation**: use a regression model to to predict the missing values.
  - **KNN Imputation**: use knn algorithm to predict missing values based on similar data points.
- **No Action**: in some case algorithms can handle missing values.

5 Assignment 6

5.1 Transformation



((a)) Exponential Distribution

((b)) Distribution after Transformation

Figure 12: Exponential Distribution and Log Transformed Line Side by Side

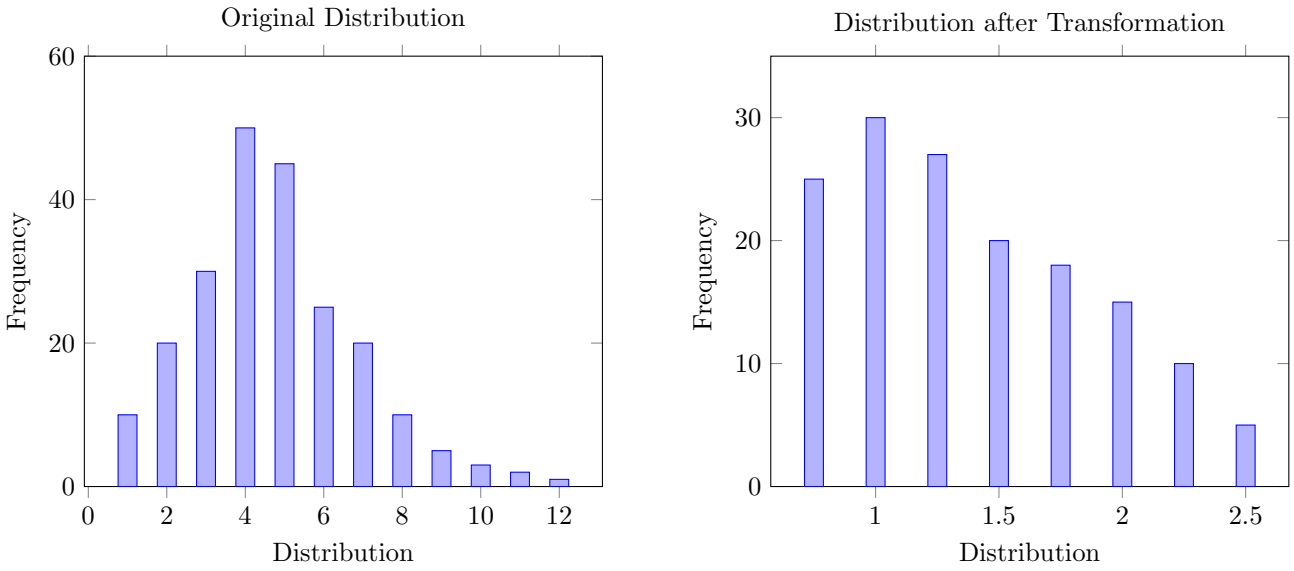


Figure 13: Original and Log-Transformed Feature Distributions

After transformation distribution becomes more normalized and the range of values is compressed.

5.2 One-Hot Encoding

Original Data		One-Hot Encoding		
Season		Autumn	Winter	Spring
Autumn	→	1	0	0
Winter		0	1	0

Figure 14: Normalization of Data

5.3 Handling Outliers

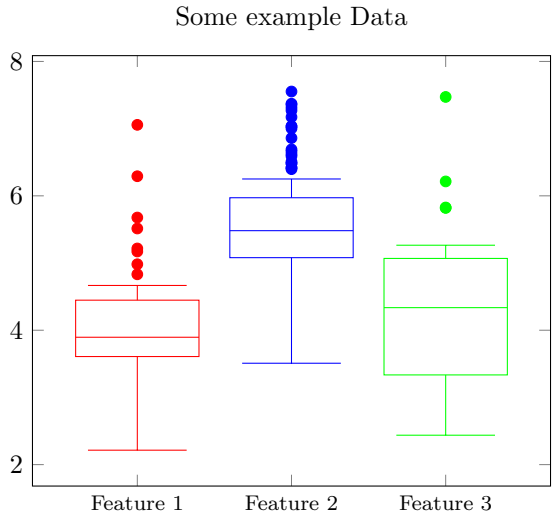


Figure 15: Bxoplots with Outliers

Outliers are observations in a given dataset that lie far away from the rest of the observations. They may occur due to variability in data or error. The following table shows the effects of outliers on three important measures to describe data (Mean, Median and Mode). The median is the middle value when a data set is ordered from least to greatest. The mode is the number that occurs most often in a data set.

	with outlier	without outlier
Mean	20.08	12.72
Median	14.0	13.0
Mode	15	15
Variance	614.74	21.28
Std dev	24.79	4.61

Table 1: Statistical Comparison: With and Without Outliers

5.4 Class Imbalance

Class imbalance occurs when the dataset has an unequal distribution of classes. This can lead to bias. Resampling is a technique to address class imbalance in machine learning by adjusting the class distribution in the training set. Oversampling increases the minority class, while undersampling reduces the majority class. Both are effective but come with downsides:

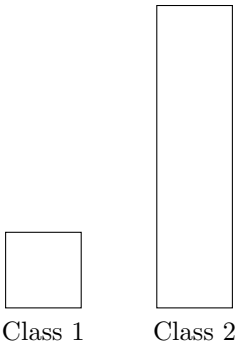


Figure 16: Class Imbalance

- **Oversampling**: Duplicating minority class samples can cause the model to overfit, learning patterns specific to those duplicated instances rather than generalizable ones.
- **Undersampling**: Reducing majority class samples can result in the loss of valuable information, causing the model to miss important patterns and generalize poorly.

5.5 Feature Selection

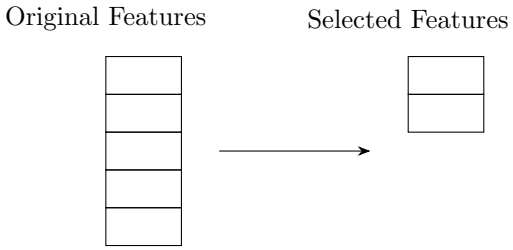


Figure 17: Feature Selection

Feature selection is crucial for improving model performance, reducing overfitting, and enhancing interpretability. It reduces the dimensionality of input data by removing irrelevant or redundant features.

- **Enhanced Model Performance**: Selecting relevant features leads to more accurate and efficient models.
- **Reduced Overfitting**: Excluding irrelevant features prevents overfitting.
- **Improved Interpretability**: Models with fewer features are easier to interpret and explain.
- **Faster Training and Inference**: A reduced feature set speeds up both training and inference phases.

5.5.1 Principal Component Analysis (PCA)

Purpose: Reduces the number of features by transforming the original features into a smaller set of uncorrelated variables, called principal components.

Application in Facial Recognition: PCA is widely used in facial recognition tasks. It captures the most important features, such as contours and shapes, that help distinguish between different faces.

Example: One popular example is *Eigenfaces*, a PCA-based approach for face recognition. Eigenfaces reduce the dimensionality of face images by focusing on the principal components that represent the most variance in facial features, allowing for more efficient facial recognition.

6 Assignment 8

6.1 Overfitting

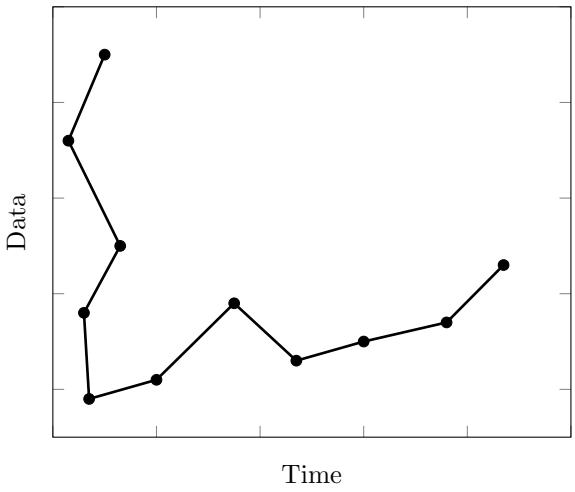


Figure 18: Overfitting

- **Definition:** Overfitting occurs when a model learns not only the underlying patterns in the training data but also the noise and fluctuations, leading to poor generalization on unseen data.
- **Example in Image Recognition:** An overfitted image recognition model might recognize specific lighting or background features in training images as essential to the class, failing to generalize to new images with different lighting or backgrounds.
- **Example in Decision Trees:** In decision trees, overfitting may occur when the tree is allowed to grow too deep, capturing minor variations in the data, which results in overly complex decision rules that don't generalize well.
- **Indicators:**
  - High accuracy on training data but low accuracy on test data.
  - Decision boundaries or fitted curves that follow data points too closely, as shown above.

6.2 Underfitting

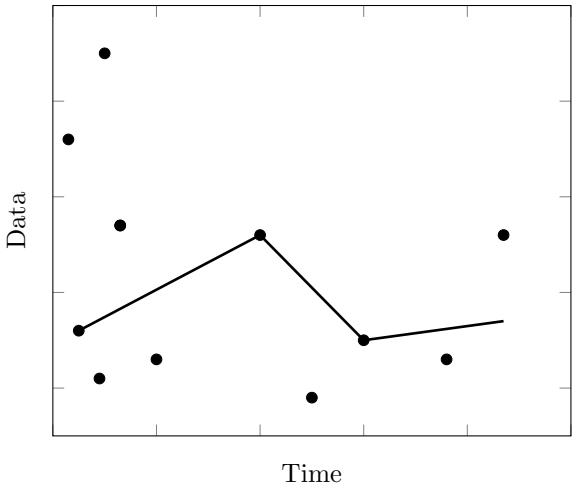


Figure 19: Underfitting

- **Definition:** Underfitting happens when a model is too simplistic, failing to capture the underlying patterns in the training data.
- **Example in Image Recognition:** An underfitted image recognition model might struggle to distinguish between different objects, as it hasn't learned sufficient details about each class.
- **Example in Decision Trees:** In decision trees, underfitting can occur when the tree depth is too shallow, producing overly general decision rules that cannot effectively separate classes.
- **Indicators:**
  - Low accuracy on both training and test data.
  - Simplistic decision boundaries or fitted lines that don't capture data variations, as shown above.

7 Assignment 9

7.1 Regression

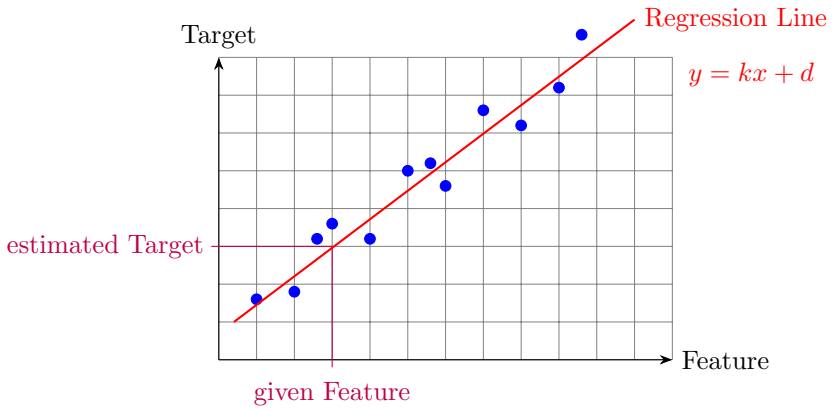


Figure 20: This figure illustrates a regression line applied to a dataset.

Linear Regression is a statistical method used to model the relationship between a dependent variable y and one or more independent variables x. In simple linear regression, where there is only one independent variable, the relationship can be represented by the equation:

$$y = \beta_0 + \beta_1 x + \epsilon$$

where:

- y is the dependent variable (target),
- x is the independent variable (feature),
- β_0 is the y-intercept of the regression line,
- β_1 is the slope of the line, which indicates the change in y for a one-unit change in x,
- ϵ represents the error term, capturing the residuals or deviations of actual values from predicted values.

The goal of linear regression is to estimate β_0 and β_1 to minimize the sum of the squared residuals, providing the best linear fit for the data.

7.2 R^2

The coefficient of determination, R^2, is a metric used to evaluate the performance of a regression model. It indicates the proportion of variance in the dependent variable y that is predictable from the independent variable x. The formula for R^2 is:

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

where:

- y_i represents the actual values of the dependent variable,
- $\hat{y}_i$ are the predicted values from the regression model,
- $\bar{y}$ is the mean of the actual values of y.

An R^2 value closer to 1 indicates a better fit, meaning that a larger portion of the variance in y is explained by x. Conversely, an R^2 close to 0 suggests that the model does not effectively capture the variance in y.

8 Assignment 10

8.1 Clustering - k-Means

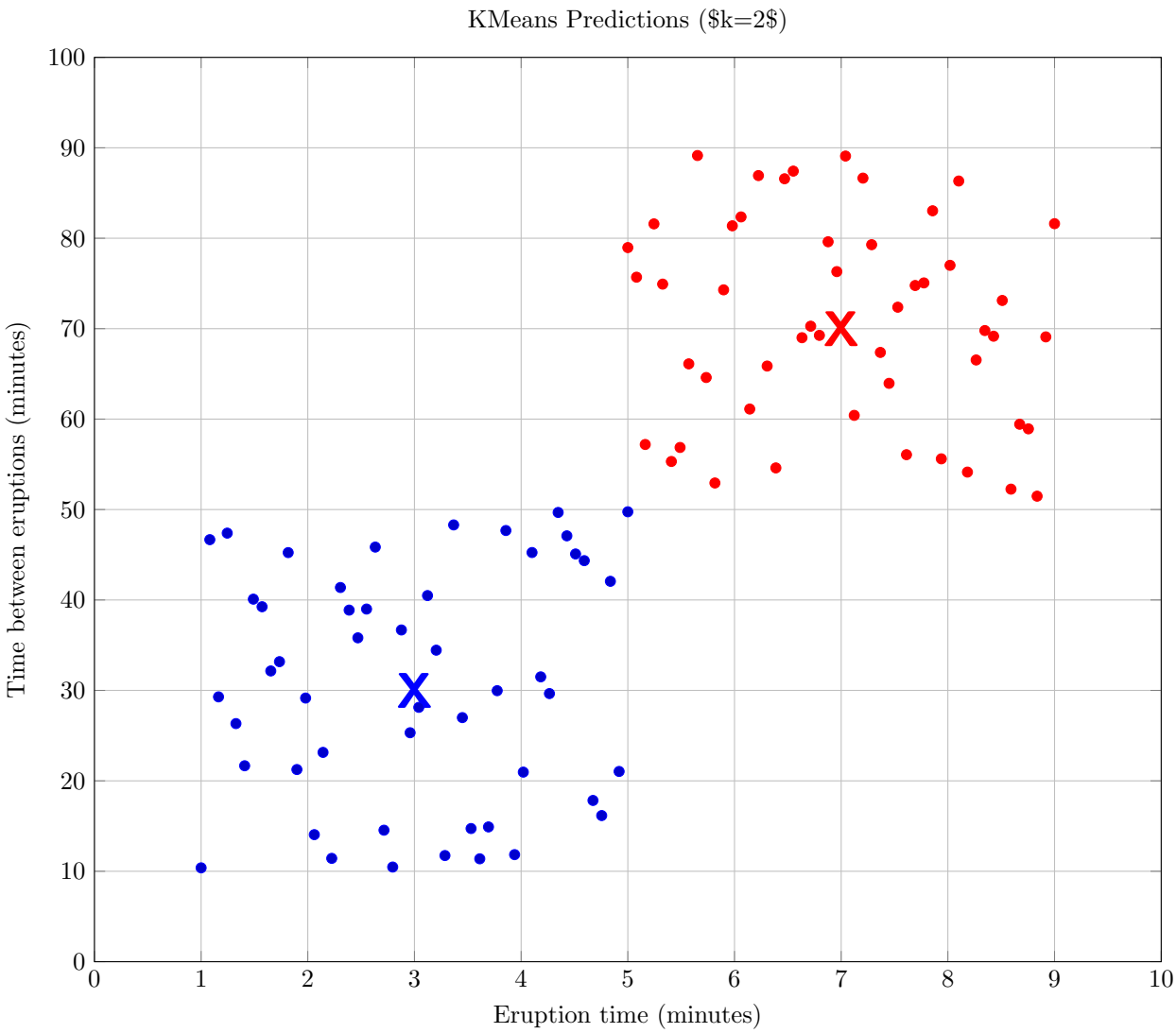


Figure 21: Illustration of the k-means clustering process with two clusters ($k=2$). The blue and red points represent the data points assigned to their respective centroids (marked with X).

8.2 Key Ideas of k-Means

1. **Random Initialization:** Centroids are initialized randomly to kickstart the clustering process.
2. **Assignment Step:** Each data point is assigned to the nearest centroid based on a distance metric (e.g., Euclidean distance).
3. **Update Step:** Centroids are updated by computing the mean of all assigned points in each cluster.
4. **Iterative Process:** The steps are repeated until the centroids stabilize (i.e., convergence) or the maximum number of iterations is reached.

8.3 Strengths and Weaknesses of k-Means

- **Strengths:**
  - Simple to understand and implement.
  - Computationally efficient for smaller datasets.
  - Works well with spherical, well-separated clusters.
- **Weaknesses:**
  - Sensitive to initialization of centroids, which can lead to suboptimal clustering.
  - Struggles with non-spherical clusters or clusters with varying sizes and densities.
  - Requires predefining the number of clusters (k).

8.4 Find the perfect k

Choosing the optimal number of clusters (k) in k-means is crucial for effective clustering. One widely used method is the **Elbow Method**, which involves the following steps:

1. **Calculate Within-Cluster Sum of Squares (WCSS):**
   - WCSS measures the compactness of clusters. A lower WCSS indicates tighter clusters.
2. **Run k-means for Multiple k Values:**
   - Evaluate WCSS for different values of k (e.g., 1 to 10).
3. **Plot the Elbow Curve:**
   - Plot k on the x-axis and WCSS on the y-axis. The curve typically decreases rapidly at first and then flattens out.
4. **Identify the "Elbow" Point:**
   - The optimal k is where the rate of WCSS reduction slows significantly (the "elbow"). Adding more clusters beyond this point provides diminishing returns in compactness.

Within-Cluster Sum of Squares (WCSS) measures the compactness of the clusters formed by k-means. It is defined as:

$$WCSS = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - c_k\|^2$$

Where:

- K is the total number of clusters.
- x_i is a data point belonging to cluster C_k.
- c_k is the centroid of cluster C_k.
- $\|x_i - c_k\|^2$ represents the squared distance between the data point and the centroid of its cluster.

Minimizing WCSS helps in finding compact clusters and is used in the elbow method to determine the optimal number of clusters.

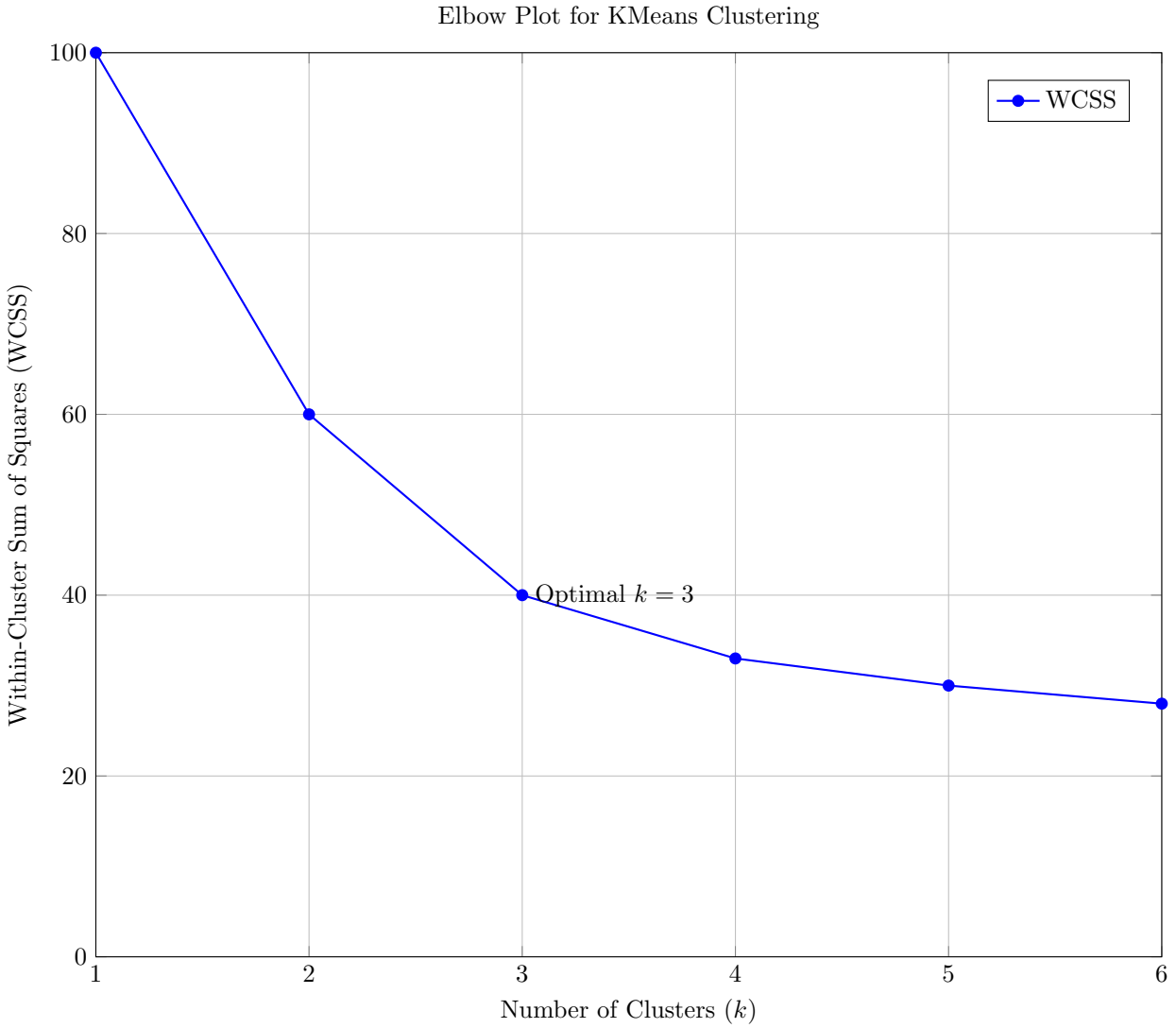


Figure 22: Elbow plot showing the optimal number of clusters ($k = 3$) for k-means clustering. The Within-Cluster Sum of Squares (WCSS) decreases sharply at $k = 3$ and levels off afterward, indicating the optimal k.

Example: In the sample plot, the "elbow" occurs at $k = 3$, indicating that three clusters provide a good balance between compactness and simplicity. This ensures meaningful clustering without overfitting or underfitting the data.

9 Assignment 11

9.1

section*Hierarchical Clustering

Conceptual Explanation

- **Hierarchical Clustering Process:** Hierarchical clustering (HC) is an unsupervised machine learning algorithm used to group data into clusters. It creates a tree-like structure (dendrogram) to represent how clusters are merged based on their similarity.

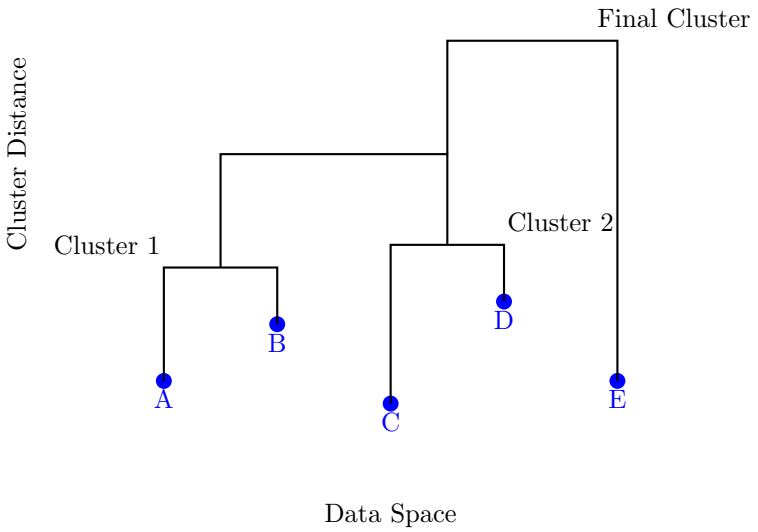


Figure 23: Hierarchical Clustering (HC) illustrated with a dendrogram. Points (A, B, C, D, E) represent data, and the merging process shows how clusters are formed based on distance.

- **Steps Illustrated:**
  1. **Start with Data Points:** Each data point (A, B, C, D, E) begins as its own cluster.
  2. **Merge Closest Clusters:** Clusters are merged iteratively based on a similarity metric (e.g., Euclidean distance). For instance, A and B form *Cluster 1*, and C and D form *Cluster 2*.
  3. **Final Merge:** Larger clusters are combined until all points belong to a single cluster.
- **Cutting the Dendrogram:** Cutting the dendrogram at a specific height (distance threshold) determines the number of clusters:
  - A low cut height results in many small clusters.
  - A higher cut height produces fewer, larger clusters.
- **Key Insights:**
  - The *height* of branches reflects the distance (dissimilarity) between clusters.
  - HC does not require predefining the number of clusters.

10 Assignment 12

What is PCA?

Principal Component Analysis (PCA) is a technique used to reduce the dimensionality of data while retaining as much variance as possible. It transforms the data into a new coordinate system where:

- **PC1** (Principal Component 1): Captures the maximum variance in the data.
- **PC2, PC3, ...**: Orthogonal components capturing decreasing amounts of variance.

The idea is to find new axes (principal components) that maximize the variance and minimize redundancy.

Illustration of PCA

The plot below demonstrates the PCA process on a 2D dataset:

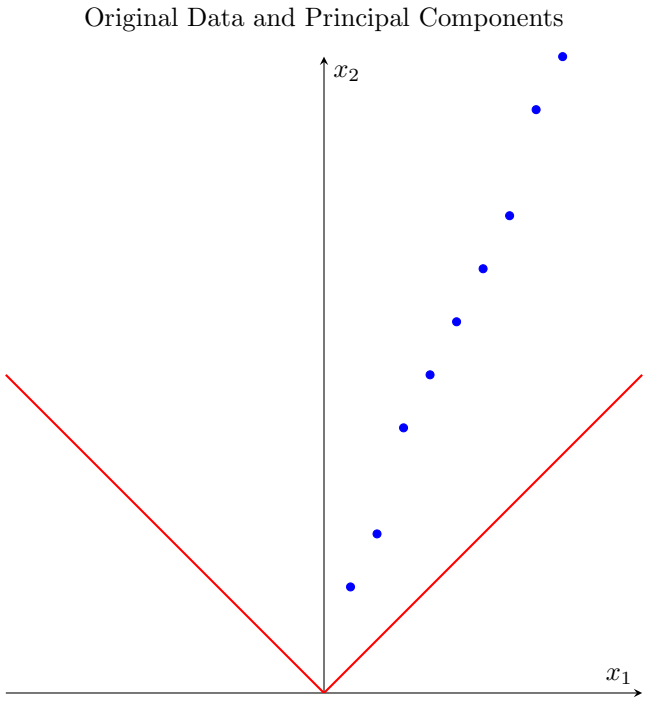


Figure 24: Visualization of the principal components in 2D space.

Interpretation of the Plot

- The blue points represent the original dataset in the x_1-x_2 space.
- The red lines show the new principal components (PC1 and PC2). These components form the new basis for the data, with PC1 capturing the maximum variance.
- The data can be projected onto PC1 to reduce it to a 1D representation, preserving most of the variance.

Projection of Data onto PC1

The second plot illustrates how the original data points are projected onto the first principal component (PC1):

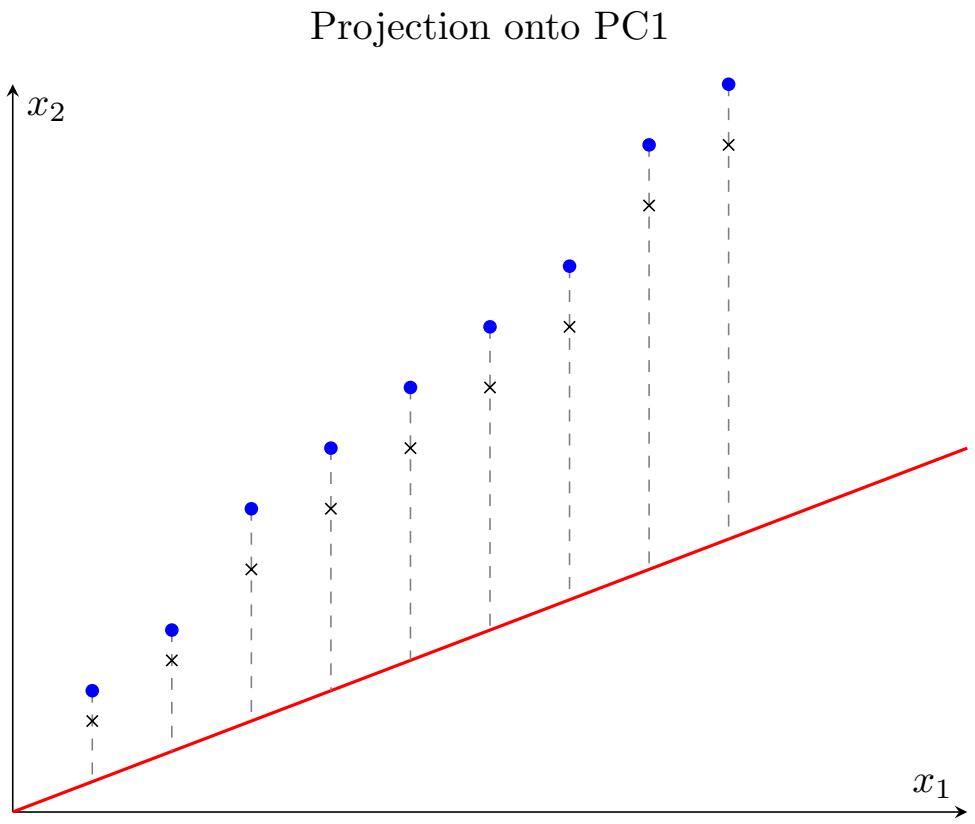


Figure 25: Projection of the original data points onto PC1.

Interpretation of the Projection

- The blue dots represent the original data points in 2D space.
- The red line is the first principal component (PC1), capturing the maximum variance in the data.
- The black crosses (×) are the projections of the original points onto PC1.
- The gray dashed lines show the distance of each data point to the projected point, representing the loss of information when reducing dimensions.
- This demonstrates how PCA transforms data by focusing on the most significant variation while simplifying the data representation.

By projecting onto PC1, we reduce the data from 2D to 1D while retaining most of the variance. This is the essence of PCA in dimensionality reduction.