План-конспект лекції: Оцінювання ринкового потенціалу й конкурентного середовища

Навчальні цілі лекції:

- Ввести поняття **предиктивної аналітики**, її **визначення та мету**.
- Ознайомити з **основними етапами** процесу предиктивної аналітики.
- Представити **ключові типи та моделі** предиктивної аналітики, зокрема регресійні моделі (лінійна, логістична), дерева рішень та моделі часових рядів.
- Акцентувати на **призначенні, формулах, припущеннях та прикладах застосування** різних моделей.
- Ознайомити з **основними інструментами та середовищами**, що використовуються в предиктивній аналітиці (мови програмування, BI-інструменти, платформи).
- Визначити **основні виклики** в предиктивній аналітиці, такі як перенавчання та недонавчання, якість даних та упередженість, а також **шляхи їх подолання**.
- Продемонструвати **сфери застосування** предиктивної аналітики в різних галузях.

Очікувані навчальні результати:

- Студент **знає** визначення, мету та етапи предиктивної аналітики.
- Студент **знає** основні типи та моделі предиктивної аналітики (лінійна та логістична регресія, дерева рішень, моделі часових рядів) та їх базові принципи.
- Студент **знає** ключові інструменти та середовища для реалізації предиктивної аналітики.
- Студент **розпізнає** проблеми перенавчання та недонавчання, упередженості даних та знає методи їх запобігання.
- Студент **розуміє** важливість інтерпретації результатів та їх трансформації у бізнес-цінність.
- Студент **розуміє** широкі можливості застосування предиктивної аналітики у різних сферах бізнесу та IT.

Структура лекції:

- **Вступ до предиктивної аналітики**
    - Визначення та мета предиктивної аналітики
    - Етапи предиктивної аналітики
- **Основні типи та моделі предиктивної аналітики**
    - Регресійні моделі
        - Лінійна регресія
        - Логістична регресія
    - Дерева рішень
    - Моделі часових рядів

        - Основи часового ряду (компоненти)
        - Популярні моделі (AR, MA, ARIMA, SARIMA, Prophet)
- **Інструменти та середовища для предиктивної аналітики**
    - Мови програмування: Python, R
    - BI-інструменти та візуалізація: Excel, Power BI, Tableau
    - Середовища та платформи: Jupyter, Spark, AutoML, SAS, KNIME, Alteryx
- **Основні виклики та шляхи їх подолання в предиктивній аналітиці**
    - Перенавчання (Overfitting) та недонавчання (Underfitting) моделі
        - Причини та ознаки
        - Методи запобігання
    - Якість даних та упередженість
    - Інтерпретація результатів і бізнес-цінність
- **Сфери застосування предиктивної аналітики**
- **Висновок**

Очікувані теми попереднього опрацювання (prerequisites): Для успішного засвоєння матеріалу лекції студенти повинні мати такі попередні знання та вміння:

- **Базові знання математики:** основи лінійної алгебри, математичної статистики та теорії ймовірностей.
- **Основи програмування:** розуміння базових концепцій програмування, бажано знайомство з Python або R.
- **Базові знання аналізу даних:** розуміння концепцій даних, типів даних, первинної обробки даних.
- **Загальне розуміння штучного інтелекту та машинного навчання:** базове уявлення про те, що таке ШІ та машинне навчання, без глибокого занурення в алгоритми.
- **Навички роботи з табличними даними:** вміння працювати з даними в табличному форматі (наприклад, Excel).
- **Логічне мислення та аналітичні здібності:** здатність аналізувати інформацію та робити висновки.

Окрім цього, до цієї лекції студентам рекомендовано пройти **модулі 1 та 2** всеукраїнського курсу «Аналітика у продуктовому ІТ».

План-конспект лекції: Предиктивна аналітика

Вступ до предиктивної аналітики

Визначення та мета предиктивної аналітики

Предиктивна аналітика — це галузь аналізу даних, що зосереджується на прогнозуванні майбутніх подій на основі історичних даних. Вона дозволяє бізнесу передбачати ключові показники, як-от ціни чи попит, використовуючи методи штучного інтелекту, машинного навчання та статистики. Завдяки цим технологіям можливо виявляти закономірності в даних і будувати точні прогнози.

Етапи предиктивної аналітики

1. **Визначення завдання**

Формулювання мети й вимірюваних цілей моделі забезпечує узгодженість очікувань та ефективне використання ресурсів.

2. **Збір і підготовка даних**

Отримання якісних, репрезентативних даних із внутрішніх і зовнішніх джерел є критично важливим етапом, оскільки "Garbage in, garbage out", тобто «сміття на вході — сміття на виході».

3. **Попередня обробка даних**

Очищення, трансформація, нормалізація та інженерія ознак підвищують якість даних і надійність прогнозів.

4. **Розробка моделей**

Вибір моделей відповідно до типу даних і задачі, а також тренування та оптимізація з використанням сучасних ML-інструментів формують основу прогнозування.

5. **Оцінка результатів**

Перевірка моделі за релевантними метриками (MAE, RMSE, Precision, Recall тощо) дозволяє оцінити її точність і стабільність.

6. **Впровадження і моніторинг**

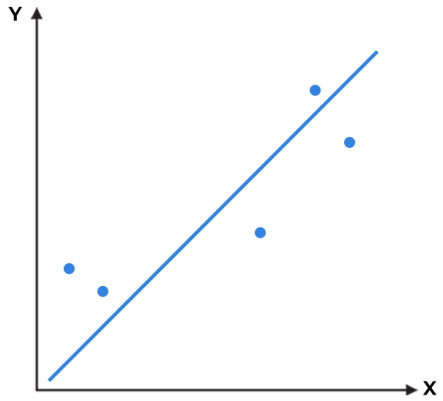
Інтеграція моделі у бізнес-процеси та подальший моніторинг забезпечують стійку продуктивність і адаптацію до змін у поведінці користувачів або ринку.

Основні типи та моделі предиктивної аналітики

Предиктивна аналітика використовує різноманітні моделі, які можна класифікувати за **типом прогнозованих значень** (неперервні або категоріальні) та **природою даних** (наприклад, часові ряди).

Регресійні моделі

Лінійна регресія



Джерело зображення

- **Призначення:** прогнозує неперервні числові значення, як-от кількість активних користувачів або очікуваний дохід застосунку.

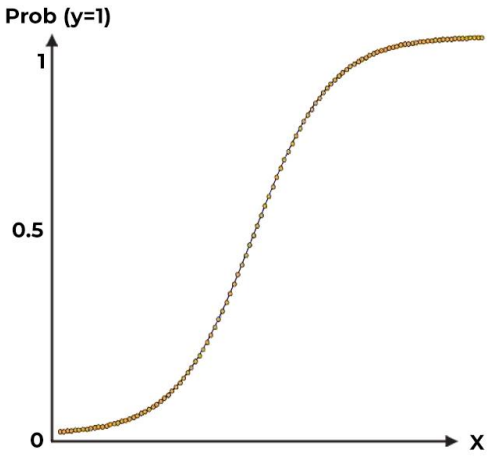
- **Формула:**
  $Y = \beta_0 + \beta_1 X_1 + ... + \beta_k X_k$,
  де Y — результат (наприклад, дохід), X — фактори (наприклад, кількість завантажень, час у додатку), β — коефіцієнти впливу.

- **Припущення:** існує лінійна залежність між змінними, предиктори не містять похибок.

- **Інтерпретація:** коефіцієнт β_1 означає, наскільки змінюється Y при збільшенні X_1 на одиницю, за незмінності інших змінних.

- **Приклади в IT:**
  Прогноз кількості нових підписок на SaaS-сервіс залежно від маркетингових витрат.
  Визначення, як зміна кількості функцій у застосунку впливає на середній час сеансу користувача.

Логістична регресія



Джерело зображення

- **Призначення:** класифікує об'єкти на категорії («так»/«ні»), наприклад, чи підпишеться користувач на платну версію.

- **Функція:** використовує сигмоїдну (логістичну) функцію:
  $P(x) = 1 / (1 + \exp(-x))$,
  яка перетворює результат на ймовірність (від 0 до 1).

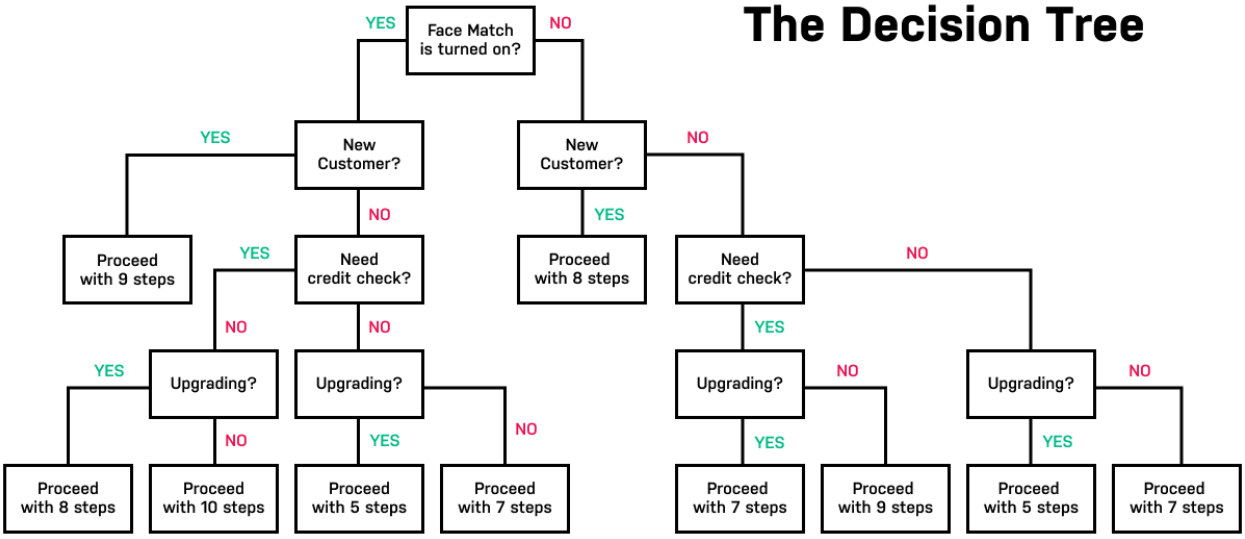
- **Поріг прийняття рішення:** якщо $P > 0{,}5$ → «так», інакше → «ні». Поріг можна адаптувати.

- **Приклади в IT:**
  Чи натисне користувач на push-сповіщення.
  Чи залишиться користувач після 7 днів у застосунку.

Дерева рішень

- **Суть:** алгоритм прийняття рішень у вигляді дерева з умовами типу «якщо-то».
  Вузли — умови, гілки — наслідки, листя — фінальний прогноз або клас.

- **Критерії розбиття:**
  Ентропія — вимірює невизначеність.
  Коефіцієнт Джині — оцінює чистоту розподілу класів.



[Джерело прикладу]

Переваги:

- Інтерпретованість, робота з числовими та категоріальними даними, мінімальна підготовка даних.

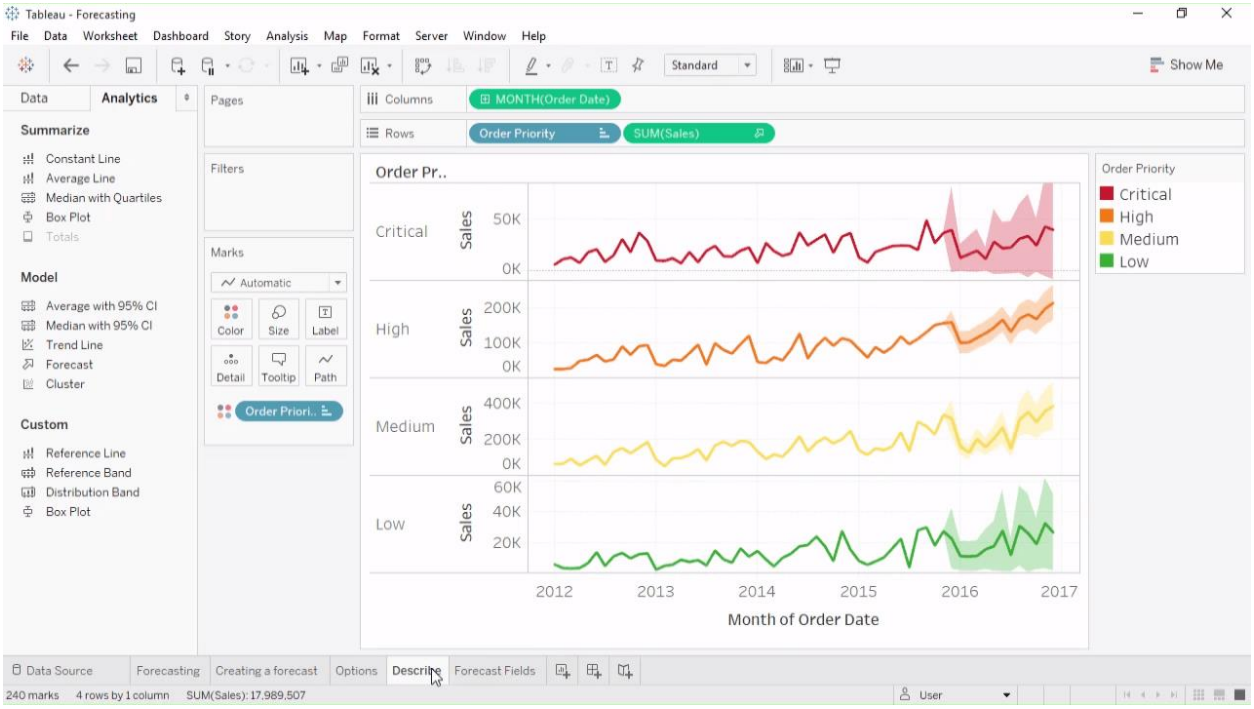
Недоліки:

- Нестабільність: незначні зміни в даних можуть сильно змінити структуру дерева.
- Схильність до перенавчання.
- Не завжди знаходять оптимальне рішення через жадібну стратегію побудови.

Приклади в IT:

- Прогноз, чи скасує користувач підписку на стримінговий сервіс на основі його поведінки.
- Класифікація відгуків користувачів як «негативний», «нейтральний», «позитивний».

Моделі часових рядів



[Джерело зображення]

Моделі часових рядів прогнозують майбутні значення на основі даних, впорядкованих у часі (наприклад, щоденна активність користувачів або кількість запитів до сервера).

Основи часового ряду

Часовий ряд — це послідовність спостережень, зроблених у рівні проміжки часу.

Компоненти:

- **Тренд** — довгострокова зміна (зростання/спад) значень.
- **Сезонність** — періодичні коливання (наприклад, пік трафіку у вихідні).
- **Автокореляція** — залежність поточних значень від попередніх (часова «пам’ять» даних).

Розклад часового ряду на компоненти допомагає краще зрозуміти структуру даних і вибрати відповідну модель. Проста регресія тут часто неефективна.

Популярні моделі

- **AR (Авторегресія).** Прогноз базується на власних минулих значеннях. **Приклад:** прогноз щоденної кількості користувачів на основі значень за попередні 7 днів.
- **MA (Ковзне середнє).** Використовує попередні помилки прогнозів для уточнення майбутніх значень. **Приклад:** прогноз кількості помилок у логах системи на основі відхилень попередніх прогнозів.
- **ARIMA.** Комбінує AR і MA та додає диференціювання для усунення тренду. Формат: ARIMA(p,d,q). **Приклад:** прогноз навантаження на сервер при складному змінному трафіку.
- **SARIMA.** Розширення ARIMA для обробки **сезонності**. Формат: ARIMA(p,d,q)(P,D,Q)_m, де *m* — довжина сезону. **Приклад:** щомісячний попит на хмарні ресурси з повторюваними піками.
- **Prophet (Facebook).** Модель із простою конфігурацією, здатна обробляти **нелінійні тренди**, **свята**, **декілька сезонних циклів**. **Перевага:** висока точність та мінімальна потреба в ручному налаштуванні. **Приклад:** прогноз обсягів транзакцій у фінтех-застосунку або активності користувачів у соціальних мережах.

Інструменти та середовища для предиктивної аналітики

Мови програмування: Python, R

Python — основна мова для аналітики завдяки гнучкості та бібліотекам:

- **Scikit-learn** — стандарт для машинного навчання (лінійна/логістична регресія, дерева рішень).
- **Statsmodels** — статистичне моделювання та тестування.
- **NumPy, Pandas** — обробка й аналіз даних.

R — мова для статистики та візуалізації з широкою бібліотекою пакетів (CRAN). **Python та R** — ключові інструменти завдяки відкритості, гнучкості та підтримці сучасних алгоритмів.

BI-інструменти та візуалізація: Excel, Power BI, Tableau

- **Excel** — базовий інструмент для обробки та попереднього аналізу даних.
- **Power BI** — зручне рішення для інтерактивної візуалізації та створення дашбордів.
- **Tableau** — інструмент для швидкої, глибокої та інтерактивної візуалізації.

BI-інструменти важливі для дослідження даних і донесення результатів до нетехнічних користувачів.

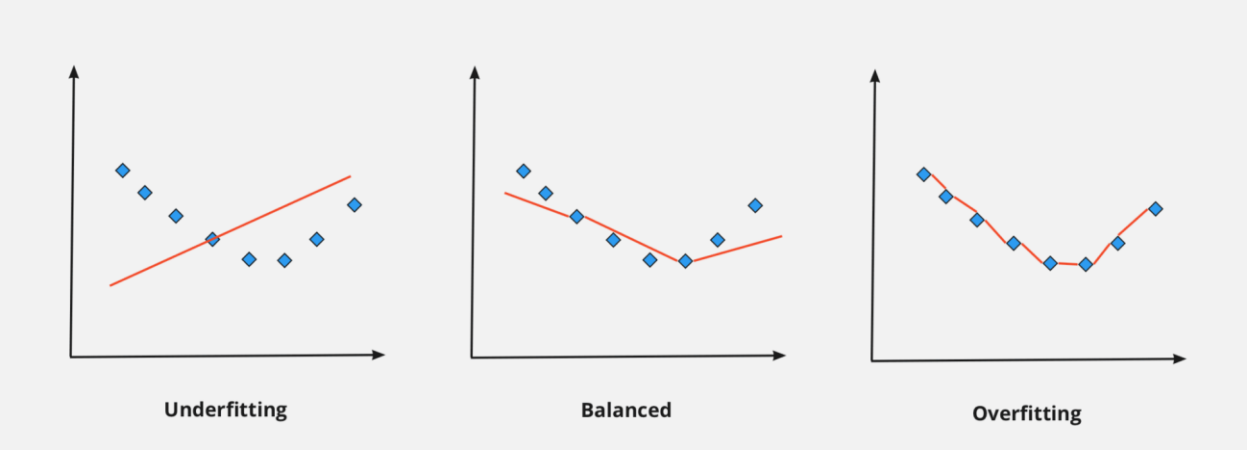
Середовища та платформи: Jupyter, Spark, AutoML, SAS, KNIME, Alteryx

- **Jupyter Notebook** — інтерактивне середовище для Python/R з підтримкою коду, графіки та описів.
- **Apache Spark** — обробка великих даних у розподілених системах.
- **Google Cloud AutoML** — автоматизоване створення моделей без глибокого кодування.
- **SAS** — класичне рішення для аналітики та прогнозування.
- **KNIME, Alteryx** — платформи з інтерфейсами low-code для машинного навчання та аналізу даних.

Вибір інструментів залежить від цілей, навичок команди та масштабів завдання — від гнучких кодових підходів до готових платформ для швидкого впровадження.

Основні виклики та шляхи їх подолання в предиктивній аналітиці

Перенавчання (Overfitting) та недонавчання (Underfitting) моделі



Джерело зображення

Перенавчання (overfitting) — модель занадто точно відтворює навчальні дані (включно з шумом), але погано узагальнює нові.
Недонавчання (underfitting) — модель надто спрощена, не вловлює закономірностей і має низьку точність на будь-яких даних.
Причини: занадто складна або надто проста модель, недостатньо або низька якість даних, нерелевантні ознаки.

Ознаки: велика різниця між точністю на тренувальних і тестових даних (перенавчання) або однаково низька точність на обох (недонавчання).

Методи запобігання

- **Крос-валідація:** оцінює модель на різних підмножинах даних для покращення узагальнення.
- **Регуляризація (L1, L2):** додає штраф за складність моделі, зменшуючи ваги параметрів.
  Обрізання дерева: скорочує надлишкові гілки в деревоподібних моделях.
- **Оптимальна складність:** налаштування архітектури моделі для уникнення надмірної або недостатньої складності.
- **Рання зупинка:** припиняє тренування, коли валідаційна помилка зростає.
- **Збільшення обсягу даних:** додає нові приклади або створює варіації існуючих.
- **Ансамблеві методи:** поєднують кілька моделей (наприклад, Random Forest, Gradient Boosting) для підвищення стабільності.

Якість даних та упередженість

Погані або упереджені дані можуть призвести до неточних або несправедливих рішень моделі.
Приклад: модель, навчена на упереджених даних, може дискримінувати певні групи.
Рішення: контроль за джерелами даних, очищення, забезпечення різноманітності та прозорості в процесі підготовки.

Інтерпретація результатів і бізнес-цінність

Робота аналітика охоплює не лише побудову моделі, а й пояснення результатів і їхню трансформацію в **бізнес-рішення**.
Цінність аналітики проявляється тоді, коли висновки допомагають прогнозувати попит, оптимізувати запаси або зрозуміти поведінку клієнтів.

Ключ: технічна точність має сенс лише за умови грамотної інтерпретації та комунікації.

Сфери застосування предиктивної аналітики

Предиктивна аналітика — це дієвий інструмент для прогнозування майбутніх подій на основі даних, який допомагає організаціям діяти проактивно.

- **Маркетинг:** Прогнозування поведінки клієнтів, персоналізація кампаній, вибір оптимальних каналів комунікації, аналіз відгуків.
- **Фінанси:** Виявлення шахрайства, оцінка кредитоспроможності, прогнозування цін.
- **Роздрібна торгівля:** Прогнозування попиту, оптимізація запасів і цін, персоналізація досвіду покупців.
- **HR:** Оцінка ризику звільнення співробітників, утримання талантів.
- **IT-безпека:** Виявлення аномалій у трафіку, попередження кібератак.

Висновок: Предиктивна аналітика — стратегічний інструмент, який підвищує дохід, знижує втрати та покращує ефективність у всіх бізнес-функціях.

Ключові аспекти застосування

- **Дані та підготовка:** якість, обробка й інженерія ознак визначають ефективність моделей.
- **Вибір моделей:** регресія для чисел, класифікація — для категорій, часові ряди — для даних у часі.
- **Інструменти:** від Python та R до BI-рішень (Power BI, Tableau) та low-code платформ (AutoML, KNIME, Alteryx).
- **Оцінка моделей:** критично важлива для надійності. Використовуються метрики, адаптовані до контексту завдання.
- **Запобігання помилкам:** крос-валідація, регуляризація, рання зупинка допомагають уникнути перенавчання/недонавчання.

Цінність для бізнесу

Предиктивна аналітика — це не лише побудова моделей, а й **перетворення прогнозів на дії**. Ефективна комунікація результатів і глибоке розуміння контексту забезпечують прийняття обґрунтованих рішень у маркетингу, фінансах, охороні здоров’я та інших галузях.

Список додаткових джерел для самостійного опрацювання

1. Descriptive, Prescriptive, and Predictive Analytics
2. Лінійні регресії
3. Steps to Create Predictive Models
4. Що таке лінійна регресія: повний гайд від robot_dreams
5. Cross-Validation Using K-Fold With Scikit-Learn
6. Sklearn Linear Regression (Step-By-Step Explanation)
7. Mastering Logistic Regression
8. Know The Best Evaluation Metrics for Your Regression Model
9. Regression Metrics: MSE, RMSE, MAE, and R-squared
10. F1 Score vs ROC AUC vs Accuracy vs PR AUC: Which Evaluation Metric Should You Choose?
11. Класифікація: точність, повнота, влучність і пов’язані метрики
12. Simple Linear Regression | An Easy Introduction & Examples
13. What Is Logistic Regression?
14. Logistic regression: Definition, Use Cases, Implementation
15. Алгоритми машинного навчання. Глибокі нейромережі в задачах механіки суцільних середовищ.
16. SkLearn Decision Trees: Step-By-Step Guide
17. Top Predictive Analytics Models & Algorithms
18. ARIMA & SARIMA: Real-World Time Series Forecasting
19. ARIMA — sktime documentation
20. Data Analysts' Toolbox - Excel, Power BI, Python, & Tableau
21. Linear Regression in Python using Statsmodels
22. MAE, MAPE, MASE and the Scaled RMSE
23. Overfitting and Underfitting
24. Underfitting and Overfitting in Machine Learning
25. Using Predictive Analytics to Reduce Dropout Rates in Online Courses
26. Онлайн-курс: **Stanford CS229: Machine Learning Full Course**