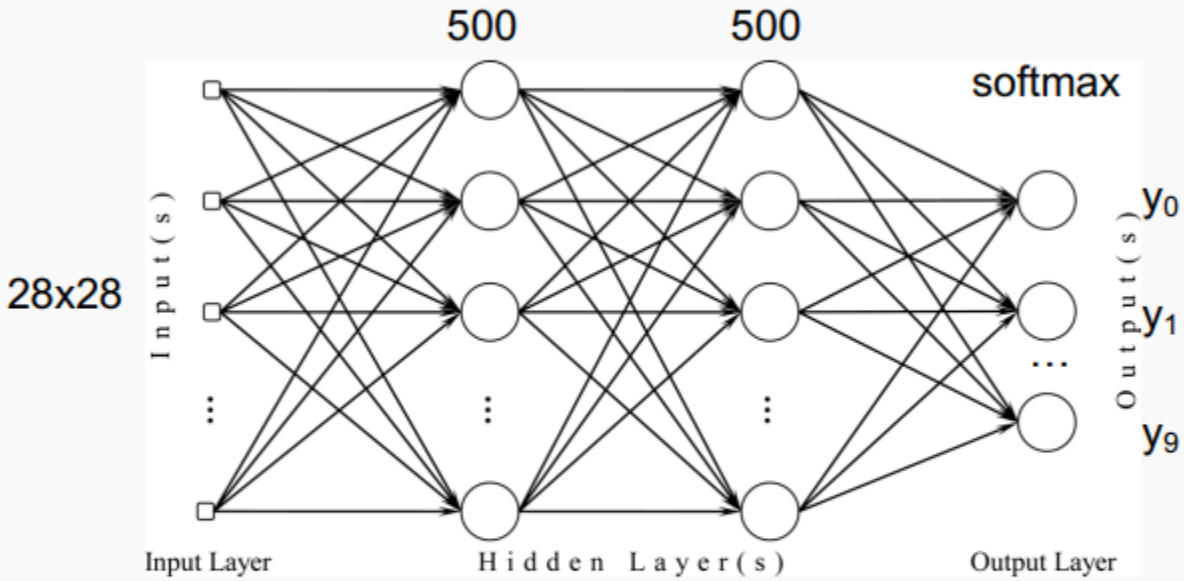
БАЗА – Ѓорги Костадинов – ВНП

Example of Deep Learning Implementation



Fully connected NN

```
model = sequential()   # layers are sequentially added
model.add( Dense(input_dim=28*28, output_dim=500))
model.add(Activation('sigmoid') )  #: softplus, softsign,relu,tanh,
model.add(Dense( output_dim = 500))
model.add (Activation('sigmoid'))
Model.add(Dense(output_dim=10))
Model.add(Activation('softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['acc
model.fit(x_train, y_train, batch_size=100, nb_epoch=20)
```

Овој код покажува како се гради и тренира **длабока невронска мрежа** користејќи библиотеката Keras.

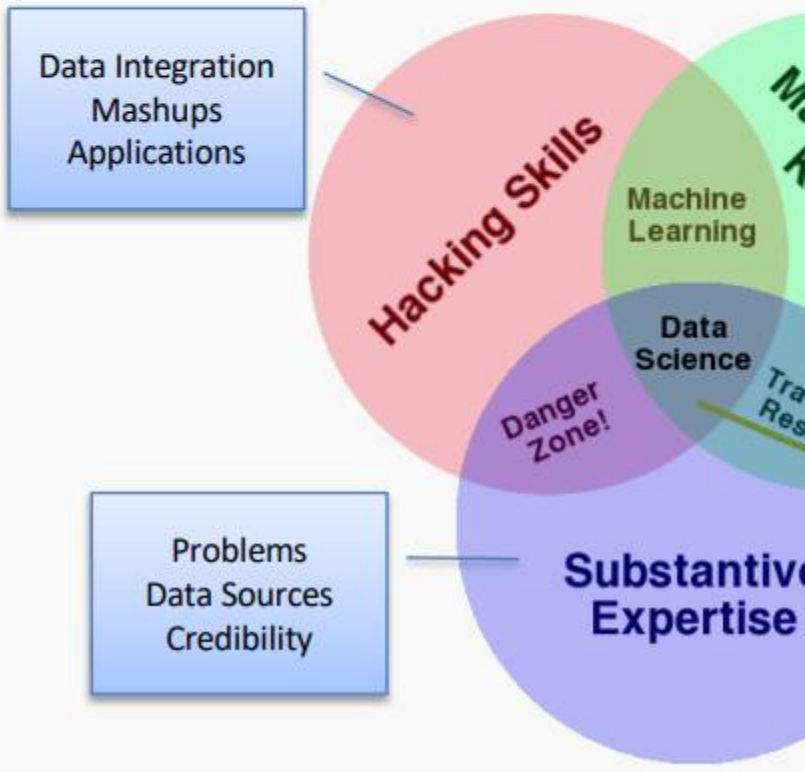
Како работи мрежата?

1. **Влезен слој:** Сликата (28x28 пиксели) се претвора во еден вектор од 784 вредности.
2. **Скриени слоеви:** Секој неврон добива информации од сите влезни пиксели (целосно поврзана мрежа).
3. **Излезен слој:** Дава 10 вредности (по една за секоја класа). softmax ги претвора овие вредности во веројатности.

Зошто се користат овие параметри?

- **Sigmoid:** Добро за скриени слоеви, но може да предизвика „vanishing gradient“ проблем за многу длабоки мрежи.
- **Softmax:** Идеален за класификација со повеќе класи.
- **Adam оптимизатор:** Автоматски ја прилагодува брзината на учење за секоја тежина.

Three Essential Skills of Data Scientist



Кратко објаснување:

Сликата ги прикажува **три клучни вештини на податочните научници**:

1. **Hacking Skills (Програмерски вештини):**
   - Работа со податоци преку алатки и кодирање.
   - Примери: Интеграција на податоци, создавање апликации.
2. **Math & Statistics Knowledge (Математика и статистика):**
   - Разбирање на модели, предвидувања и анализа на несигурност.
   - Пример: Примена на машинско учење и статистички методи.
3. **Substantive Expertise (Доменско знаење):**
   - Познавање на специфични проблеми и кредибилитет на изворите на податоци.
   - Пример: Разбирање на контекстот на податоците (здравство, маркетинг, итн.).

Преклопувања:

- **Машинско учење:** Каде се среќаваат програмерските и статистичките вештини.
- **Традиционално истражување:** Комбинација на статистика и доменско знаење.
- **Опасна зона:** Само програмерски вештини без разбирање на статистика или домен.

Цел: Балансирана комбинација на овие вештини за ефективни апликации на податоците.

- **Превод и објаснување:**
- **Зошто „Опасна зона“?**
- **Рони Кохави** (Ronny Kohavi), во својот главен говор на KDD 2015, објаснува дека луѓето се исклучително снаодливи во објаснување на „многу изненадувачки резултати“. За жал, повеќето од тие резултати се предизвикани од грешки во обработката на податоците.
- **Внимание кон HiPPOs:** Тоа се „Мислења на највисоко платените личности“ (Highest Paid-

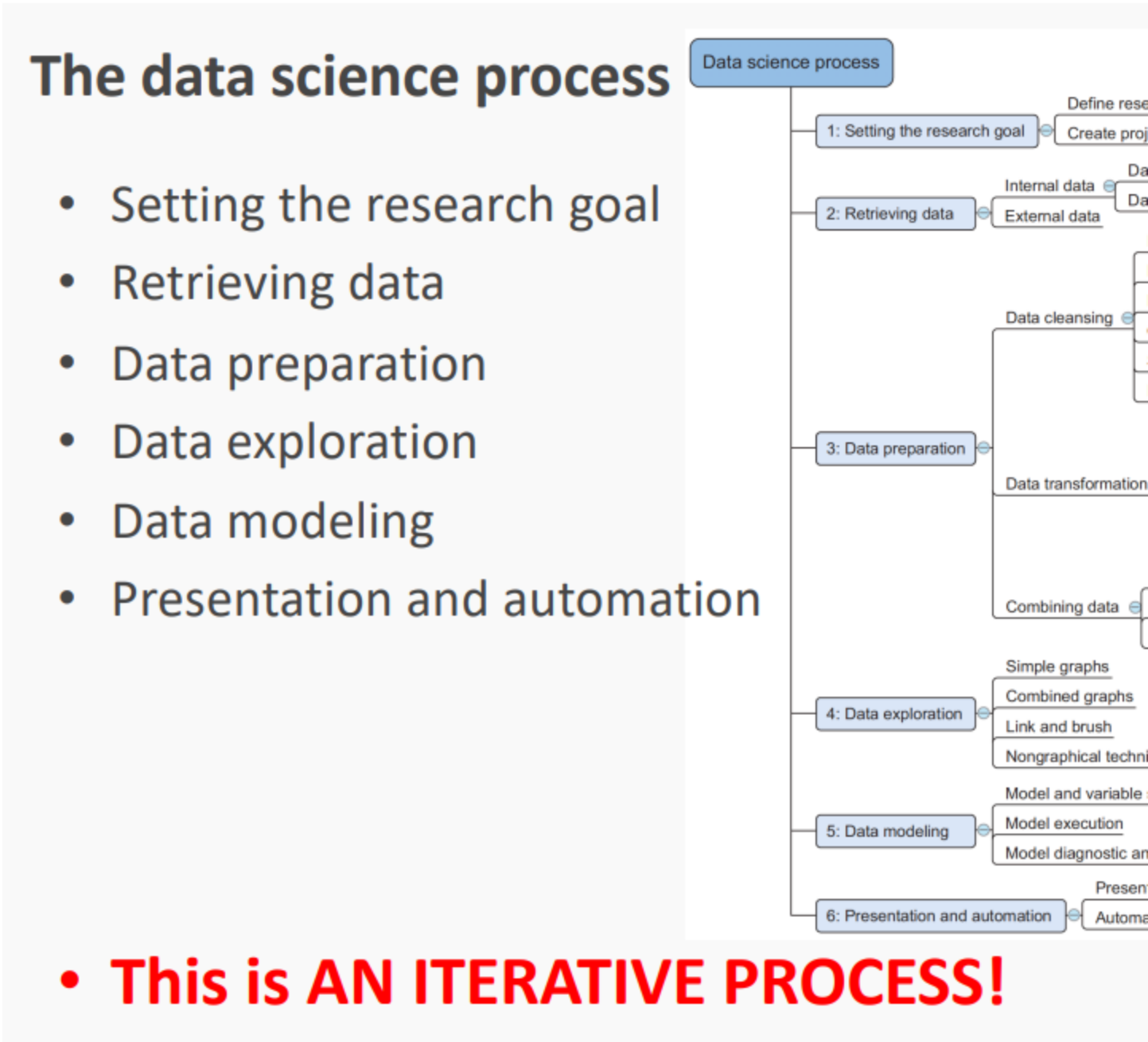
Person’s Opinion). Кохави предупредува дека треба да се слушаат клиентите и да не се дозволи HiPPO да уништи добри идеи.

- **Пример:**
- **Грешки во податочниот процес:** Во еден систем за предвидување продажба, грешка во форматот на податоците (недостасувачки вредности) може да доведе до „изненадувачки“ резултат – на пример, дека еден производ има нулта продажба.
- **Решение:** Ревизија на обработката на податоци и верификација на излезот.
- **HiPPO сценарио:** Во маркетинг кампања, највисоко платениот менаџер инсистира на одреден пристап (на пример, користење ТВ реклами), иако анализата покажува дека онлајн маркетинг би бил поефикасен.
- **Решение:** Докажете со податоци и експериментирајте за да го поддржите предложениот пристап.

- **Поука: „Опасната зона“ се јавува кога не се поставуваат прашања за податоците или**

резултатите и кога одлуките се базираат на субјективни мислења наместо на докази.

--- **Клучна поента:** Граф-податоците се идеални за моделирање сложени односи, како оние во социјалните мрежи или комуникациски системи.



Чекори во процесот:

1. Поставување на истражувачки цели (Setting the research goal)

- **Што е целта?**

- Разјаснување на проблемот или прашањето што треба да се реши.
- **Пример:** Како да се зголемат продажбите на одреден производ?

- **Акции:**

- Создавање „проектен план“ со јасни цели и мерки на успех.

---

2. Преземање податоци (Retrieving data)

- **Што е потребно?**

- Собирање на соодветни податоци од **внатрешни извори** (бази, логови) или **надворешни извори** (APIs, отворени податоци).

- **Предизвици:**

- Податоци со грешки при внесување.
- Недостасувачки вредности или несоодветни формати.

---

3. Подготовка на податоците (Data preparation)

- **Чистење на податоците (Data cleansing):**
  - Отстранување на грешки, невозможно големи вредности, аутлаери, празни полиња.
- **Трансформација на податоци (Data transformation):**
  - Агрегација на податоци, создавање нови метрики, намалување на бројот на променливи.
  - **Пример:** Заменување на празни полиња со медијана или просек.

---

4. Истражување на податоците (Data exploration)

- **Цел:** Да се визуализираат податоците за разбирање на трендовите и шемите.
- **Акции:**
  - Креирање графици (хистограми, корелациони графици).
  - Неформални анализи за откривање потенцијални проблеми или обрасци.

---

5. Моделирање на податоците (Data modeling)

- **Цел:** Креирање модели за предвидување или класификација.
- **Акции:**
  - Избор на модел (регресија, класификација, кластеринг).
  - Тренирање и тестирање на моделот.
  - Дијагностика на моделот (мерење точност, прецизност).

---

6. Презентација и автоматизација (Presentation and automation)

- **Презентација:**
  - Јасно комуницирање на резултатите преку графици, извештаи или алатки за одлучување.
  - **Пример:** Прикажување на трендови за продажба пред менаџментот.
- **Автоматизација:**
  - Креирање на алатки или скрипти кои автоматски ќе го извршуваат анализираниот процес.

Внатрешни податоци (Internal Data)

- Проценете ја релевантноста и квалитетот на податоците кои се лесно достапни во вашата компанија.
- Повеќето компании имаат програми за одржување на клучни податоци, па дел од работата за чистење можеби веќе е завршена.
- Овие податоци може да се чуваат во официјални складишта за податоци, како бази на податоци, data marts, data warehouses или data lakes.
- Сепак, постои можност податоците сè уште да се чуваат во Excel или текстуални фајлови на компјутерот на доменски експерт.
- Пристапот до податоците е уште еден тежок предизвик. Организациите ја разбираат вредноста и чувствителноста на податоците и често имаат политики кои осигуруваат дека секој има пристап само до она што му е потребно и ништо повеќе.

Надворешни податоци (External Data)

- Ако податоците не се достапни во рамките на вашата организација, побарајте ги надвор.
- Многу компании се специјализирани за собирање на вредни информации. На пример, **Nielsen** и **GFK** се добро познати за ова во трговската индустрија.
- Други компании обезбедуваат податоци за да ги збогатите нивните услуги и екосистем. Така функционираат Twitter, LinkedIn и Facebook.
- **Open Data:** Отворените податоци се уште еден извор на надворешни информации.

Аутлаерите можат да бидат важни:

- Можат да покажат грешки во податоците.
- Можат да укажат на области со недоволно податоци.

- Можат да бидат валидни, но невообичаени точки.

 Прв чекор: Користи графикон или хистограм за лесно идентификување.

 Ефект врз моделот: Аутлаерите може значително да го изобличат моделот ако не се третирани правилно.

Кратко објаснување:

Истражувачката анализа на податоци е фаза каде што:

1. **Главна цел:** Се користат графички и визуелни алатки за подобро разбирање на податоците.
   - **Пример:** Хистограми, корелациони матрици, scatter плотови.
2. **Дополнителни техники:** Вклучува табели, кластеринг и едноставни модели за анализа.
3. **Примерна примена:** Може да се идентификуваат трендови, аномалии и интеракции меѓу променливите.
4. **Зошто е важна:** Оваа анализа помага во дефинирање на податочните структури и

откривање на првични увидувања кои ќе ја водат понатамошната работа.

-**Превод и објаснување:**

Стриминг податоци се податоци кои пристигнуваат во системот веднаш кога се случува некој настан, наместо да бидат внесени наеднаш како група (batch). Заради оваа динамичност, потребно е процесот да се прилагоди за обработка на ваков тип информации.

Пример:

- „Што е трендинг“ на Твитер (обновување на податоците во реално време).
- Пренос во живо од спортски или музички настани.
- Пазар на акции (обновување на цените секоја секунда).

Ова значи дека системот мора да биде брз и ефикасен за да обработи и прикаже информации во реално време.

Превод и објаснување:

Ламбда архитектурата е начин за обработка на огромни количини податоци со комбинирање на batch (групна) и stream (во реално време) методи. Целта е да се постигне баланс меѓу брзина, капацитет и сигурност.

- Batch обработката дава точни и детални анализи на големи сетови податоци.
- Stream обработката овозможува анализа во реално време на тековните податоци.

Пример:

Е-комерц платформа:

- Batch обработка за да генерира дневни извештаи за продажба.
- Stream обработка за прикажување на производи што се моментално најпопуларни.

Превод и објаснување:

Data Lake е огромен складиште за податоци, базирано на евтини технологии, кое овозможува полесно

зачувување, обработка, архивирање и анализа на сурови податоци.

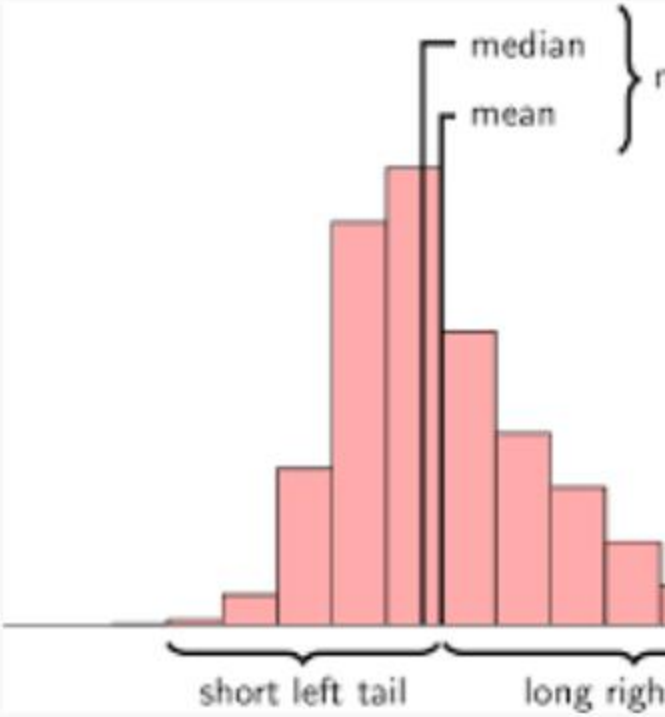
Превод и објаснување:

1. **Табеларни податоци**: Податоци претставени како табела со две димензии (редови и колони). Секој ред е еден запис, а секоја колона претставува еден тип на податок (пример: CSV, Excel).
   Пример: Табела со имиња, години и адреси.
2. **Структурирани податоци**: Секој запис е во форма на речник, кој може да биде комплексен и со повеќе нивоа (пример: JSON, XML).
   Пример: JSON запис за производ: { "име": "Чевли", "цена": 50 }.
3. **Полуструктурирани податоци**: Не сите записи имаат исти клучеви или некои записи не користат структура на клуч-значење.
   Пример: Еден запис има поле за "адреса", друг не.

Овие формати зависат од природата на податоците и нивната примена.

Mean, median, and skewness

- The mean is sensitive to outliers:



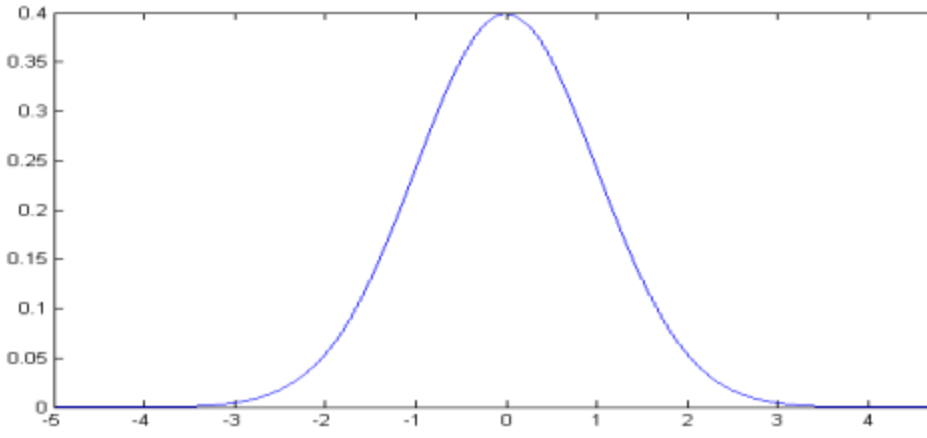
- The above distribution is called **right** greater than the median. Note: **skew** tail".

На кратко:

Кога дистрибуцијата е десно-искосена, средината се движи кон десната страна заради екстремно големи вредности, додека медијаната останува поблиску до центарот на податоците.

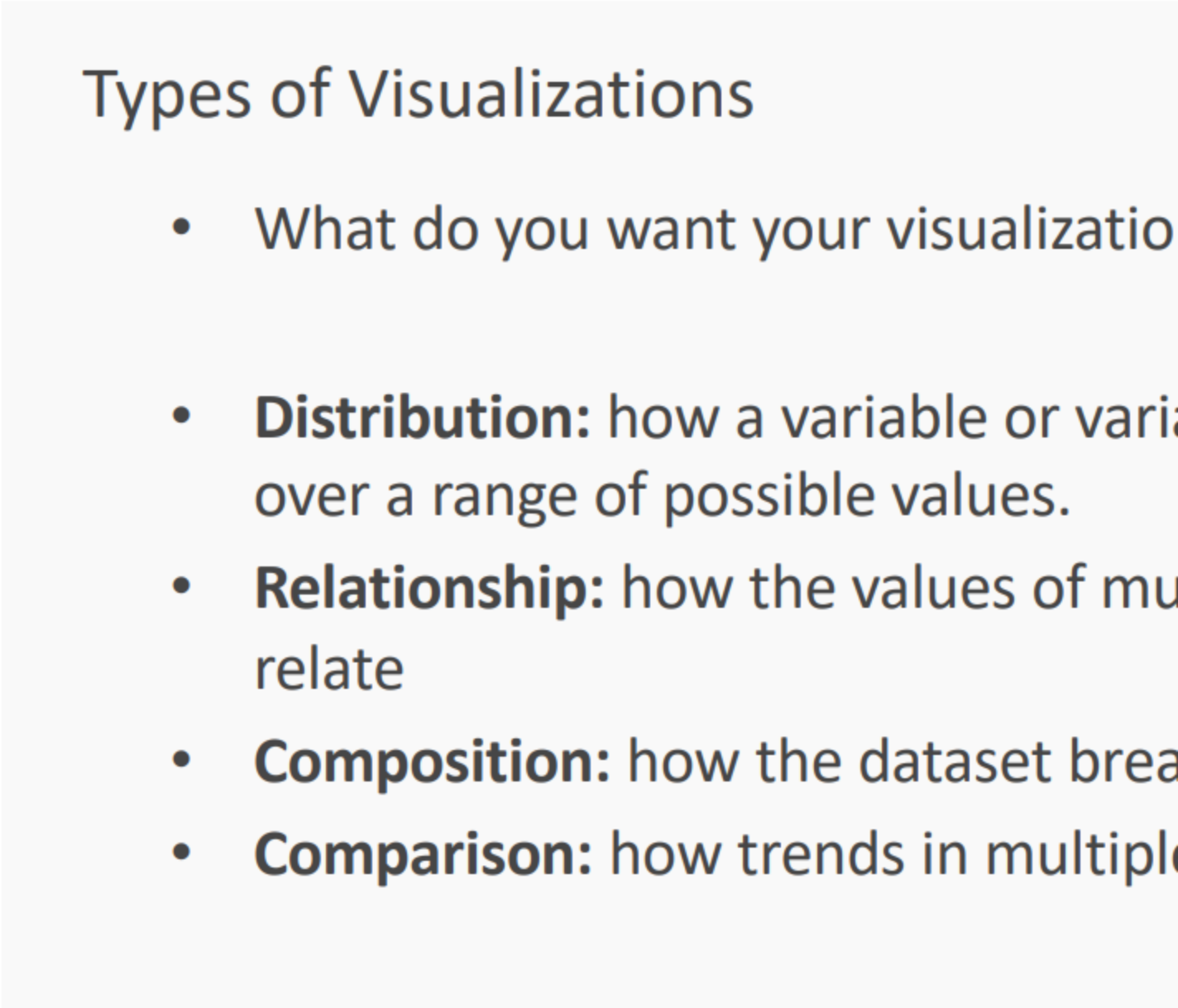
Central Limit Theorem

- The distribution of the sum (or mean) of independent identically distributed random variables Xi approaches a normal distribution as $n \to \infty$.
- The common parametric statistical tests assume normally-distributed data, but the mean and variance measures of the test statistic
- They typically work reasonably well even if the data are not normally distributed as long as the s



Централната гранична теорема објаснува зошто многу статистички тестови работат дури и кога податоците не се нормални, бидејќи со доволно голем примерок

збирот или средината станува нормално распределена.



Сликата ги прикажува **типовите на визуелизации** и целта што можат да ја постигнат во анализа на податоци:

1. **Distribution (Распределба):**

- Покажува како променливите во сетот на податоци се распределени во рамките на можни вредности.
- Пример: Хистограми, boxplots.

2. **Relationship (Однос):**
   - Покажува како вредностите на повеќе променливи се поврзани меѓусебно.
   - Пример: Scatterplots, линиски графикони.

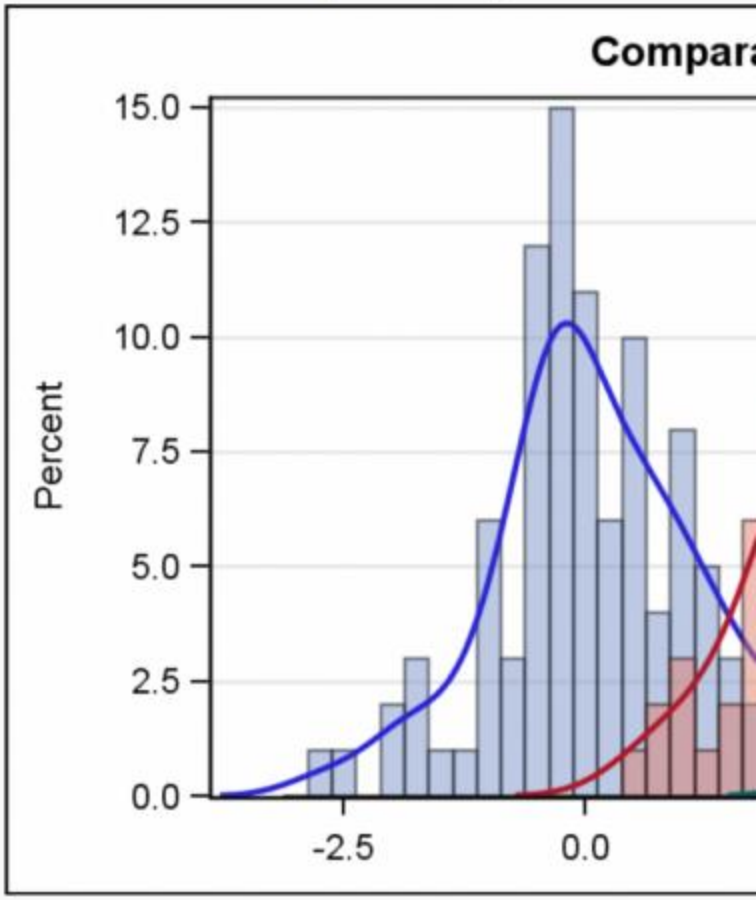
3. **Composition (Состав):**
   - Покажува како податоците се делат на подгрупи или делови.
   - Пример: Pie charts, stacked bar charts.

4. **Comparison (Споредба):**
   - Покажува како трендовите или вредностите на повеќе променливи или податочни сетови се споредуваат.
   - Пример: Bar charts, line charts.

На кратко:
Изборот на визуелизација зависи од тоа што сакате да покажете – распоредот на податоци, нивните релации, поделби или споредби меѓу категории.

Multiple histograms

- Plotting **multiple histograms (**and **ke**
  distribution, here) on the same axes
  different variables compare (or how
  groups).



Сликата покажува **множество хистограми** и **kernel density estimates** на истите оски за да се споредат

различни променливи или да се анализира како една променлива се разликува меѓу групи.

1. **Хистограми (Histogram):**
   - Ги покажуваат фреквенциите на податоците за различни групи (x, y, z) преку столбови.
   - Секој столб претставува опсег од вредности и бројот на податоци што припаѓаат на тој опсег.
2. **Kernel Density Estimate (KDE):**
   - Ги покажува распределбите на податоците како мазни криви.
   - Ова е корисно за визуелизирање на основната форма на распределбата без да се ограничите на дискретни столбови.
3. **Цел:**
   - Се користи за **споредба** на променливи (на пр., x, y, z) или за да се идентификуваат разлики во нивните распределби.

На кратко:
Оваа визуелизација комбинира хистограми и мазни

криви (KDE) за да ги спореди распределбите на повеќе групи податоци (x, y, z) на истите оски.

Introduction to ML 1

1. Полиноминална регресија, преклопување (overfitting), и регуларизација во науката за податоци

Полиноминална регресија

- **Што е? Во науката за податоци, ова е техника за моделирање на сложени односи помеѓу независните променливи (X) и зависните променливи (Y), кога врската не е права линија.**
- **Пример во Data Science: Ако предвидуваш цената на куќа базирано на нејзината квадратура, може да забележиш дека цената расте побрзо за многу големи куќи (не линеарно). Полиноминалната регресија ќе ја моделира таа крива врска.**

Преклопување (Overfitting)

- **Што е? Кога моделот „ги учи наизуст“ тренинг податоците, но не може да се справи со нови податоци.**
- **Пример во Data Science: Ако трениращ модел за предвидување на вредноста на акциите и ги вклучиш сите можни детали, моделот ќе се „залепи“ за шумовите во податоците и нема да предвидува добро за нови акции.**

Регуларизација

- **Што е? Метод за да се направи моделот попрецизен, дури и за нови податоци, со ограничување на сложеноста на функцијата.**
- **Пример во Data Science: Во моделирање на времето (температура), регуларизацијата може да спречи премногу сложени полиноми кои не се применливи за идни предвидувања.**

---

2. Одлуковни дрва (Decision Trees) во науката за податоци

Што се одлуковни дрва?

- **Што е? Во науката за податоци, одлуковните дрва се алгоритми кои ги делат податоците на основа на прашања или услови. Тие работат добро за класификација или регресија.**
- **Пример во Data Science: Класификација на клиенти дали ќе купат производ базирано на нивната возраст и приход. Прво прашање: „Дали клиентот има плата над 50,000?“**

Како работат?

- **Пример: Ако работиш на задача за одредување дали некој е погоден за заем:**
  1. **Прво прашање: „Дали лицето има кредитна историја?“**
  2. **Ако „да“ -> Следно прашање: „Дали има приход над 30,000?“**
  3. **Ако „не“ -> Одлука: „Непогоден за заем.“**

Преклопување кај одлуковните дрва

- **Што е? Ако дрвото е премногу длабоко, ќе ги „запамети“ деталите од податоците за обука и нема да биде корисно за нови податоци.**
- **Пример: Ако дрвото за предвидување на времето научи правила специфични за денови со 20°C, ќе биде бескорисно за други денови.**

Едноставни примери во науката за податоци

1. **Полиноминална регресија: Предвидување на продажбата на производ базирано на сезонските трендови (зимска и летна продажба).**
2. **Одлуковни дрва: Класификација на тип на корисници (премиум или обични) базирано на нивното купување.**

K-fold cross validation

- **K-fold cross validation** is one way to improve over the holdout method.
- The data set is divided into *k* subsets, and the holdout method is repeated *k* times.
- Each time, one of the *k* subsets is used as the test set and the other *k-1* subsets are put together to form a training set. Then the average error across all *k* trials is computed.
- The advantage of this method is that it matters less how the data gets divided.
  - Every data point gets to be in a test set exactly once, and gets to be in a training set *k-1* times.
  - The variance of the resulting estimate is reduced as *k* is increased.
- A variant of this method is to randomly divide the data into a test and training set *k* different times.
  - The advantage of doing this is that you can independently choose how large each test set is and how many trials you average over.

K-fold cross validation е техника во машинското учење која се користи за проценка на перформансите на моделот, со цел да се намали ризикот од преклопување (overfitting) или подприспособување (underfitting).

Најбитните работи од презентацијата за полагање на предметот „Вовед во науката за податоци“

---

1. Полиноминална регресија (Polynomial Regression)

- **Што е тоа?** Полиноминалната регресија е нелинеарен модел кој користи полиноми од повисок ред (M-ти степен) за да ја објасни врската помеѓу зависната променлива YYY и независната променлива XXX.
- **Формула:** y=β0+β1x+β2x2+…+βMxM+ϵy = \beta_0 + \beta_1x + \beta_2x^2 + \ldots + \beta_Mx^M + \epsilony=β0+β1x+β2x2+…+βMxM+ϵ
- **Клучни концепти:**
  - **Curve fitting:** Процесот на прилагодување на моделот на податоците. Премногу сложен модел може да резултира со „overfitting“.

- **Overfitting:** Се јавува кога моделот е премногу сложен и ги запомнува податоците од обуката, но не може добро да ги предвиди новите податоци.

---

2. Преклопување (Overfitting) и Регуларизација

- **Overfitting:**
  - Моделот е премногу сложен, со висока точност на обука, но лоши предвидувања за нови податоци.
  - **Како да се избегне?**
    - **Регуларизација (Regularization):** Казнување на сложеноста на моделот.
    - **Примери за регуларизација:**
      - **LASSO (L1 регуларизација):** Ги минимизира апсолутните вредности на параметрите.

$$L_{\text{LASSO}}(\beta) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{J}$$

LLASSO(β)=1n∑i=1n(yi−y^i)2+λ∑j=1J|βj|

|\beta_j|LLASSO(β)=n1i=1∑n(yi−y^i)2+λj=1∑J|βj|

- **Ridge (L2 регуларизација):** Ги минимизира квадратите на параметрите.

LRidge(β)=1n∑i=1n(yi−y^i)2+λ∑j=1Jβj2L_{\text{Ridge}}(\beta) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{J} \beta_j^2LRidge(β)=n1i=1∑n(yi−y^i)2+λj=1∑Jβj2

---

3. Одлуковни дрва (Decision Trees)

- **Што се тоа?**
  - Модел кој користи прашања („да“/„не“) за да ги подели податоците на подгрупи и да донесе одлука. На крајот, секоја гранка води до класа или предвидување.
  - Пример: Одлука дали да носиш чадор:
    1. „Дали врне?“ → Ако „да“, носи чадор; ако „не“, не носи чадор.

---

4. Геометрија на одлуковните дрва

- **Како работат?**
  - Секој јазол претставува услов за поделба (на пр., „Дали приходот е > 50,000?“).
  - Листовите на дрвото претставуваат класите (на пр., „Кредит одобрен“ или „Кредит одбиен“).

---

5. K-fold Cross Validation

- **Што е тоа?**
  - Техника за проценка на перформансите на моделот. Податочниот сет се дели на kkk делови (folds).
  - Во секоја итерација, еден дел се користи за тестирање, а останатите k−1k-1k−1 делови за обука.
  - **Пример:**
    - Ако имаш 100 податоци и користиш k=5k=5k=5, секој „fold“ ќе има 20 податоци. Моделот ќе се тестира на

различен „fold“ во секоја итерација, а точноста ќе се пресмета како просек.

---

6. Клучни метрики за поделба кај одлуковни дрва

- **Критериуми за поделба:**

1. **Gini Index:** Мери чистота на подгрупите.
   $$Gini = 1 - \sum_{k=1}^{K} p_k^2$$
   Gini=1−∑k=1Kpk2Gini = 1 - \sum_{k=1}^{K} p_k^2Gini=1−k=1∑Kpk2
2. **Entropy (Ентропија):** Мери несигурност во подгрупите.
   $$Entropy = -\sum_{k=1}^{K} p_k \log_2(p_k)$$
   Entropy=−∑k=1Kpklog⁡2(pk)Entropy = -\sum_{k=1}^{K} p_k \log_2(p_k)Entropy=−k=1∑Kpklog2(pk)

---

7. Преклопување и кастрење (Pruning) на дрвата

- **Проблем:** Длабоките дрва можат да го „запомнат“ тренинг сетот и да бидат лоши за нови податоци.
- **Решение:**

- **Кастрење (Pruning):** Се намалува длабочината на дрвото со отстранување на непотребни гранки.
- **Раното запирање (Early Stopping):** Прекинување на растењето на дрвото при одредени услови (на пр., минимален број на податоци во лист).

Примери за практична примена:

1. **Полиноминална регресија:** Прогнозирање на цена на недвижности базирано на квадратура.
2. **Одлуковно дрво:** Предвидување дали клиент ќе купи производ базирано на неговата возраст и приход.
3. **K-fold Cross Validation:** Проверка дали модел за класификација на е-пораки (спам/не-спам) работи добро со сите делови од податоците.

Ова се најбитните работи за твоето полагање. Дали сакаш да разјасним некој дел подетално?

Најбитните работи од презентацијата за полагање на предметот

---

1. Ensemble Models: Bagging, Random Forest, Boosting

1. Bagging (Bootstrap Aggregating)

- **Што е тоа?**
  - Метод кој користи повеќе подгрупи на податоци (со замена) за да изгради модели. Се комбинираат резултатите од сите модели преку **гласање** (за класификација) или **просек** (за регресија).
- **Како работи?**

1. Податоците се делат на повеќе примероци со замена (bootstrapping).
2. Се тренира модел на секој примерок.
3. Се комбинираат резултатите од сите модели.

- **Предности:** Намалување на варијансата и зголемување на точноста.
- **Пример:** Ако предвидуваш дали студент ќе положи испит, можеш да изградиш повеќе дрва од различни групи податоци.

---

2. **Random Forest**

- **Што е тоа?**
  - Модифицирана верзија на Bagging со декорелација на дрвата.
- **Како работи?**

1. Секое дрво користи различен примерок (bootstrap sample).
2. На секој сплит, само подгрупа од предиктори е достапна.

- **Предности:**
  - Намалување на корелацијата меѓу дрвата.
  - Подобра точност и помала варијанса.

- **Пример:** Предвидување на цените на куќи базирано на карактеристики како големина, број на соби итн.

---

3. **Boosting**

- **Што е тоа?**
  - Метод каде моделите се гради последователно, при што секој нов модел ги поправа грешките на претходниот.
- **Типови:**
  - **Adaptive Boosting (AdaBoost):** Преоценува тежините на податоците, фокусирајќи се на оние со поголеми грешки.
  - **Gradient Boosting (на пр., XGBoost):** Тренира нови модели на остатоците (residuals) од претходните.
- **Предности:** Намалување на пристрасноста (bias) и зголемување на точноста.
- **Пример:** Класификација на е-пораки (спам/не-спам).

---

2. Naïve Bayes Classifier

1. Што е тоа?

- Класификатор базиран на **Бејесовиот теорем**, кој ги проценува веројатностите за секоја класа базирано на дадени атрибути.
- **Формула:** P(C|A)=P(A|C)P(C)P(A)P(C|A) = \frac{P(A|C)P(C)}{P(A)}P(C|A)=P(A)P(A|C)P(C)
- Претпоставува **условна независност** на атрибутите.

2. Како работи?

- За секоја класа CCC:
  1. Се пресметува P(A|C)P(A|C)P(A|C), P(C)P(C)P(C) и P(A)P(A)P(A).
  2. Класата со највисока веројатност P(C|A)P(C|A)P(C|A) е изборот.

3. Пример:

- Дадени се податоци за даночни измами (refund, статус, приход).

- За нов запис XXX: „Yes, Single, 80K“, пресметај $P(C=Yes|X)P(C=Yes|X)P(C=Yes|X)$ и $P(C=No|X)P(C=No|X)P(C=No|X)$.

4. **Лапласово мазнење (Laplace Smoothing):**
   - Се користи за да се избегнат веројатности од 0: $P(A_i=a|C=c)=N_{i,c}+1N_c+N_iP(A_i = a|C=c) = \frac{N_{i,c} + 1}{N_c + N_i}P(A_i=a|C=c)=Nc+Ni Ni,c+1$

5. **Примена:**
   - Анализа на текст (класификација на е-пораки).
   - Пример: Прогнозирање дали вест е за „Политика“ или „Спорт“.

Клучни споредби меѓу методите

Метод	Предности	Мани
Bagging	Намалување на варијансата, лесна имплементација	Тешка интерпретација

Метод	Предности	Мани
Random Forest	Намалена корелација, подобра точност	Може да стане бавен за многу дрва
Boosting	Ги поправа грешките, висока точност	Склон кон преклопување
Naïve Bayes	Брз, лесен за текстуални податоци	Претпоставката за независност

Совети за полагање:

- **Фокусирај се на примери:** Разбери како методите се користат за реални задачи.
- **Запомни формули:** Особено за Бејесовиот класификатор и Лапласовото мазнење.
- **Практична примена:** Разбери кога да користиш Bagging, Random Forest или Boosting.

Дали ти требаат дополнителни објаснувања или примери?

1. Ensemble Models: Bagging, Random Forest, Boosting

Ensemble Models се техники во машинското учење кои комбинираат повеќе модели за да создадат појака и попрецизна предвидувачка моќ. Главната идеја е дека група од слаби модели може да работи подобро од индивидуален силен модел.

Bagging (Bootstrap Aggregating)

- **Што е?** Метод каде повеќе модели се тренираат на различни подгрупи од податоците, добиени преку **повторно земање примероци со замена** (bootstrap).
- **Како работи?** Секој модел се тренира независно, а финалниот излез е просекот (за регресија) или гласањето (за класификација).
- **Пример:** Методот **Bagging** најчесто се користи со едноставни модели како Decision Trees.

- **Клучна предност:** Намалување на варијансата на моделот и избегнување на преклопување.

Пример за примена: Ако имаш сет на податоци за предвидување дали пациент има дијабетес, можеш да користиш bagging за да ги намалиш грешките од индивидуалните модели.

Random Forest

- **Што е?** Специфична имплементација на Bagging која користи **Decision Trees** како основни модели.
- **Како е различен од Bagging?** Random Forest додава случајност во изборот на променливи при секое гранче на дрвото, што го прави помалку склони кон преклопување.
- **Клучна карактеристика:** Го комбинира гласањето на повеќе дрва за конечна предикција.

Пример за примена: Анализа на ризик за кредитирање каде Random Forest го одредува дали клиентот е ризичен базирано на неговите финансиски податоци.

Boosting

- **Што е?** Техника каде моделите се тренираат последователно, и секој следен модел ги коригира грешките на претходните.
- **Како работи?**

1. Прв модел се тренира и се анализираат неговите грешки.
2. Следниот модел е трениран за да ги поправи овие грешки.
3. Процесот се повторува и финалната предикција е комбиниран излез од сите модели.

- **Популарни алгоритми за Boosting:**
  - **AdaBoost:** Работи со пондерирани грешки.
  - **Gradient Boosting:** Го минимизира градиентот на грешките.
  - **XGBoost:** Побрза и поефикасна варијанта на Gradient Boosting.

Пример за примена: Boosting често се користи во натпревари за предвидување, како Kaggle, каде што се бара максимална прецизност, на пример, во

предвидување на продажба базирано на историски податоци.

---

2. Bayes Classifier

Што е Bayes Classifier?

- Bayes Classifier е метод за класификација базиран на **Бејесовиот теорем**, кој ги користи веројатностите за донесување одлуки.
  - Формула за Бејесовиот теорем: P(C|X)=P(X|C)·P(C)P(X)P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)}P(C|X)=P(X)P(X|C)·P(C) каде што:
    - P(C|X)P(C|X)P(C|X) е веројатноста класата CCC да е точна за дадените податоци XXX.
    - P(X|C)P(X|C)P(X|C) е веројатноста за податоците XXX, ако класата е CCC.
    - P(C)P(C)P(C) е претходната веројатност за класата CCC.
    - P(X)P(X)P(X) е вкупната веројатност за податоците XXX.

Типови на Bayes Classifier:

- **Наивен Бајес (Naive Bayes):**
  - Претпоставува дека сите променливи се независни.
  - Брз, лесен за имплементација и ефективен кај текстуални податоци (на пр. анализа на емоции во пораки).

Пример: Ако имаме е-пораки и сакаме да одредиме дали тие се спам или не, Naive Bayes може да се користи за анализа на зборовите во пораката.

Предности на Bayes Classifier:

1. Брз и ефикасен за големи сетови на податоци.
2. Добро се справува со категоријални податоци.
3. Лесно се имплементира и толкува.

Мани на Bayes Classifier:

1. Претпоставката за независност кај Naive Bayes може да биде нереалистична.
2. Не работи добро со континуирани податоци без дополнителни претпоставки.

Пример за примена:

- Предвидување на болести базирано на симптоми.
- Анализа на кориснички коментари за класификација како позитивни или негативни.

4о