2-)Probability & Random Variables

- $h(x_i) = f_i/n$ → relative frequency
- $h(x_i) = f_i/n*(c_i - c_{i-1})$ → density height
- f_i: frequency
- n: sample size
  - $P(A) \geq 0$
  - $P(S) = 1$
  - Mutually Exclusive Events
    - $P(A1 \cup A2 \cup \cdots) = P(A1) + P(A2) + \cdots$
- nPr=n! / (n−r)!
- nCr=n! / r! * (n−r)!
- MR (Multiplication Rule)
  - $P(A \cap B) = P(A \mid B) \times P(B)$
  - $P(A \cap B) = P(B \mid A) \times P(A)$
- *EMR*
  - $P(A \cap B \cap C) = P((A \cap B) \mid C) \times P(C \mid (A \cap B)) \times P(A \cap B)$
    - $P(A \cap B) = P(B \mid A) \times P(A)$
  - $P(A \cap B \cap C) = P((A \cap B) \mid C) \times P(C \mid (A \cap B)) \times P(B \mid A) \times P(A)$
- *VENN*
  - $\frac{P(A_1 \cap B) \cup P(A_2 \cap B) \cup \ldots P(A_k \cap B)}{P(B)}$
- *ME*
  - $\frac{P(A_1 \cap B) + P(A_2 \cap B) + \cdots P(A_k \cap B)}{P(B)}$
- *CP*
  - $P(A_1 \mid B) + P(A_2 \mid B) + \ldots . P(A_k \mid B)$
- *Independent Events*
  - $P(A \cap B) = P(A) \times P(B)$
- Conditional Probability
  - $P(T^+ \mid D) = \frac{N(T+ \cap D)/N(S)}{N(D)/N(S)}$
  - $P(T^+ \mid D) \neq P(D \mid T^+)$
  - $P(A \mid B) = \frac{\mathbf{P(A \cap B)}}{\mathbf{P(B)}}$ **!=** $P(c \mid x) = \frac{p(x \mid c)\, p(c)}{p(x) \rightarrow total\ demek}$
- Hyper Geometric Distribution
  - $P(X = x) = f(x) = \frac{\binom{m}{x}\binom{N-m}{n-x}}{\binom{N}{n}}$
  - select *n* items without replacement from a set of *N* items
  - *m* of the items are of one type
  - and *N* − *m* of the items are of a second type
- Expected Value & Variance & Standard Deviation
  - $E(X) = 3(0.3) + 4(0.4) + 5(0.3) = 4$
  - $E(Y) = 1(0.4) + 2(0.1) + 6(0.3) + 8(0.2) = 4$
  - $\sigma = E(\mu(x)) = E((X-\mu)^2)$ → Standard Deviation
  - $\sigma^2 = E(\mu(x)) = E((X-\mu)^2)$ → Variance = $\sqrt{\sigma^2}$

3-) Bayes & Naïve Bayes

- Bayesian Rule

$$P(c \mid x) = \frac{p(x \mid c)\, p(c)}{p(x)} = Posterior = \frac{likelihood \times prior}{evidence} = \frac{joint}{normalizer}$$

  - Updates the probability of an event
    - Based new evidence
  - Run Out Of Space & Time (modelling problem)
  - Tons Of Data (modelling problem)
  - Learning Joint Probability is infeasible (modelling problem)
- Posterior Probability ($P(\omega_j/x)$)
  - hypothesis is true or not in the light of relevant observations
- Joint Probability
  - $\boldsymbol{x} = (x_1, x_2), P(\boldsymbol{x}) = P(x_1, x_2)$
  - $P(x_1, x_2) = P(x_2|x_1)P(x_1) = P(x_1|x_2)P(x_2)$
  - $P(C, x) = P(x|C)P(C) \rightarrow Joint\ Probability$
  - $P(\neg C, x) = P(x|\neg C)P(\neg C) \rightarrow Joint\ Probability$
  - Sensitivity: “ ”, Specificity: “¬” {Maximum A Posterior}
- Normalized Histogram (3-1 Page 21 , 3-2 Page 8)
  - $P(x) = P(C, x) + P(\neg C, x)$
  - Total or Evidence
- Posterior
  - $P(C \mid x) = \frac{P(C, x)}{Normalizer}$
  - $P(\neg C \mid x) = \frac{P(\neg C, x)}{Normalizer}$
  - $P(C) \rightarrow Prior$
  - $P(Pos \mid C) \rightarrow Sensitivity$
  - $P(Neg \mid \neg C) \rightarrow Specificity$

- Naïve Bayes Normal Distribution (independent→y)

$$\hat{P}(x_j|c_i) = \frac{1}{\sqrt{2\pi}\sigma_{ji}} exp\left(-\frac{(x_j - \mu_{ji})^2}{2\sigma_{ji}^2}\right)$$

μ_{ji}: mean (avearage) of feature values x_j of examples for which $c = c_i$

σ_{ji}: standard deviation of feature values x_j of examples for which $c = c_i$

- $\mu = \frac{1}{N}\sum_{n=1}^{N} x_n, \ \sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu)^2$

4-) KNN & Regression

$$L_p(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{d} |\mathbf{x}_i - \mathbf{y}_i|^p\right)^{1/p} \quad L_2(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{d} |\mathbf{x}_i - \mathbf{y}_i|^2\right)^{1/2} \quad L_1(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{d} |\mathbf{x}_i - \mathbf{y}_i|$$

- Lp Norm(Minkowski), Euclidian Distance, Manhattan Distance
- y = mx + b (y→dependent & x → independent)
- $y = mx*b_1 + b_0$
- Q = $\sum_{i=1}^{n}(yi - \hat{y}_i)^2$ → Square Error
- Regression Line
- $\hat{y} = b_0 + b_1 x$
- Sum of Squared Errors – Point Estimation of Mean
- $SSE = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n}(y - b_0 - b_1 x)^2$
- $n = 10 \quad \sum x = 564 \quad \sum x^2 = 32604$
  - $\sum y = 14365 \quad \sum xy = 818755$
- $b_1 = \frac{n\sum xy - \sum x \sum y}{n\sum x^2 - (\sum x)^2} = 10.8$
- $b_0 = \frac{\hat{y}}{n} - b_1 \frac{x}{n}$
- $b_0 = 1436.5 - 10.8(56.4) = 828$
- $cov(x, y) = \frac{\sum_{i=1}^{n}(x_i - \bar{X})(y_i - \bar{Y})}{n-1}$ (relation whenever one changes)
- $r = \frac{cov\,ariance(x,y)}{\sqrt{var\,x}\sqrt{var\,y}} = \frac{\sum_{i=1}^{n}(x_i - \bar{X})(y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{X})^2 \sum_{i=1}^{n}(y_i - \bar{Y})^2}}$ (correlation)
- $\hat{r} = \frac{cov\,ariance(x,y)}{\sqrt{var\,x}\sqrt{var\,y}} = \frac{\frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n-1}}{\sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}\sqrt{\frac{\sum_{i=1}^{n}(y_i - \bar{y})^2}{n-1}}}$ (Relative strength)
- $\hat{\beta} = \frac{Cov(x,y)}{Var(x)}$
- Intercept = Calculate: $\bar{\propto} = \bar{y} - \hat{\beta}\bar{x}$
- $\hat{r} = \hat{\beta}\frac{SD_x}{SD_y}$
- $e_i = Y_i - \hat{Y}_i$ → Residual = Observed – Predicted
- Observed = Real Dot (Ground Root)
- Predicted = Value of Y in Line Correspond to X
- $s_{y.x}{}^2 = \frac{1}{n-2}\sum \boldsymbol{e_i^2} = \frac{1}{n-2}\sum(y_i - \hat{y}_i)^2$

5-) Clustering & PCA

- Euclidian, Manhattan, Infinity

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{d}(x_i - y_i)^2} \qquad d(\mathbf{x}, \mathbf{y}) = |\mathbf{x} - \mathbf{y}| = \sum_{i=1}^{d}|x_i - y_i| \qquad d(\mathbf{x}, \mathbf{y}) = \max_{1 \leq i \leq d}|x_i - y_i|$$

- K-Means: Initial Centre – Distance Matrix – Object Clustering – New Centre – Distance Matrix – Object Clustering – New Centre – Recompute Until No Change

6-) Decision Tree & Linear Classification

- $Entropy(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$
- $H(Y) = -\sum_{i=1}^{k} P(Y = y_i) \log_2 P(Y = y_i)$
- $H(Y \mid X) = -\sum_{j=1}^{v} P(X = x_j) \sum_{i=1}^{k} P(Y = y_i \mid X = x_j) \log_2 P(Y = y_i \mid X = x_j)$
  - $x = true$ *iken y′nin true ve false olma olasılıklarını topla*
  - $x = false$ *iken y′nin true ve false olma olasılıklarını topla*
  - *iki toplamında eksisini al dıştan*
- $IG(X_i) = H(Y) - H(Y \mid X_i)$
- $\text{argmax}\, IG(X_i) = \text{argmax}\, H(Y) - H(Y \mid X_i)$
- $H(Y \mid X{:}t) = P(X < t)H(Y \mid X < t) + P(X \geq t)H(Y \mid X \geq t)$
- $IG(Y \mid X{:}t) = H(Y) - H(Y \mid X{:}t)$
- $IG^*(Y \mid X) = max_t IG(Y \mid X{:}t)$
- $E = \frac{1}{2}(y - f(\sum w_i x_i))^2$
- $\mathbf{w}_j^r(\text{new}) = \mathbf{w}_j^r(\text{old}) - \mu\sum_{i=1}^{N}\frac{\partial\varepsilon(i)}{\partial\mathbf{w}_j^r}$ where $\frac{\partial\varepsilon(i)}{\partial\mathbf{w}_j^r} = \delta_j^r(i)\mathbf{y}^{r-1}(i) = Batch\ Learning$
- $\mathbf{w}_j^r(\text{new}) = \mathbf{w}_j^r(\text{old}) - \mu\frac{\partial\varepsilon(i)}{\partial\mathbf{w}_j^r}$ where $\frac{\partial\varepsilon(i)}{\partial\mathbf{w}_j^r} = \delta_j^r(i)\mathbf{y}^{r-1}(i) = Online\ Learning$

3-) Bayes & Naïve Bayes

- Zero Conditional Probability
- $\hat{P}(a_{jk}|c_i) = \frac{n_c + mp}{n + m}$
- n_c: number of training examples for which $x_j = a_{jk}$ and $c = c_i$
- n: number of training examples for which $c = c_i$
- p: prior estimate (usually, $p = 1/t$ for t possible values of x_j)
- m: weight to prior (number of "virtual" examples, $m \geq 1$)

7-) SVM

- $f(x) = w^T x_i + w_0$, $(w_0 = bias)$, bias corresponds to the output of an CNN when it has zero input
- $Init\ w = 0$
- $Cycle\ Through\ \{x, y\}$
- $If\ x\ is\ misclassified\ w \leftarrow w + \alpha sign(f(x))x$
- $Continue\ Until\ Data\ is\ Correctly\ Classified$
- Linear SVM
- $f(x) = \sum_i \alpha_i x_i \left(x_i^T x\right) + b$
- $r = \frac{wx_i + b}{\|w\|}$ where $\|w\| = \sqrt{w_1^2 + \cdots + w_n^2}$
- $margin = \frac{2}{\|w\|}$ → minimize weight vector, it will maximize margin
- $wx_i + b \geq 1\ if\ y_i = +1$ $\quad$ $wx_i + b \leq 1\ if\ y_i = -1$
- $y_i(wx_i + b) \geq 1)\ for\ all\ i$
- $\varphi = \frac{1}{2}(w^t w), find\ unique\ minimum\ by\ using\ training\ data$
- $\frac{1}{2} w * w + C \sum_{k=1}^{R} \varepsilon_k$ C is the Slack Variables
- $\max\left(0, 1 - yif(xi)\right)\ uses\ hinge\ loss\ approximates\ 0 - 1\ loss$
- Small C Allows constraints easily ignored → Large Margin – Wider (Soft) (c=1 e.g.)
- Large C Allows constraints hard to ignored → Narrow Margin (100) – Close to Hard (c=100)
- Infinite C enforces all constraints → Hard Margin ($c = \infty$)

Non-Linear SVM

- Linear SVM'de W Vektörünü ve S Değişkenlerini minimize edecek çözüm aradım.
- Non-Linear SVM'de higher dimension yapıyorum. (Polar Coordinates + Higher dimension)
- $\varphi\binom{x_1}{x_2} \rightarrow \binom{r}{\theta} R^2 \rightarrow R^2$
- $\varphi\binom{x_1}{x_2} \rightarrow \begin{pmatrix} {x_1}^2 \\ {x_2}^2 \\ \sqrt{2} x_1 x_2 \end{pmatrix} R^2 \rightarrow R^3$
- $f(x) = w^T x_i + w_0 \rightarrow f(x) = w^T \boldsymbol{\varphi(x)} + b$
- $Classifier: f(x) = w^T \boldsymbol{\varphi(x)} + b$
- RBF SVM
  - $f(x) = \sum_i \alpha_i y_i exp\left(-\|x - x_i\|^2 / 2\sigma^2\right) + b$
  - $\sigma \rightarrow Seperate\ Data$
    - $Hyper\ Parameter\ for\ Slack\ Variable$
    - $Decide\ our\ data\ by\ C\ \&\ \sigma$
  - Decrease σ, it moves towards NN Classifier
- Kernel Trick : Classifier can be learnt and applied without explicitly compute φ
  - RBF

9-) Deep Learning

- $f(x, W) = Wx$
- $L_i = \sum_{j \neq y_i} max\left(0, s_j - s_{y_i} + 1\right)$
  - $s_j \rightarrow Others$
  - $s_{y_i} \rightarrow Target$
- $L_i = \frac{1}{N} \sum_{i=1}^{N} L_i$
- MSE, Quadratic, L2 = $\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\boldsymbol{n}}$
- Mean Absolute Error, L1 = $\frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{\boldsymbol{n}}$
- Mean Bias Error = $\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{\boldsymbol{n}}$
- Cross Entropy Loss = $-(y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i))$
- Activations Functions: Sigmoid, tanh, ReLU, LReLU, Maxout, ELU
  - $\sigma(x) = \frac{1}{1 - e^{-x}}$
  - $\tanh(x)$
  - $\max(0, x)$
  - $\max(0.1x, x)$
  - $\max(w_1^T x + b_1, w_2^T x + b_2)$
  - $\begin{cases} \boldsymbol{x} & \boldsymbol{x \geq 0} \\ \boldsymbol{\alpha(e^x - 1)} & \boldsymbol{x < 0} \end{cases}$ → Exponential Linear Unit
- $f[n, m] * h[n, m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k, l] h[n - k, m - l]$
- Details of CNN
  - Output Size:
    - O: (N-F)/Stride + 1
    - N – Input Size
    - F – Frame Size
    - The O value can't be float.
  - Generally, Output Sizes
    - Stride: 1
    - Filters: FxF
    - Zero Padding with: (F-1)/2
      - Will preserve size stability
      - F= 3 → P=1
      - F= 5 → P=2
      - F= 7 → P=3
  - Output Volume Size:
    - OVS: (N+2*P-F)/Stride+1
    - 2 is double sided padding coefficient
  - Output Volume:
    - OV: OVS x OVS x Feature_Size
  - Number of Parameters:
    - Each Filter Has Parameters: F * F * Input_Depth + Bias
    - Total → NOP: Parameters * Feature_Size(new)
    - NOP: Parameters * Feature_Size -poolingde degisim olmaz
    - Bias: 0 or +1
    - Pooling Layers doesn't consist parameters
  - Summary
    - $\boldsymbol{W_1 * H_1 * D_1}$
    - K: Number of Filter
    - F: Spatial Extent
    - S: Stride
    - P: Amount of Zero Padding
    - $\boldsymbol{W_2 * H_2 * D_2}$
    - $\boldsymbol{W_2 = (W_1 - F + 2P)/S + 1}$ $\boldsymbol{W_2 = (W_1 - F)/S + 1}$
    - $\boldsymbol{H_2 = (H_1 - F + 2P)/S + 1}$ $\boldsymbol{H_2 = (H_1 - F)/S + 1}$
    - $\boldsymbol{D_2 = K}$ $\boldsymbol{D_2 = D_1}$