🎬 Transcript: Hyperparameter Tuning – Grid Search vs. Random Search

By Arpita Thokal | Student ID: 24005907

⏱️ 00:00 – Slide 1: Title Slide

"Hi everyone! My name is Arpita Thokal, and welcome to my tutorial on *Hyperparameter Tuning: Grid Search vs. Random Search*.
In this short video, I'll walk you through what hyperparameters are, why they matter, and how to tune them effectively using two popular methods.
So let's dive in and find the best hyperparameters — efficiently!"

⏱️ 00:28 – Slide 2: Agenda

"Here's a quick overview of what we'll cover.
We'll start with a quick introduction to hyperparameter tuning.
Next, we'll break down two widely used tuning methods: **Grid Search** and **Random Search**.
I'll show you a hands-on implementation using Python on the Titanic dataset using a Random Forest model, compare the results, and wrap up with a few key takeaways.
Let's get started!"

⏱️ 00:57 – Slide 3: What is Hyperparameter Tuning?

"Let's start by understanding what hyperparameters are and why they're important.
Hyperparameters are configuration settings that control how a machine learning model learns.
They are not learned from the data and must be set before training.
If set poorly, a model might underfit or overfit, leading to poor performance on new data.
Hyperparameter tuning helps us find the best combination for optimal results."

⏱️ 01:26 – Slide 4: Examples of Hyperparameters

"Random Forest is a machine learning model that uses many decision trees to improve predictions.
In Random Forest, hyperparameters include:
✔️ n_estimators – the number of trees
✔️ max_depth – how deep each tree can go

✔️ min_samples_split – controls when to split a node
If these are set poorly, we again risk underfitting or overfitting."

⏱️ 01:55 – Slide 5: Cooking Analogy

"Think of hyperparameter tuning like cooking.
Too much spice or too little salt ruins the dish.
Similarly, poorly tuned hyperparameters can ruin model performance."

⏱️ 02:24 – Slide 6: What is Grid Search?

"Grid Search is a brute-force approach.
It tests **all** combinations of hyperparameter values to find the best.
Imagine trying every possible combination of dishes in a restaurant — it's effective but takes time.
That's Grid Search."

⏱️ 02:53 – Slide 7: How Grid Search Works

"Here's how it works:
Let's say we try 3 values for n_estimators and 3 for max_depth.
That gives us 3 × 3 = 9 combinations.
Grid Search trains and evaluates a model for **each** of those combinations."

⏱️ 03:21 – Slide 8: Grid Search Grid Layout

"This grid shows the 9 combinations visually.
Even if one hyperparameter doesn't affect performance much, Grid Search still tests it — which can be inefficient and slow."

⏱️ 03:50 – Slide 9: What is Random Search?

"Random Search takes a smarter shortcut.
Instead of testing every combination, it selects a few random ones and tests just those.
This saves time and resources."

⏱️ 04:19 – Slide 10: Lottery Analogy

"Think of Random Search like picking lottery tickets.
Each ticket is a hyperparameter combo.
You don't need to try all of them — just a few random ones to get a good result."

⏱️ 04:48 – Slide 11: How Random Search Works

"In Random Search, we still define ranges like:
✔️ n_estimators = np.arange(50, 200, 50)
✔️ max_depth = [None, 10, 20]
But we might test only 5 combinations instead of all 9 — making it much faster."

⏱️ 05:17 – Slide 12: Grid vs. Random Search

"Let's compare.
Grid Search is thorough and reliable, but slow and resource-heavy.
Random Search is faster, efficient, and often just as effective — though it doesn't always find the absolute best combination."

⏱️ 05:46 – Slide 13: Implementation in Python

"Now let's look at how we implemented this in Python.
First, I imported the necessary libraries.
Then I loaded the Titanic dataset and selected relevant features.
I handled missing values, encoded categorical columns, and split the data into training and test sets.
I trained a baseline Random Forest and got 81% accuracy.
Then I used Grid Search, which evaluated 27 combinations — took 14.47 seconds and improved accuracy to 82.12%.
Next, I used Random Search with just 5 combinations — and it matched Grid Search's accuracy but only took 5.12 seconds."

⏱️ 06:15 – Slide 14: Performance Comparison

"Both Grid Search and Random Search gave us the same accuracy — around 82%.
But Random Search was 3 times faster.
So, for this task, Random Search was the more efficient choice."

⏱️ 06:43 – Slide 15: Final Takeaways

"To wrap up:
✅ Hyperparameter tuning improves model performance.
✅ Grid Search is exhaustive but slow.
✅ Random Search is faster and often just as effective.
Choose your method based on time, data size, and complexity."

⏱️ 07:12 – Slide 16: Thank You

"Thank you for watching!
I hope this tutorial helped you understand how to tune hyperparameters using Grid and Random Search.
You can find all the code and materials on my GitHub — linked below.
Happy tuning!"