Google Developer Experts

ML Algorithms and Hyperparameter Model Tuning

Tarun Jain
GDE in AI/ML and Super Agent
@TRJ_0751- Twitter

ai planet

$whoami

- Super Agent at AI Planet
- Google Developer Expert in AI
- Google Summer of Code 2024 Red Hen Lab
- Google Summer of Code 2023 caMicroscope
- AI with Tarun - YouTube Content Creator
- Watch Anime and Read Manga in free time.

Agenda

- Intuition for Feature engineering and Feature Selection
- ML Algorithms- Different model building techniques
- Approaches for Hyperparameter Model Tuning
- Tip of the day
- Code Demo
- QA

How many of you watch Netflix Series or Movies?

What's your favourite series?


THE
SOC|ETY
MAY 10 | NETFLIX

YOU SAY APOCALYPSE LIKE IT'S A BAD THING.
DAYBREAK
OCT 24 | NETFLIX

BASED ON TRUE EVENTS
NO ONE BELIEVED
A NETFLIX LIMITED SERIES
UNBELIEVABLE
TONI COLLETTE
MERRITT WEVER
KAITLYN DEVER
SEPT 13 | NETFLIX

GOAL:
Netflix Show Cancellation Prediction

Why wasn’t there a Season 2?

Feature matters in finding patterns

Viewership Metrics:

- Total unique viewers (first 28 days)
- Episode completion rates (% who finish each episode)
- Series completion rate (% who finish entire season)
- Average watch time per episode
- Rewatching frequency
- International vs domestic viewer split

Production & Content:

- Production budget per episode
- Genre classification
- Episode count in season 1
- Runtime per episode
- Release date
- Language/country of origin

Engagement Metrics:

- Social media mentions (Twitter, Instagram, TikTok)
- User ratings (thumbs up/down)
- Time spent browsing show details
- Trailer view counts
- Search frequency for show name

Feature Engineering and Feature Selection

Original Features

- **show_id** - Unique identifier for each Netflix show (e.g., "NS_001", "NS_002")
- **viewer_completion_rate** - % of viewers who finished the entire season
- **total_viewers** - Total unique viewers in first 28 days
- **international_viewership_ratio** - International viewers / Total viewers
- **genre** - Show genre (Drama, Comedy, Thriller, etc.)
- **production_cost_per_episode** - Budget per episode in millions
- **social_media_mentions** - Total mentions across Twitter, Instagram, TikTok

Feature Engineering- New features

- **cost_efficiency_score**:

viewer_completion_rate × total_viewers / production_cost_per_episode

- **global_appeal_index:**

international_viewership_ratio × social_media_mentions / total_viewers

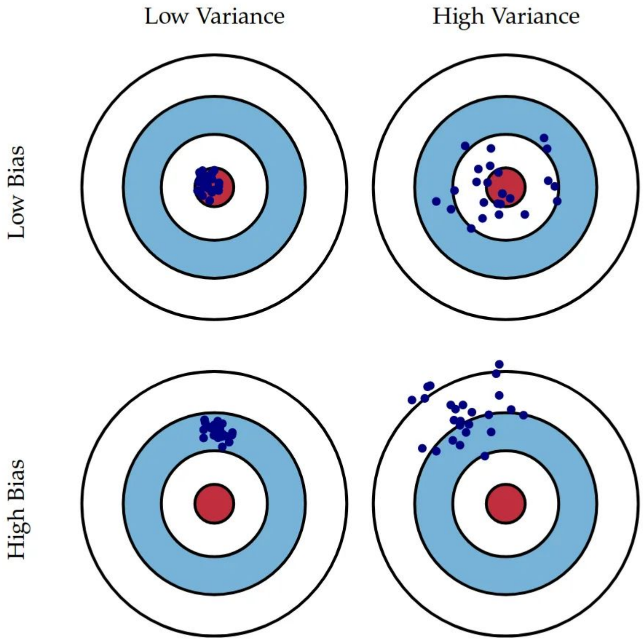
Feature Selection

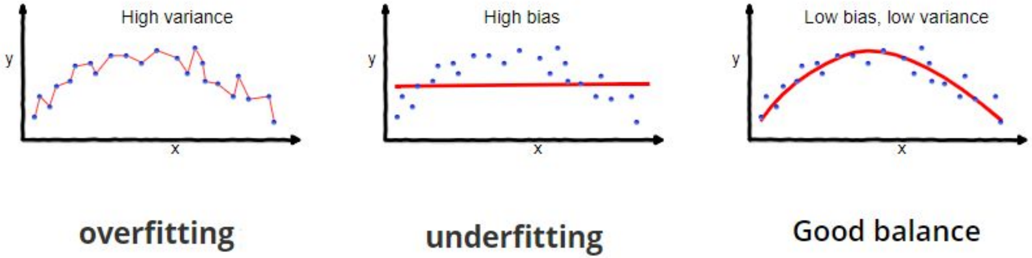
- **social_media_mentions**
- **cost_efficiency_score**
- **viewer_completion_rate**
- **genre**
- **production_cost_per_episode**
- **international_viewership_ratio**

Aspect	Feature Engineering	Feature Selection	Feature Extraction
Purpose	Create better features	Choose best features	Reduce feature dimensions
Input/Output	6 features → 7 features	7 features → 5 features	100 features → 10 components
Interpretability	High (we know what each feature means)	High (original features)	Low (components are abstract)
Domain Knowledge	Required (business understanding)	Helpful (know what's important)	Not required (mathematical)

Let's talk about Model building

Machine Learning Algorithms- when how and why?





Combine multiple weak learners to reduce both:

- **Bias**: By combining diverse models (e.g., Gradient Boosting).
- **Variance**: By averaging predictions (e.g., Random Forest, Bagging).

Ensemble Technique- Tree Based

Bagging - Strong Learners:

- Uses complex models (deep trees, SVMs, neural networks)
- Reduces their high variance through averaging. Bagging primarily aims to reduce variance and prevent overfitting.
- **E.g., Random Forest:** 100 deep decision trees, each trained on bootstrap samples, final prediction by majority vote

Boosting - Weak Learners:

- Uses simple models (decision stumps, shallow trees)
- Combines many weak learners to create strong ensemble. It focuses on reducing bias. High bias- underfitting
- **E.g., AdaBoost:** 50 decision stumps trained sequentially, each correcting previous errors, weighted combination

Stacking - Mixed/Diverse Learners:

- Uses different types of models (both strong and weak)
- Combines Random Forest + SVM + Neural Network + Logistic Regression
- Diversity is key - different model types capture different patterns
- **E.g., Stacking Ensemble:** Level 1 has Random Forest + XGBoost + SVM, Level 2 uses Logistic Regression to learn optimal combination weights

Gradient Boosting Machines

XGBoost (Extreme Gradient Boosting):

- Uses gradient boosting with regularization
- Builds trees sequentially, each correcting residuals of previous trees

LightGBM (Light Gradient Boosting Machine):

- Uses Leaf-wise Growth i.e., continues prioritizing high-delta-loss leaves for faster convergence.
- Based on Histogram-based algorithms reduce memory usage and accelerate training.
- Uses GOSS (Gradient-based One-Side Sampling) and EFB (Exclusive Feature Bundling) for hard samples and feature compression.

CatBoost (Categorical Boosting):

- Handles categorical features automatically without preprocessing
- Uses ordered boosting to reduce overfitting

Neural Network

For Tabular Data

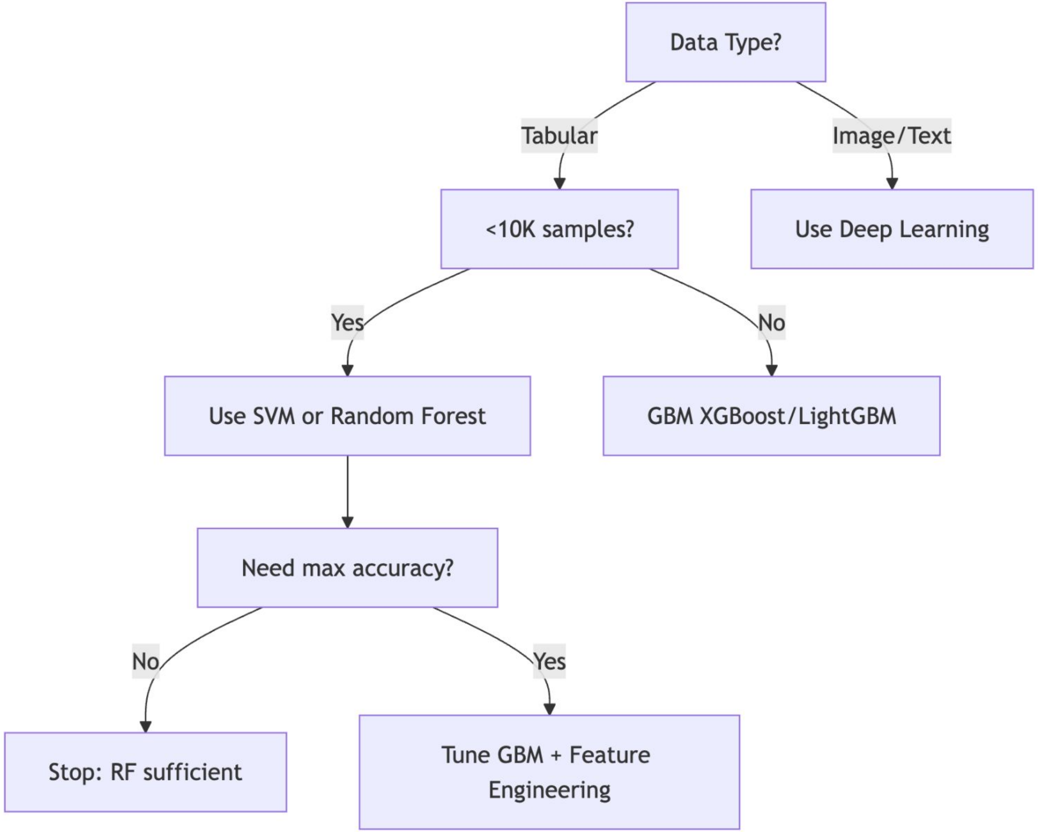
- Combines wide linear models (memorization) with deep neural networks (generalization).
- For tabular data, e.g., recommendations, classification, regression.
- ANN (Artificial Neural Networks) / Feed-Forward:
  - Learns non-linear patterns via layered neurons.
  - General tabular classification/regression.

For Sequential Data

- RNN with gates i.e., LSTM/GRU
- For short to moderate sequences i.e., mainly text data.
- Use Transformers for the long sequences.

For Image Data

- For image classification, feature extraction- ResNet, VGG16- CNN model.
- Real-time object detection via a single regression pass - Yolo: DarkNet.



Use Cases

Credit Scoring (Random Forest):

- Why: Handles 100+ financial features with missing values.
- Winning Trait: Built-in feature importance detects income/debt ratio dominance

E-commerce Fraud Detection (XGBoost):

- Why: Captures complex interactions like (device_type=mobile) & (purchase_velocity > $1000/hr)
- Edge: Custom loss functions optimize for $ recovery

Medical Diagnosis (LightGBM):

- Why: Integrates lab values (continuous) + symptoms (categorical)
- Advantage: Leaf-wise growth detects rare disease patterns

Hyperparameter Model Tuning

Choose the best parameters to train the model

Grid Search:
The Brute-Force Approach

- Imagine you're trying to find the perfect pizza recipe.
- You systematically test every combination of dough thickness (thin, medium, thick) and baking temperature (400°F, 450°F, 500°F).
- Grid Search works exactly like this - it methodically tries every possible combination of hyperparameters from your predefined options and picks the winner.

```
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier()
param_grid = {'n_estimators': [50, 100, 200],
              'max_depth': [3, 5, 7]}

grid_search = GridSearchCV(model, param_grid, cv=5)
grid_search.fit(X_train, y_train)

best_params = grid_search.best_params_
```

Random Search: The Experimental Bet

- Now imagine you randomly pick pizza combinations from your ingredient ranges.
- Maybe you try thin crust at 475°F, then thick crust at 425°F, then medium at 510°F.
- You're not making every possible pizza - just random combinations from your ranges.
- Random Search works the same way - it randomly samples hyperparameter combinations instead of testing everything.

```
from sklearn.model_selection import RandomizedSearchCV
from scipy.stats import randint
from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier()
param_dist = {'n_estimators': randint(50, 200),
               'max_depth': randint(3, 10)
            }

random_search = RandomizedSearchCV(model, param_dist,
n_iter=20, cv=5)
random_search.fit(X_train, y_train)

best_params = random_search.best_params_
```

Bayesian Optimization: The Smart One

- Imagine you have a assistant who remembers every pizza you've made and how good it tasted.
- After seeing that thin crust at high temperature was delicious, they intelligently suggest "try thin crust at an even higher temp" or "maybe thin crust with different toppings."
- Bayesian Optimization works like this assistant - it learns from each pizza attempt and cleverly suggests the next recipe to try.

```
import optuna
from sklearn.ensemble import RandomForestClassifier


def objective(trial):
    n_est = trial.suggest_int('n_estimators', 50, 200)
    max_d = trial.suggest_int('max_depth', 3, 10)
    model = RandomForestClassifier(
                                n_estimators=n_est,
                                max_depth=max_d)
    return cross_val_score(model,
                              X_train,
                              y_train,
                              cv=5).mean()


study = optuna.create_study(direction='maximize')
study.optimize(objective, n_trials=20)
```

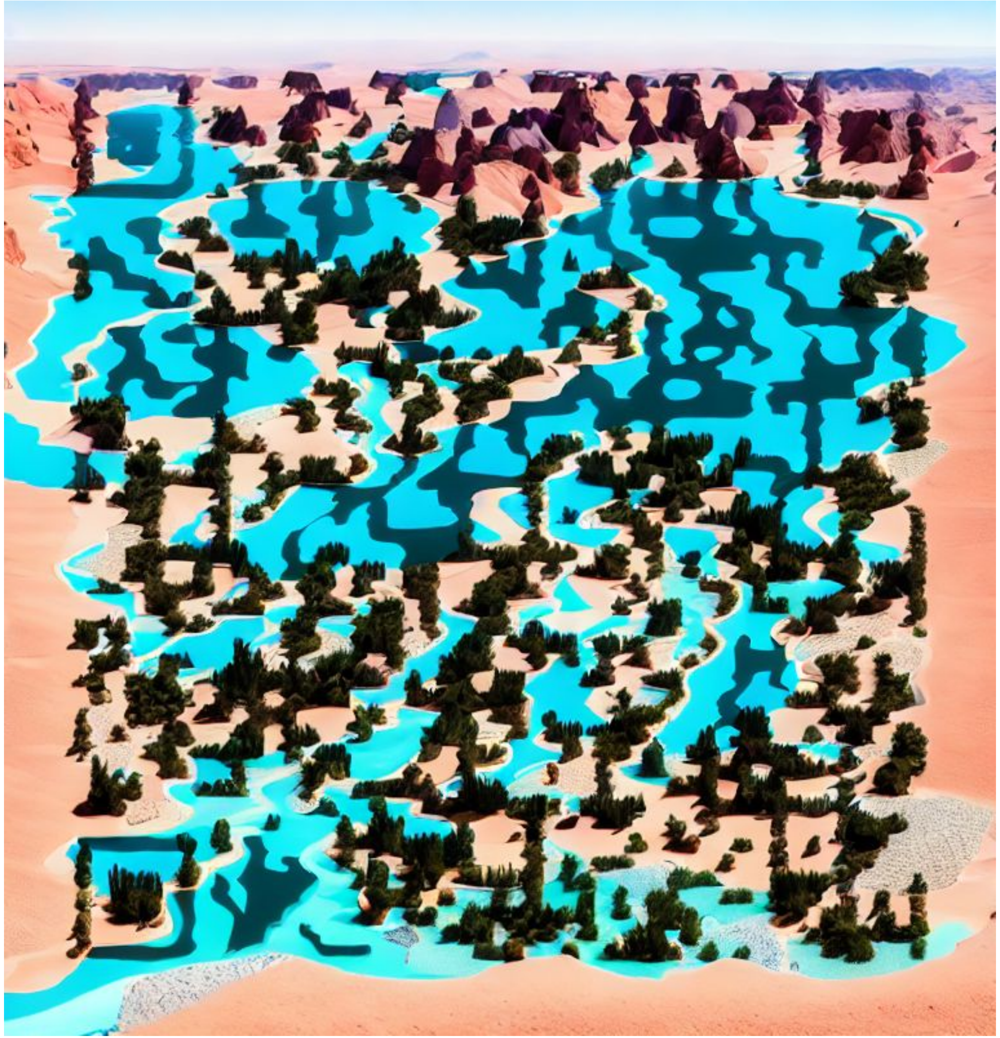
Tip of the Day:

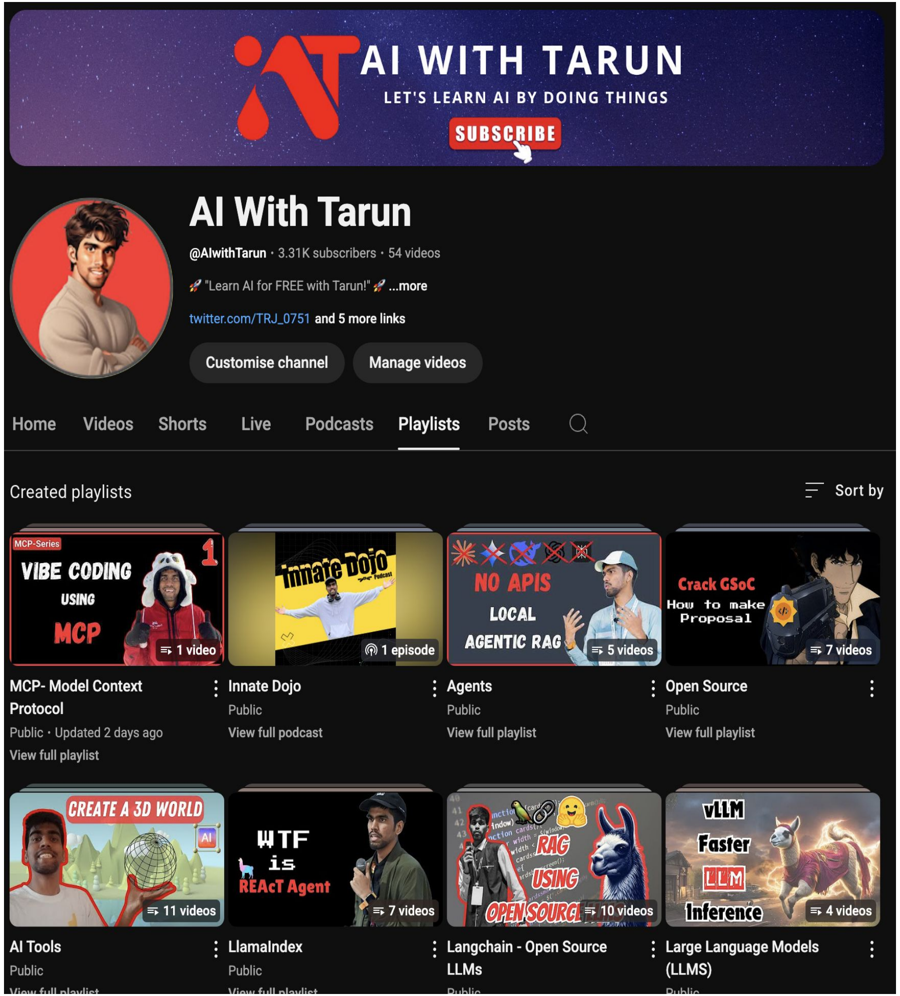
- Experimentation is key
- Define Seed
- Success metrics as per the problem statement
- Spend time performing EDA
- Remember:
    - The best model is the simplest one that solves your business problem.
    - Complexity should be justified by significant performance gains, not just because you can.
- Refer to Kaggle top notebooks and refer to their EDA and model building approach.

bit.ly/aiplanet-discord

- AI Planet - LinkedIN
- @aiplanet - YouTube
- @aiplanethub- Twitter
- aiplanethub- Instagram

Join the AI Planet Discord Server





AI WITH TARUN
LET'S LEARN AI BY DOING THINGS
SUBSCRIBE
AI With Tarun
@AIwithTarun · 3.31K subscribers · 54 videos
"Learn AI for FREE with Tarun!" ...more
twitter.com/TRJ_0751 and 5 more links
Customise channel
Manage videos
Home
Videos
Shorts
Live
Podcasts
Playlists
Posts
Created playlists
Sort by
VIBE CODING USING MCP
1 video
MCP- Model Context Protocol
Public · Updated 2 days ago
View full playlist
Innate Dojo
1 episode
Innate Dojo
Public
View full podcast
NO APIS LOCAL AGENTIC RAG
5 videos
Agents
Public
View full playlist
Crack GSoC
How to make Proposal
7 videos
Open Source
Public
View full playlist
CREATE A 3D WORLD
11 videos
AI Tools
Public
WTF is REAcT Agent
7 videos
LlamaIndex
Public
RAG USING OPEN SOURCE
10 videos
Langchain - Open Source LLMs
vLLM Faster LLM Inference
4 videos
Large Language Models (LLMS)

bit.ly/ml-summer-analytics-25

Slides: bit.ly/iitg-aiplanet

- Tarun R Jain - LinkedIN
- @TRJ_0751- Twitter
- AI with Tarun - YouTube
- AIwithTarun.ai - Instagram

bit.ly/aiplanet-discord

- AI Planet - LinkedIN
- @aiplanet - YouTube
- @aiplanethub- Twitter
- aiplanethub- Instagram

Join the AI Planet Discord Server