✅ الصفحة 4 – What Is K-Means?

ما هي خوارزمية K-Means؟

• خوارزمية تعلم غير مُراقَب تُقسّم البيانات إلى K مجموعات (Clusters).
• الهدف: تقليل المسافة بين كل نقطة ومركز مجموعتها (Centroid).
• تحاول جعل المجموعات متماسكة ومنفصلة قدر الإمكان.
• يتم حساب دالة الهدف التي تقلل مجموع المسافات داخل كل مجموعة.

✅ الصفحة 5 – Example

مثال على K-Means

بيانات العملاء تحتوي على:
• العمر
• الدخل السنوي

قد تقسمهم إلى 3 مجموعات: K-Means
• شباب ذوي دخل منخفض
• متوسطو العمر بدخل متوسط
• كبار بدخل مرتفع

✅ الصفحة 6 – How K-Means Works?

كيف تعمل K-Means؟

الخطوات:

1. **تهيئة المراكز عشوائياً**
2. **تخصيص النقاط** لأقرب مركز
3. **إعادة حساب المراكز** حسب متوسط النقاط
4. **التكرار** حتى تثبت المراكز

الهدف: جعل كل مجموعة تحتوي على نقاط متشابهة أكثر من نقاط المجموعات الأخرى.

✅ الصفحة 7 – Euclidean Distance

المسافة الإقليدية

الطريقة الأكثر استخداماً لقياس المسافة بين نقطتين.

● 2 فيD:
$$ d = \sqrt{(x2-x1)^2 + (y2-y1)^2} $$

● 3 فيD:
$$ d = \sqrt{(x2-x1)^2 + (y2-y1)^2 + (z2-z1)^2} $$

هي ببساطة "مسافة الخط المستقيم".

✅ الصفحة 9 – Challenge of k

تحدي اختيار عدد المجموعات k

• إذا اخترنا **k** صغيرة جداً ← دمج مجموعات مختلفة (Underfitting)
• إذا اخترنا **k** كبيرة جداً ← تقسيم زائد (Overfitting)
• الحل الشائع: **طريقة Elbow**

✅ الصفحة 10 – Elbow Method

طريقة Elbow – الخطوات

1. تشغيل K-Means بقيم k مختلفة
2. حساب قيمة WCSS
3. رسم WCSS مقابل k
4. اختيار نقطة الانحناء (Elbow) كأفضل قيمة k

✅ الصفحة 11 – Notes

ملاحظات إضافية

• قد لا يكون الـ Elbow واضح أحياناً
• طرق أخرى:
– Silhouette Score
– Gap Statistic

✅ الصفحة 14 – Strengths of K-Means

مميزات K-Means

• سريع وفعال
• بسيط وسهل الفهم
• جيد للمجموعات الكروية المتماسكة
• يضمن الوصول لحل (Convergence)

✅ الصفحة 15 – Limitations of K-Means

عيوب K-Means

• يجب تحديد k مسبقاً
• حساس للنقاط الشاذة
• حساس لبدء المراكز
• لا يعمل جيداً للمجموعات غير الكروية أو ذات الكثافة المختلفة

✅ الصفحة 16 – Why K-Means++

لماذا نستخدم K-Means++؟

• حل مشكلة اختيار المراكز العشوائي
• يجعل النتائج أفضل وأسرع
• يقلل عدد التكرارات مقارنة بـ K-Means العادي

✅ الصفحة 17 – How K-Means++ Works

كيف يعمل K-Means++؟

الخطوات:

1. اختيار مركز أول عشوائي
2. حساب المسافة لبقية النقاط
3. اختيار مراكز بعيدة بشكل احتمالي
4. بعد اختيار k نكمل ← مراكز K-Means
   ▪ يجعل التجميع أكثر جودة

✅ الصفحة 19 – What is DBSCAN?

ما هي خوارزمية DBSCAN؟

• خوارزمية قائمة على الكثافة
• تحدد المجموعات حسب المناطق التي تحتوي على كثافة نقاط عالية
• تحدد النقاط غير المنتمية لأي مجموعة كـ **ضوضاء** **(Noise)**

✅ الصفحة 20 – DBSCAN Parameters

معايير DBSCAN

1. **ε (Epsilon)** – نصف قطر الجوار
2. **MinPts** – الحد الأدنى لعدد النقاط لتشكيل مجموعة
   ▪ قاعدة شائعة: MinPts ≥ عدد الأبعاد + 1

✅ الصفحة 21 – Core / Border / Noise

أنواع النقاط في DBSCAN

• **Core Point**: لديها MinPts داخل ε
• **Border Point**: بجوار نقطة Core لكن لا تملك كثافة كافية
• **Noise Point**: لا تنتمي لأي مجموعة

✅ الصفحة 22 – How DBSCAN Works

كيف تعمل DBSCAN؟

1. تحديد النقاط الأساسية
2. توسيع المجموعات حسب الاتصال بالكثافة
3. النقاط غير المتصلة ← ضوضاء
   ▪ لا تحتاج عدد مجموعات مسبق

✅ الصفحة 26 – Strengths

مميزات DBSCAN

• يتعامل مع الضوضاء جيداً
• لا يحتاج تحديد عدد مجموعات
• يناسب الأشكال غير المنتظمة
• لا يفرض انتماء النقاط الشاذة لأي مجموعة

✅ الصفحة 27 – Limitations

عيوب DBSCAN

• حساس جداً لقيمة ε و MinPts
• غير مناسب للكثافات المختلفة
• لا يعمل جيداً مع البيانات عالية الأبعاد
• أبطأ من K-Means للبيانات الكبيرة

✅ الصفحة 29 – When to Use

متى نستخدم K-Means أو DBSCAN؟

K-Means

• عندما نعرف عدد المجموعات
• عندما تكون المجموعات كروية ومتساوية
• البيانات قليلة الضوضاء

DBSCAN

• عندما لا نعرف عدد المجموعات
• عندما تكون الأشكال غير منتظمة
• عندما يوجد ضوضاء كثيرة
• Spatial مناسب للبيانات المكانية

✅ الصفحة 30 – Key Differences

الفروق الأساسية

الجانب	K-Means	DBSCAN
شكل المجموعات	كروية	أي شكل
عدد المجموعات	يجب تحديده	يُستنتج تلقائياً
الضوضاء	تُجبر داخل مجموعة	تُوسم بضوضاء
المعايير	k + initialization	ε + MinPts
السرعة	أسرع	أبطأ

✅ الصفحة 31 – What They Handle

ما الذي يمكن لكل منهما التعامل معه؟

• المجموعات الكروية ← K-Means
• المجموعات غير الكروية ← DBSCAN
• الكثافات المختلفة ← كلاهما يعاني
• الضوضاء ← DBSCAN يتفوق