MLOPS MASTERS

Production Grade MLOps Application Development Course

Data Scientist and Mentor with more than 5 years of working experience in the field of Python, Machine Learning, Deep Learning, MLOps, Generative AI, and Robotics systems. Hands-on experience on Classification, Regression, Clustering, Computer vision, Natural language processing, and Transfer learning models to solve challenging business problems.

BOKTIAR AHMED BAPPY

@boktiarahmed73

@dswithbappy

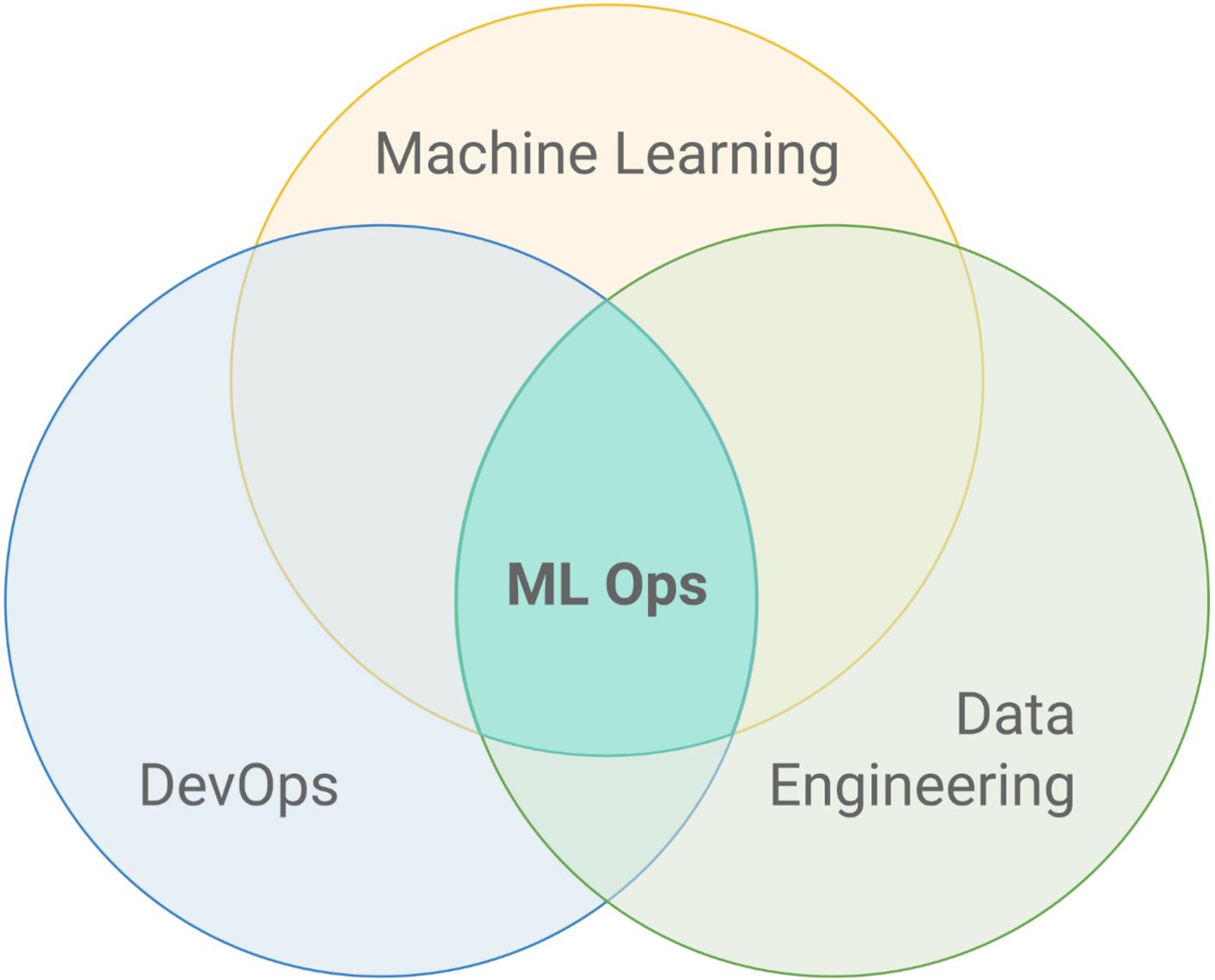
entbappy

UNDERSTAND MLOPS WITH REAL WORLD ANALOGY

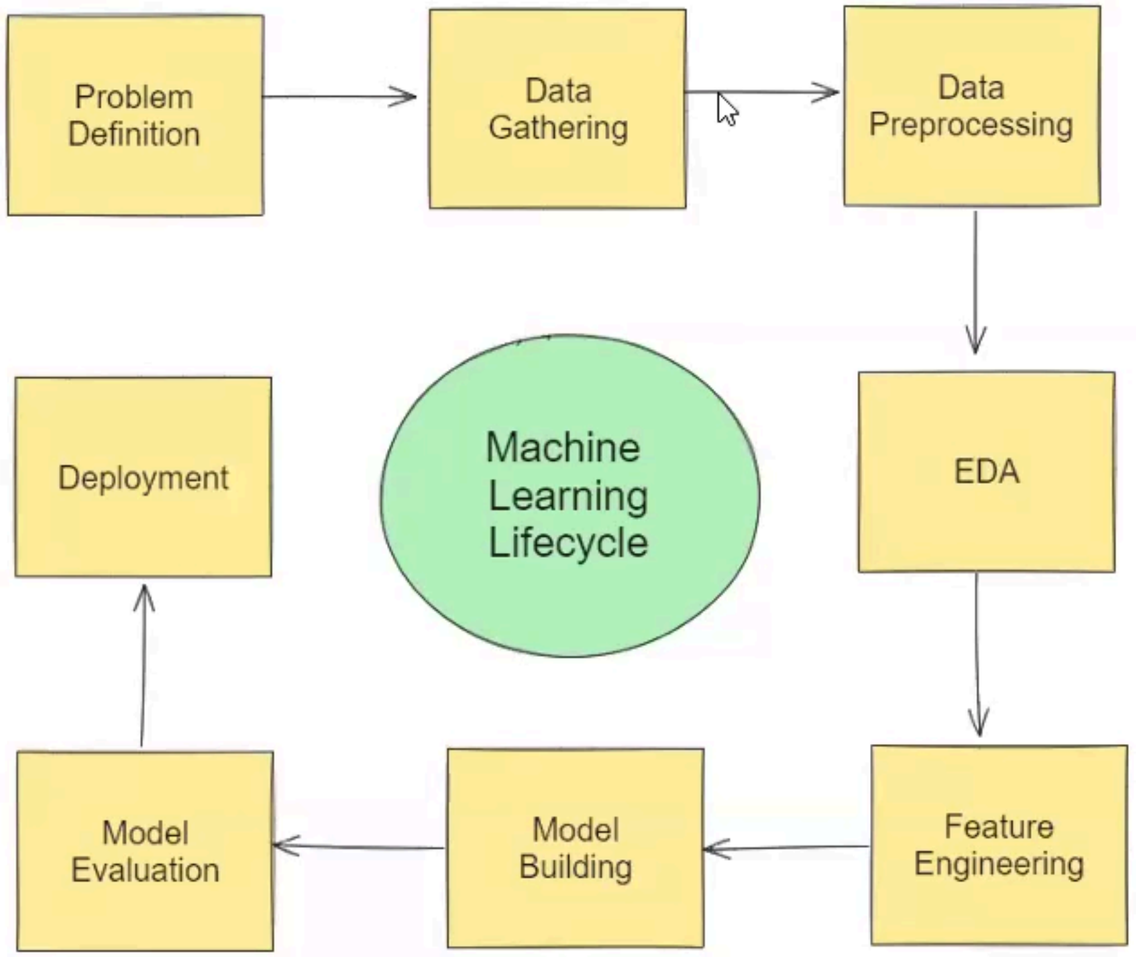
"MLOps is the practice of deploying machine learning models into production"

"MLOps refers to the practice and discipline within machine learning that aims to unify and streamline the machine learning system development (Dev) and machine learning system operation (Ops). It involves collaboration between data scientists, ML engineers, and IT professionals to automate and optimize the end-to-end lifecycle of machine learning applications."

MLOps



INTRODUCTION TO MLOPS & IMPORTANCE


Problem Definition
Data Gathering
Data Preprocessing
EDA
Feature Engineering
Model Building
Model Evaluation
Deployment
Machine Learning Lifecycle

Delhi Capitals •

Rajasthan Royals

(1.5/20 ov) **12/0**

RR chose to field.

Current RR: 6.54

Live Forecast: DC 173

BATTERS	R	B	4s	6s	SR	This Bowler	Last 5 Balls	T20 CAREER Mat	Runs	HS	Ave
Abishek Porel* (lhb)	5	4	1	0	125.00	5 (4b)	5 (4b)	26	501	73*	26.37
Jake Fraser-McGurk (rhb)	7	7	1	0	100.00	1 (1b)	1 (1b)	44	911	84	23.97

BOWLERS	O	M	R	W	Econ	0s	4s	6s	This spell	Mat	Wkts	BBI	Ave
Sandeep Sharma (rm)	0.5	0	6	0	7.20	2	1	0	0.5 - 0 - 6 - 0	183	201	5/18	25.11
Trent Boult (lfm)	1	0	6	0	6.00	4	1	0	1 - 0 - 6 - 0	219	251	4/13	26.01

Partnership: 12 Runs, 11 B (RR: 6.54)

Reviews Remaining: Delhi Capitals - 2 of 2, Rajasthan Royals - 2 of 2

RECORD **56** Sanju Samson has broken the record for playing most IPL matches (56) as captain for RR, going past Shane Warne

• | 4 | • | 1 | 1 | 1st | • | 4 | • | • | • | 2 | See all ›

Current Over 2 • DC 12/0

Live Forecast: DC 173

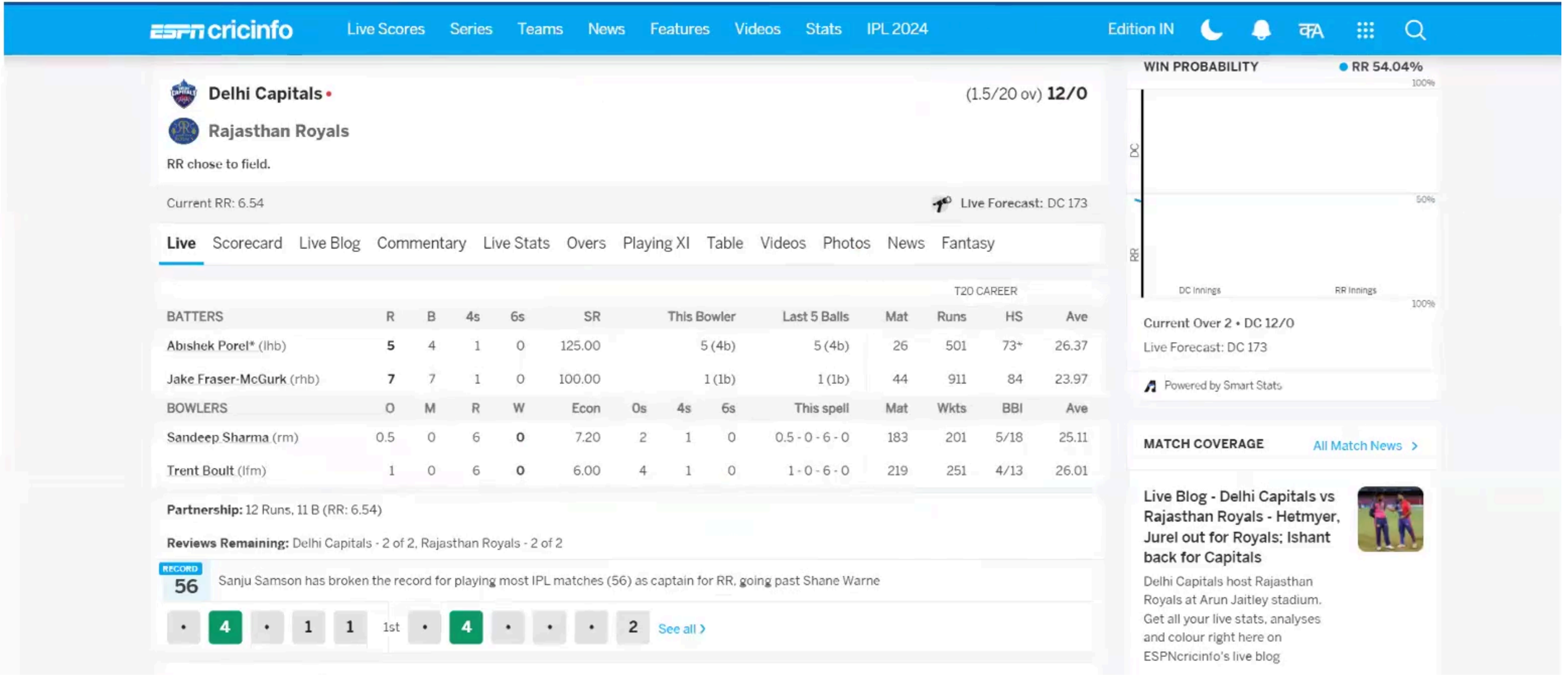
Powered by Smart Stats

MATCH COVERAGE All Match News ›

Live Blog - Delhi Capitals vs Rajasthan Royals - Hetmyer, Jurel out for Royals; Ishant back for Capitals

Delhi Capitals host Rajasthan Royals at Arun Jaitley stadium. Get all your live stats, analyses and colour right here on ESPNcricinfo's live blog

Data Problem



Code Implementation Problem

```python
# Import necessary libraries
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

# Load the Iris dataset
data = load_iris()
X = data.data
y = data.target

# Preprocess the data: Standardize features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.3, rando

# Train a logistic regression model
model = LogisticRegression()
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")
```

1. Difficulty in Maintenance: As the complexity of the code increases, it becomes harder to maintain. Changes in one part of the code can unexpectedly affect other parts due to the tightly coupled nature of the implementation. This can lead to increased time spent debugging and verifying that changes do not break existing functionality.

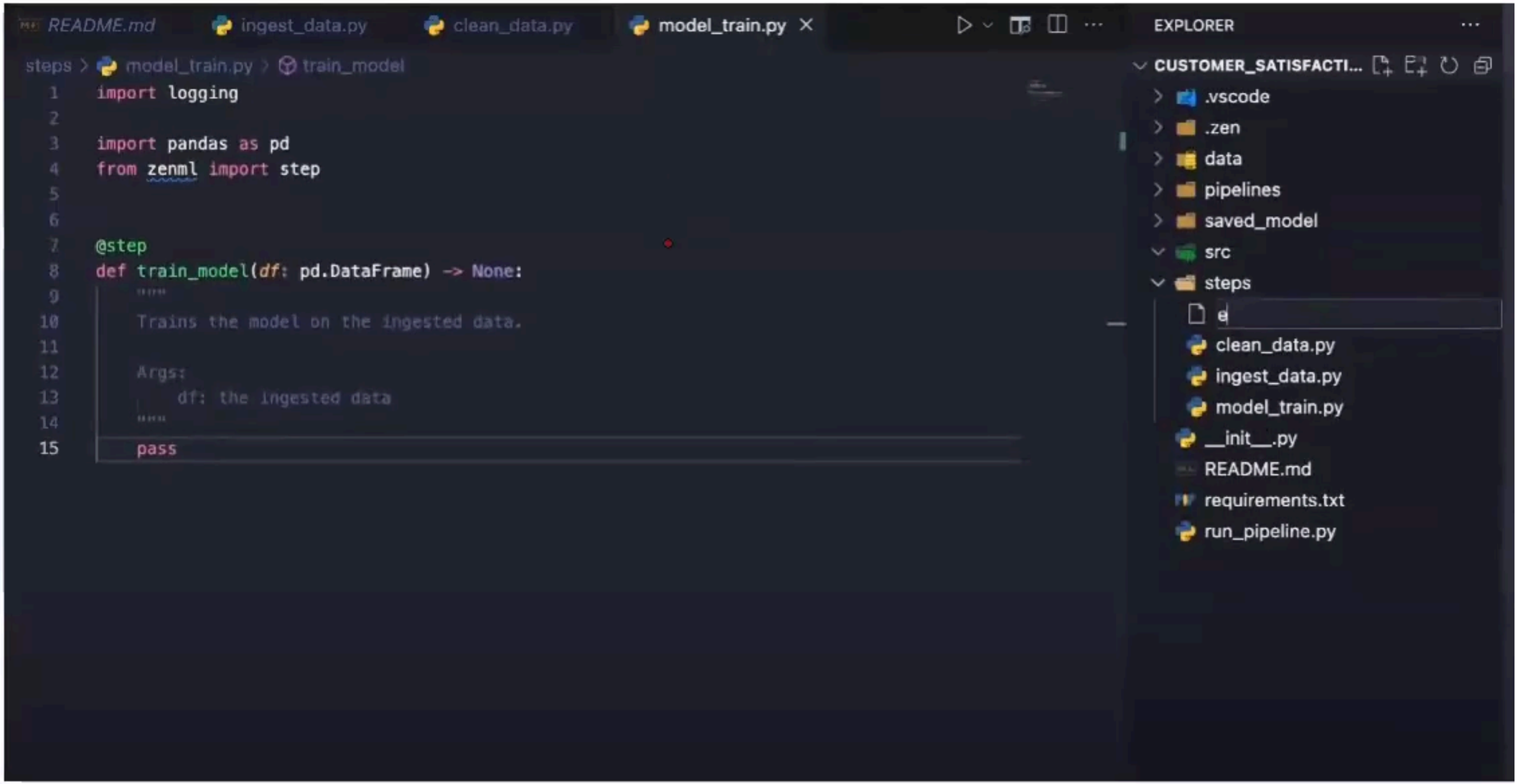
2. Limited Reusability: In a non-modular setup, code reusability is minimal. Functions and components are often written to solve a specific problem and are tightly integrated with other parts of the code, making them difficult to extract and reuse in other projects or contexts.

3. Harder Collaboration: When code is not modular, it's more challenging for multiple developers to work on the same project simultaneously. Since everything is interconnected, developers need to be more cautious about the changes they make, which can slow down development and increase the risk of merge conflicts.

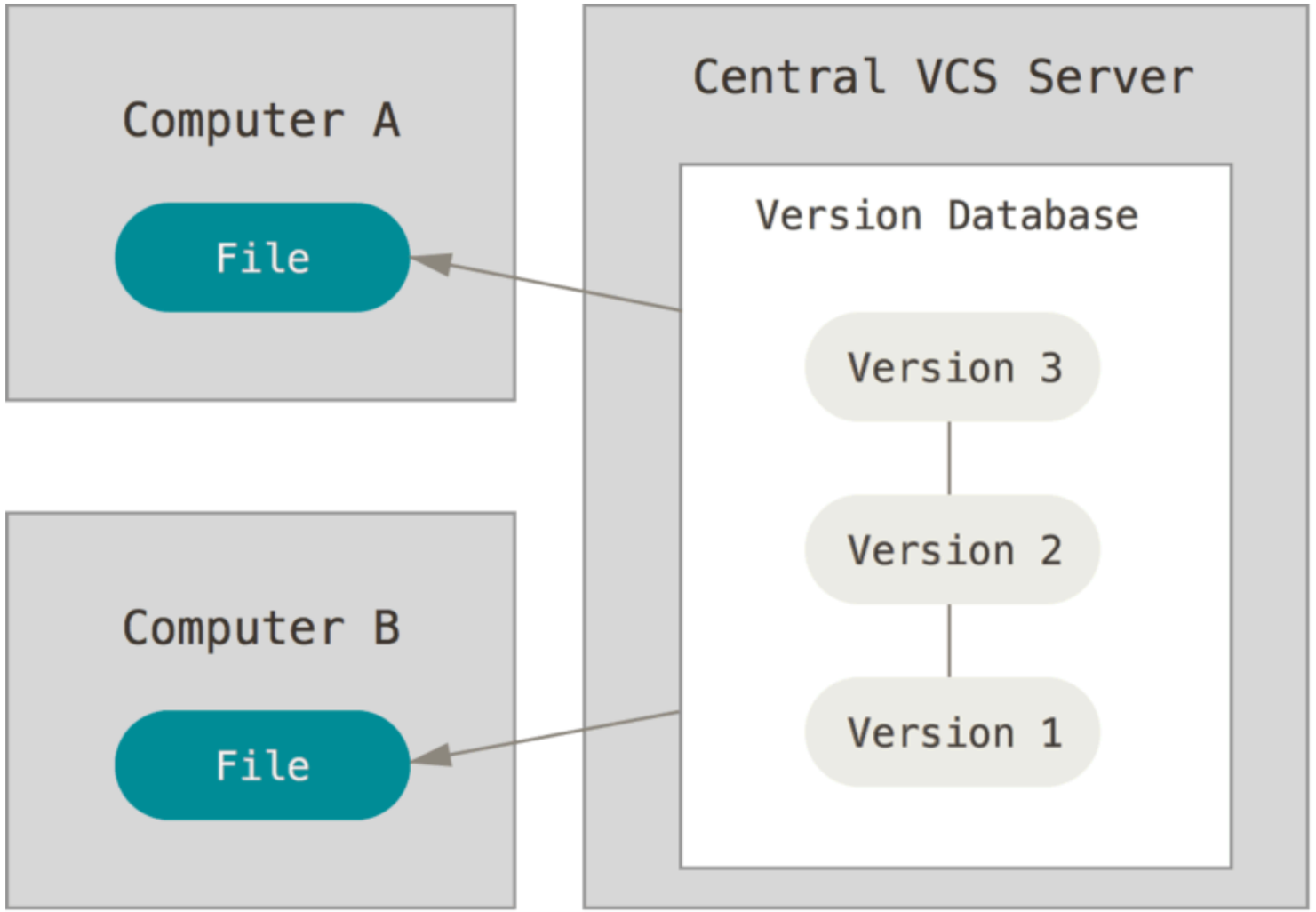
4. Poor Scalability: Scaling a non-modular application can be problematic. As more features and functionalities are added, the codebase can become unwieldy and difficult to manage. This lack of scalability extends not only to the size of the codebase but also to the handling of larger data sets or more complex models.

5. Testing Challenges: Testing a non-modular codebase is typically more challenging and less effective. Without clear boundaries between

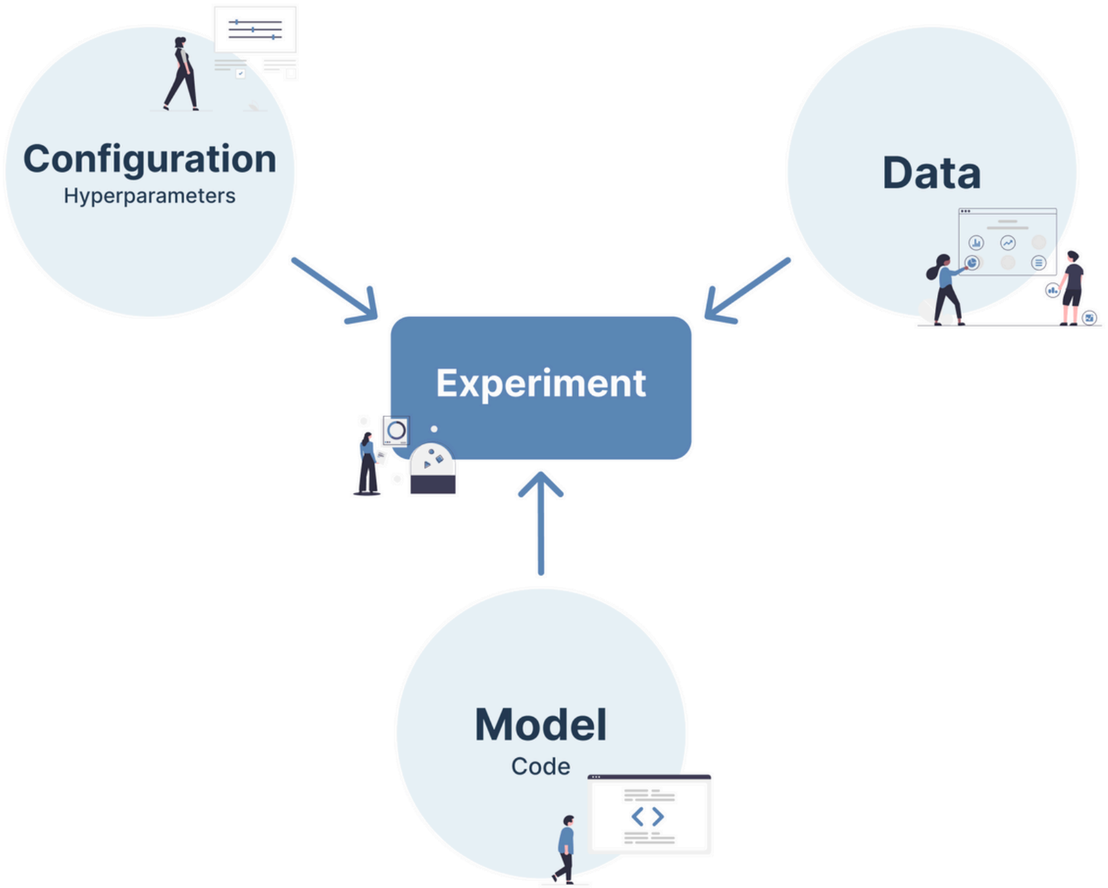
Code Implementation Problem



Version Problem



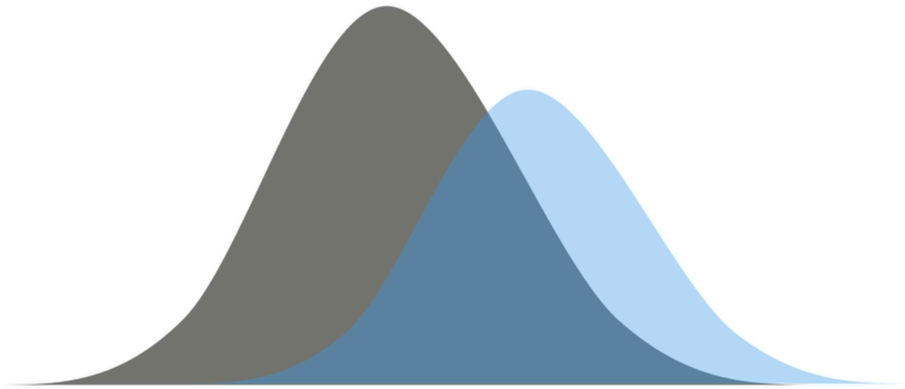
Experiment Problem



Deployment Problem

Data Drift Problem

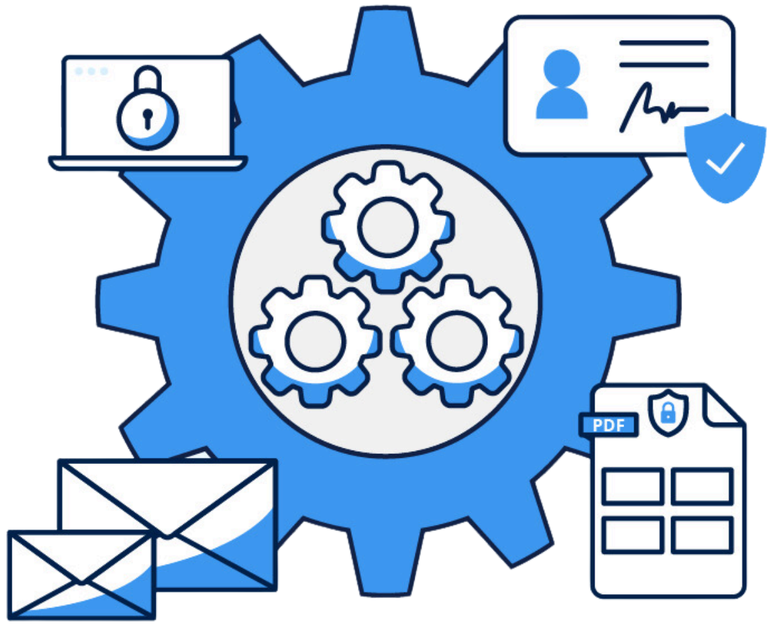
Input data **DRIFTED**



Predictions **DRIFTED**

Infrastructure Problem

Automation Problem



Legal Problem

Benefits of MLOps

1. Scalability
2. Improved performance
3. Reproducibility
4. Collaboration and efficiency
5. Risk reduction
6. Cost Savings
7. Faster time to market
8. Better compliance and governance

Challenges

1. Complexity of ml models [variability, black box nature]
2. Quality of data
3. Cost and resource constraints
4. Handling scale
5. Security risks
6. Compliance and regulatory concerns
7. Integration with existing systems
8. Limited Expertise/Skill gap

LINUX FUNDAMENTALS FOR MLOPS