PG Certificate Course in Data Science, AI/ML and Data Engineering by IIT Roorkee

Final Project Submission – Chandramauli Joshi

Agenda

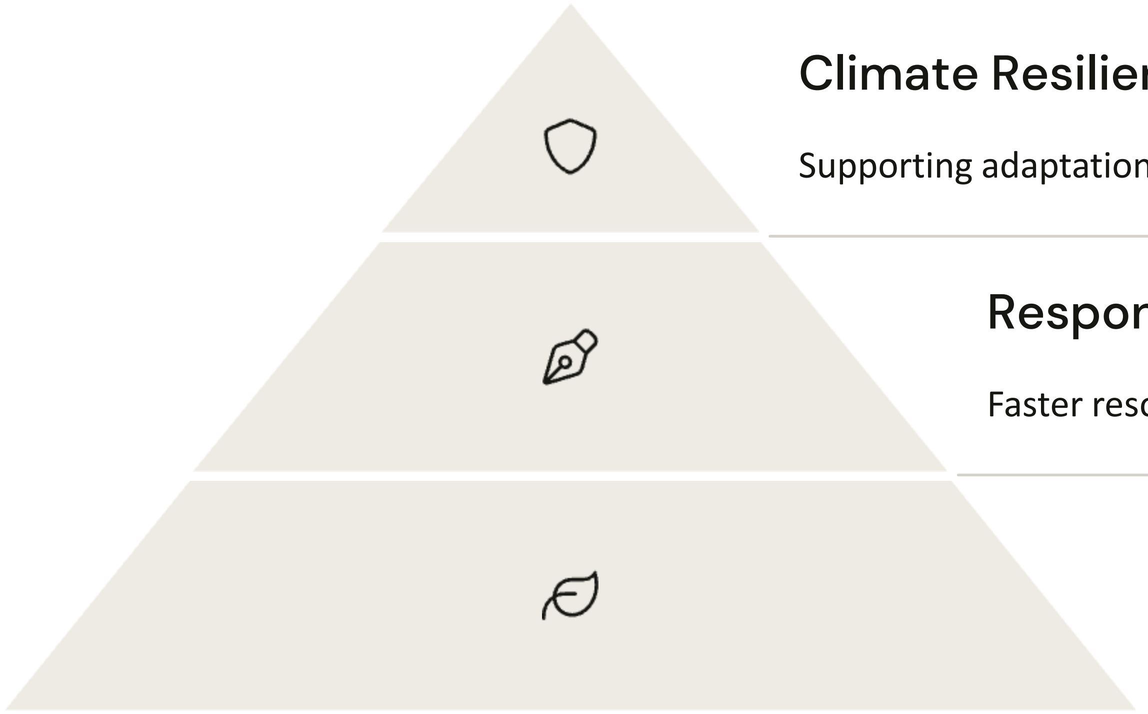
Agenda (Conti.)

Agenda (Conti.)

Wildfire Prediction Project

- Predicting burned forest area using historical weather and environmental data
- Enables preventive action and response planning
- Bridges gap between meteorological data and actionable insights insights for early warning systems.

Impact/Goal



Climate Resilience

Supporting adaptation strategies

Response Optimization

Faster resource allocation

Environmental Protection

Aligns with sustainability initiatives

Proof of Concept –

To Validate Predictive Feasibility

1. ## Minimal Model

   Built using data subset

2. ## Baseline Accuracy

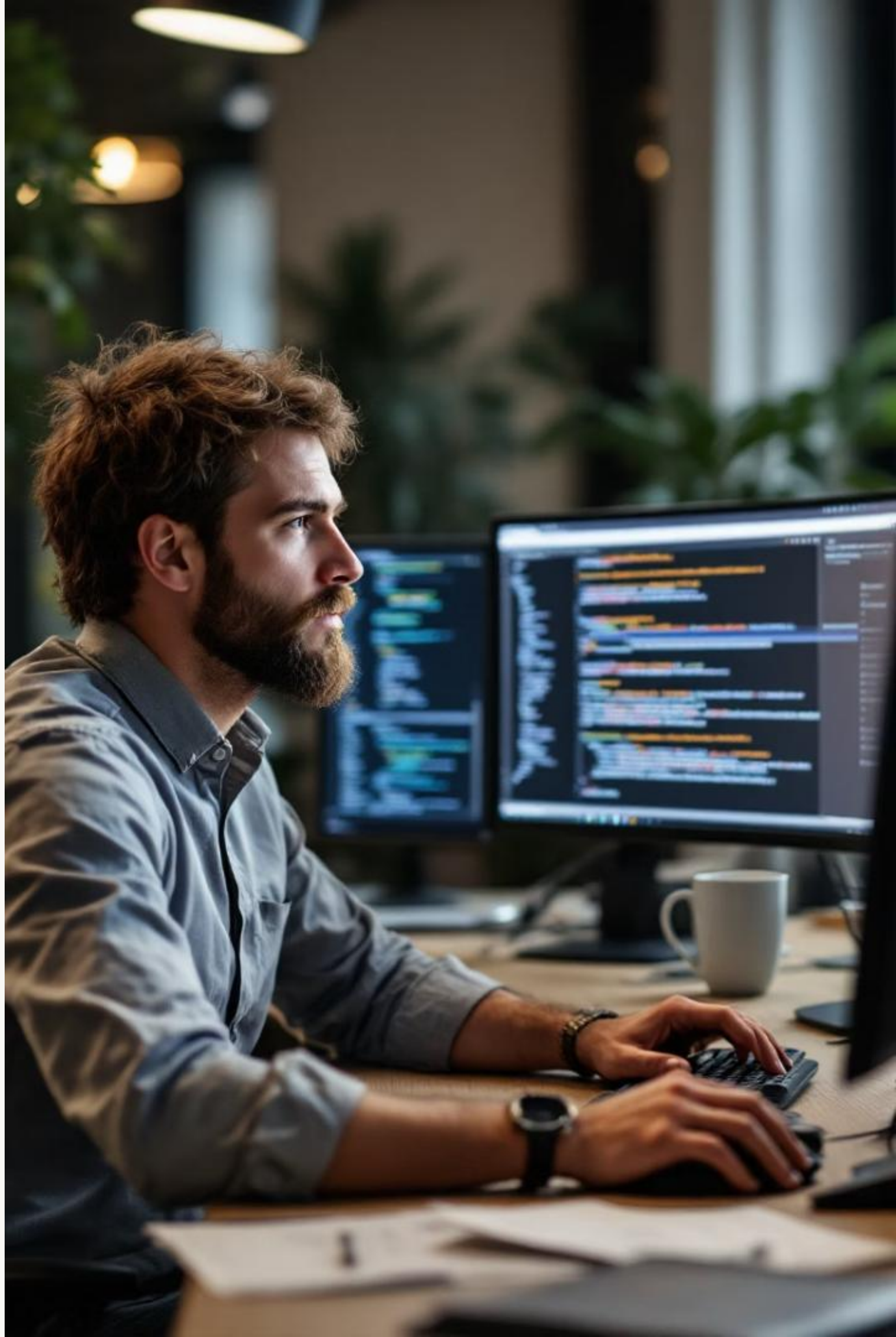
   Established performance benchmarks

3. ## Feasibility Validation

   Confirmed regression approach viability

4. ## Production Justification

   Provided rationale for full-scale implementation



Model Exploration

Linear Regression

Underperformed due to multicollinearity

Decision Trees

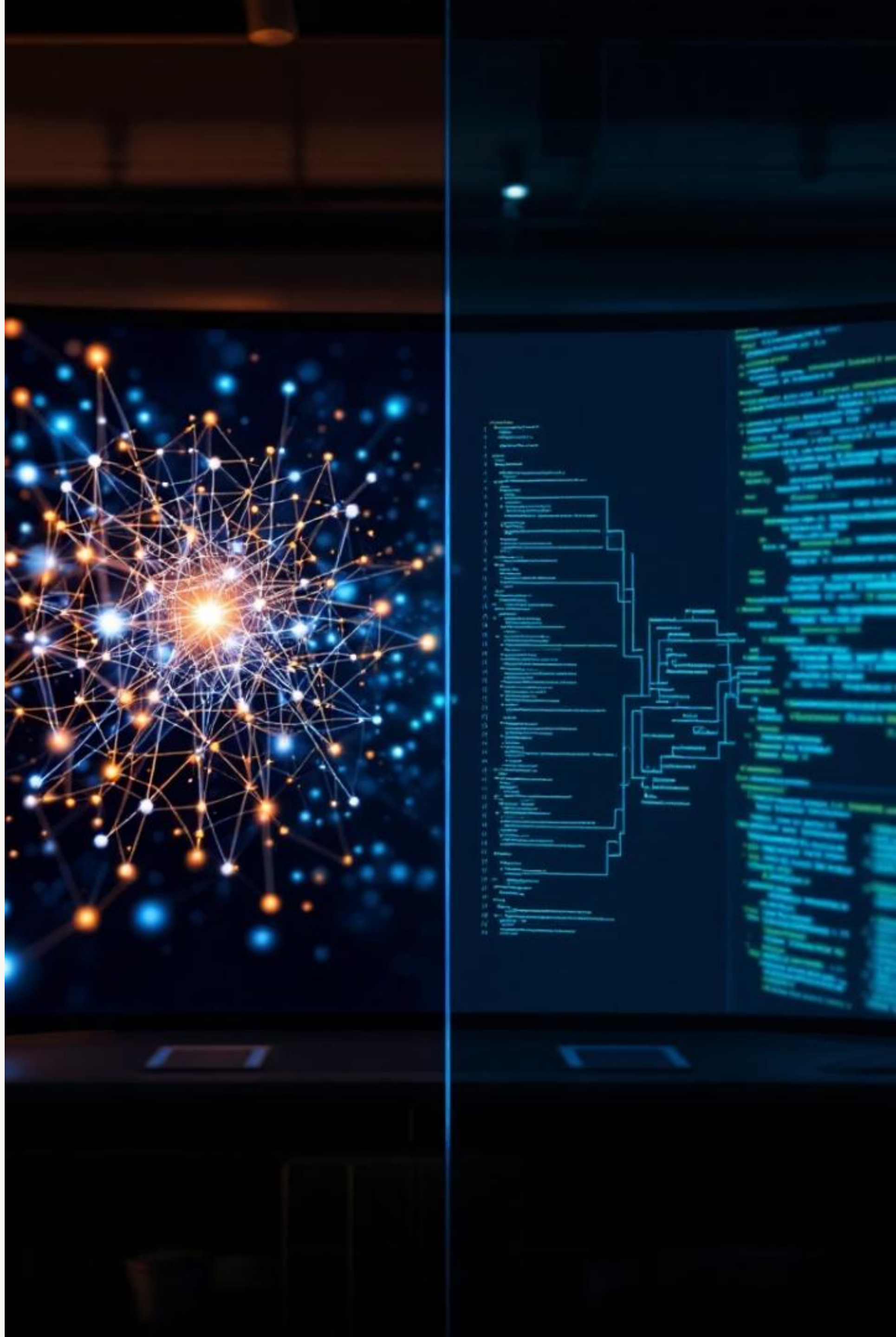
Overfitting risks, lacked interpretability

Lasso Regression

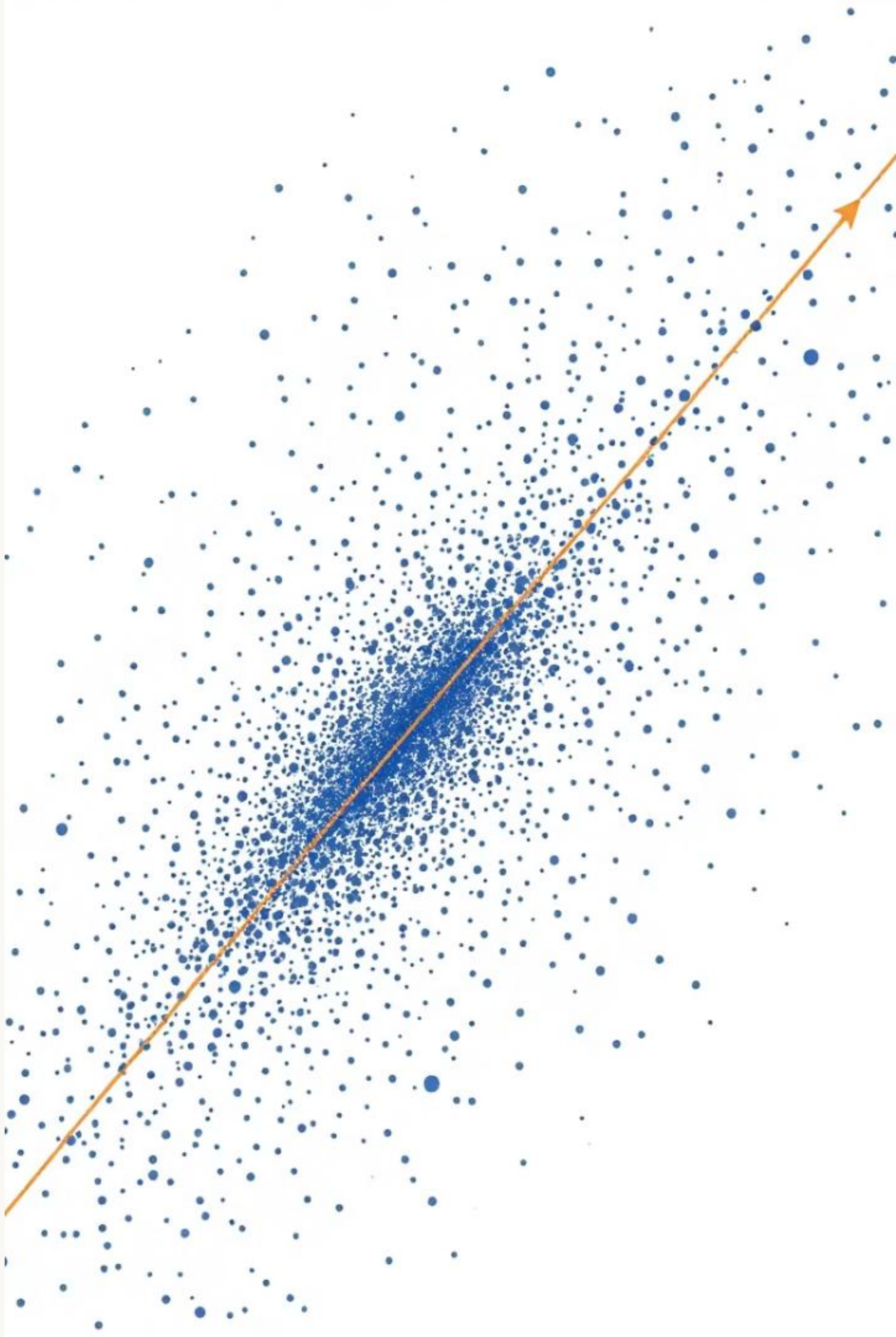
Good for feature selection, less stable

Ridge Regression

Selected for regularization and robustness



Regression Approach



Regression Problem

Predicting continuous numeric value

Target Variable

Burned area in hectares

Method Selection

No deep learning needed due to dataset size

Performance Metrics

R^2 Score

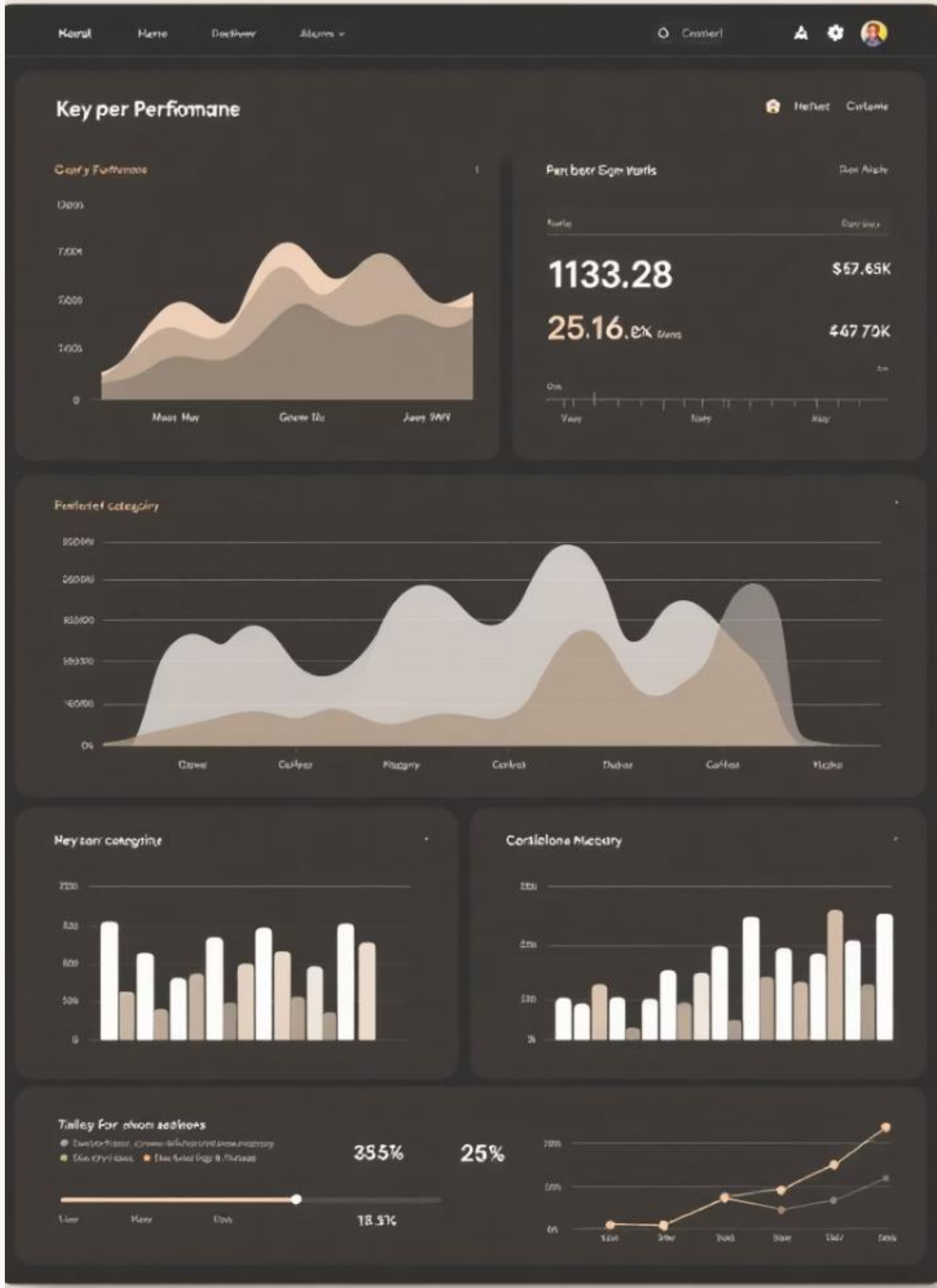
Measures prediction-actual match

Mean Absolute Error

Average prediction deviation

Visualization

Predicted vs. actual plots plots



Performance Impacts: Simplicity vs. Power

Ridge

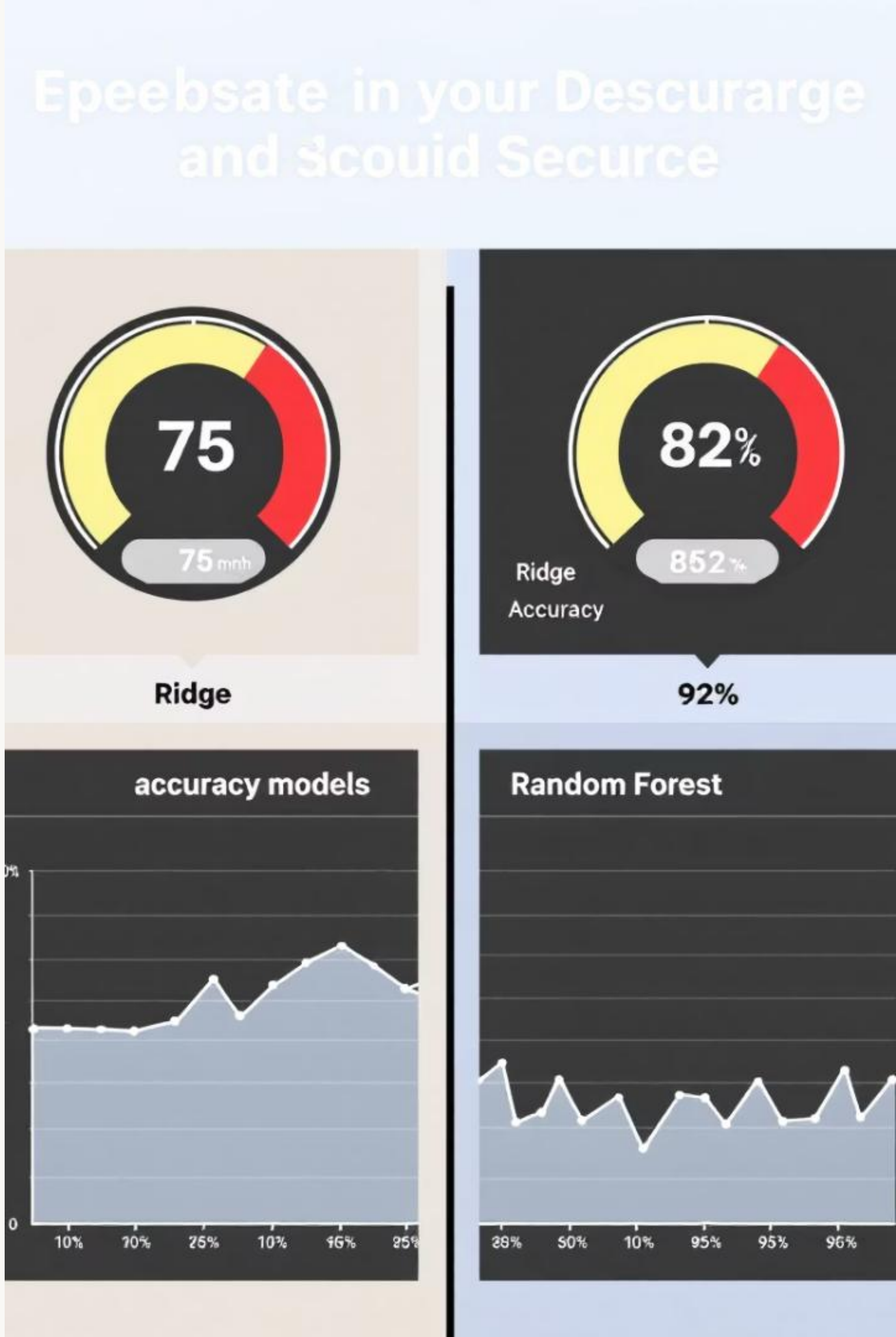
- Fast training
- Consistent generalization

Random Forest

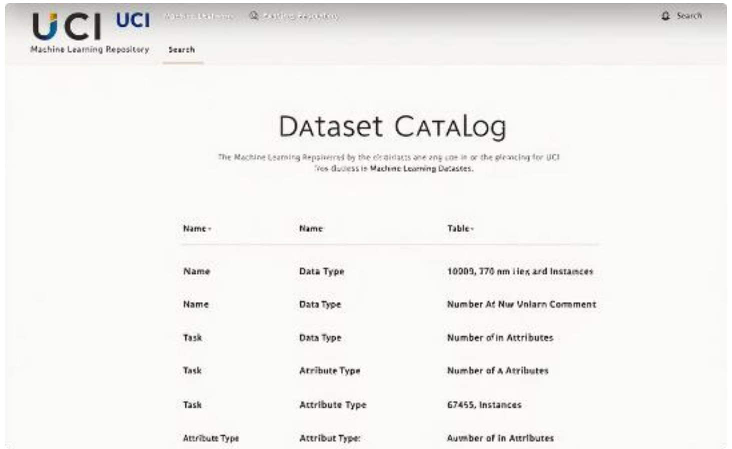
- Potentially higher accuracy
- More overfitting on small data data

Neural Network

- High complexity
- Data hungry



Data Sources



UCI Repository

Trusted academic data source

Geographic Coverage

Bejaia and Sidi Bel-abbes regions

Data Components

Weather conditions and fire indices

Data Preparation – **Ensuring Quality Inputs**

Cleaning

Removing duplicates, filling missing values

values

Scaling

Normalizing input with StandardScaler

StandardScaler

Date Conversion

Creating machine-readable formats

Regional Splits

Ensuring data homogeneity

Feature Engineering – Creating Meaningful Inputs

Fire Weather Index

DC, DMC, FFMC components emphasized

Correlation Analysis

Selected most impactful features

Regional Handling

Prevented data leakage between regions

Feature Reduction

Dropped redundant attributes

Technology Stack – Tools that Power the Solution

Languages & Frameworks

- Python
- Scikit-learn
- Flask

Libraries & Deployment

- Pandas, NumPy, Matplotlib, Seaborn
- AWS Elastic Beanstalk
- HTML Templates

System Architecture

Data Ingestion

CSV loading with Pandas

Data Cleaning

Handling nulls, formatting dates

Feature Engineering

Transforming and scaling features

Model Training

Ridge regression with Scikit-learn

Evaluation & Deployment

Metrics and Flask interface

File Sequence

1. README.md
2. dataset/Algerian_forest_fires_cleaned_dataset.csv
3. notebooks/3.0-Model Training.ipynb
4. models/scaler.pkl
5. models/ridge.pkl
6. application.py
7. templates/home.html
8. templates/index.html
9. .ebextensions/python.config
10. requirements.txt
11. .vscode/settings.json, extensions.json, tasks.json

Key Concepts & Functionalities – Core Technical Elements

- Core regression concepts with L2 regularization (Ridge).
- Feature scaling and correlation analysis.
- Flask routes and templates to build a responsive web interface.
- Backend inference pipeline using pre-trained .pkl models.

Model Selection & Training – Making the Right Choice

Multicollinearity

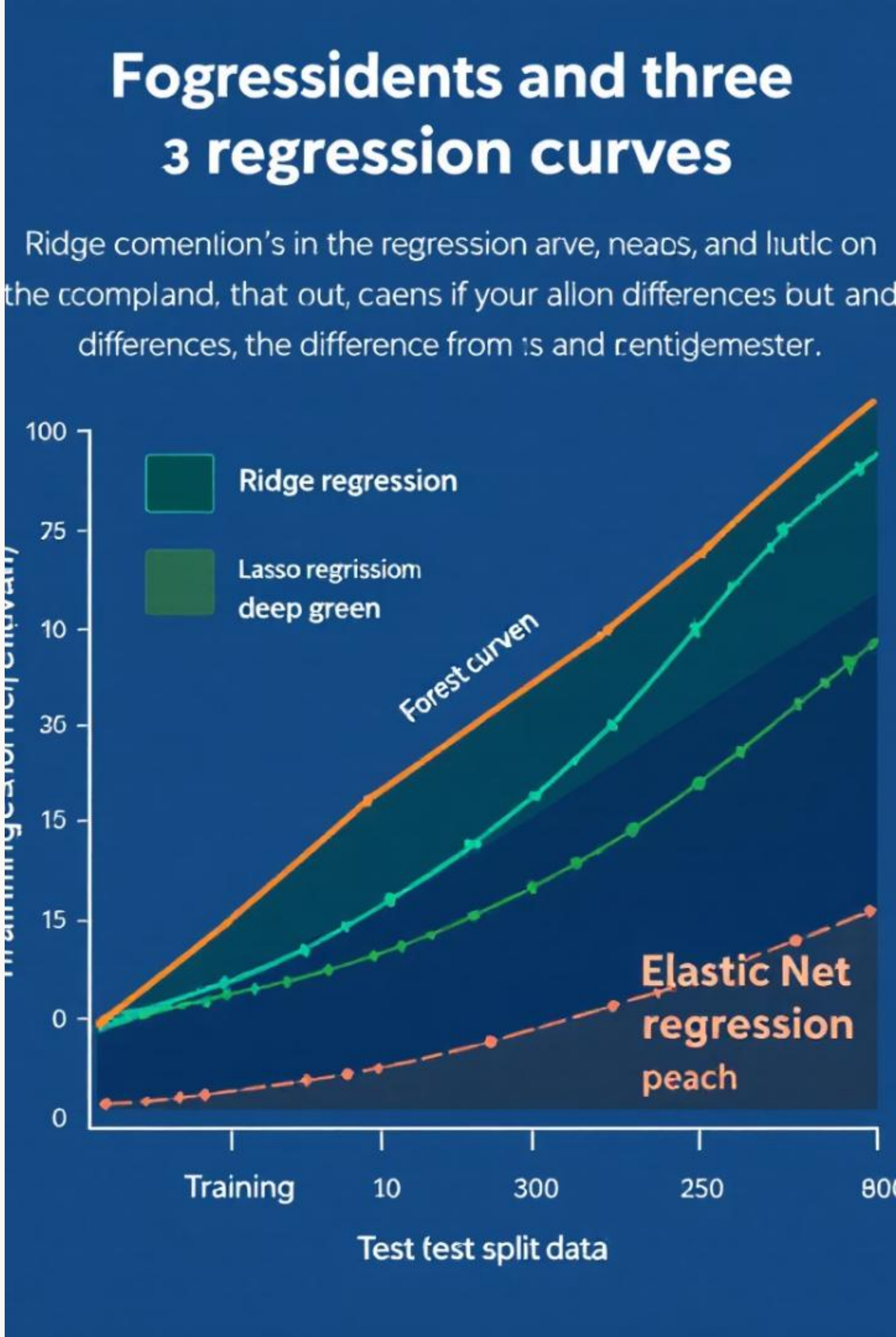
Ridge chosen for feature correlation

Training Split

80% train, 20% test

Pickle Serialization

Models ready for deployment



Evaluation & Validation – Testing for Generalization

1. **R^2 Score**
   Explains variance

2. **MAE Metric**
   Measures average error

3. **Residual Plots**
   Visualize prediction errors

4. **Extreme Inputs Check**
   Test robustness

Testing Approach – Validating End-to-End Performance

1. **Model Testing**

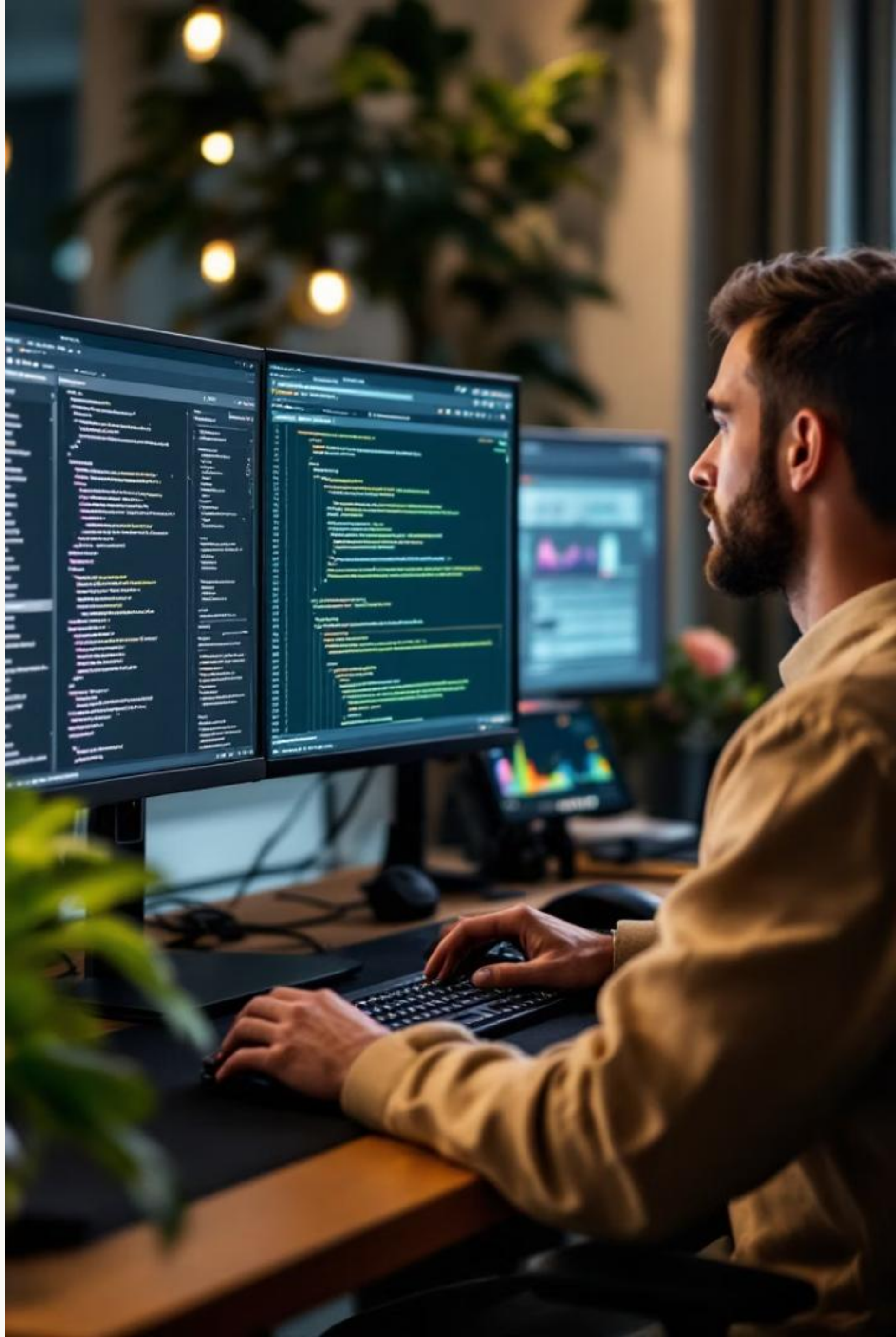
   Prediction accuracy check

2. **Web UI Testing**

   Form and output validation

3. **Scenario Testing**

   Edge weather case analysis



Integration & Deployment – Making It Accessible

Flask App

application.py, modular routes

Frontend

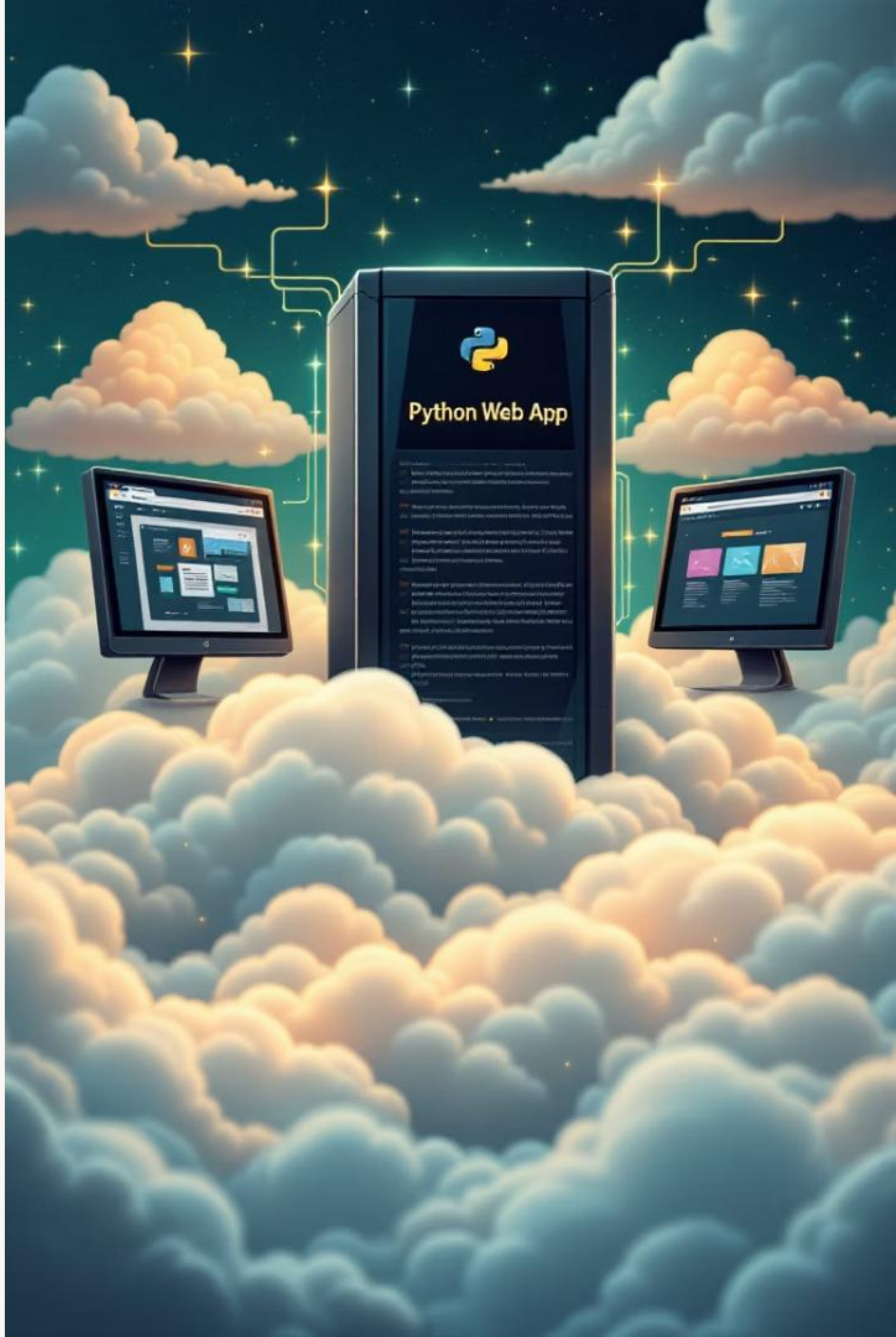
home.html, index.html templates

AWS Deployment

Public access, scalable cloud

Real-time Prediction

Live user input & inference



Requirements – Functional and Non-Functional

Functional

- Accept user input for weather and FWI data.
- Display predicted burned area.

Non-Functional

- Fast response time.
- Simple and clean UI.
- Secure and stable deployment environment.

Modeling Trade-Offs in Machine Learning Solutions

Exploring key trade-offs: interpretability vs. complexity, backend flexibility, performance vs. dataset constraints. Examining real-world model selection decisions and their impacts.

Interpretable vs. Complex Models

Ridge Regression

- High interpretability
- Lower variance
- Efficient on small data

Random Forest

- Less transparency
- Higher complexity
- Harder to tune

Dataset Size: Constraints & Opportunities

Lightweight Models

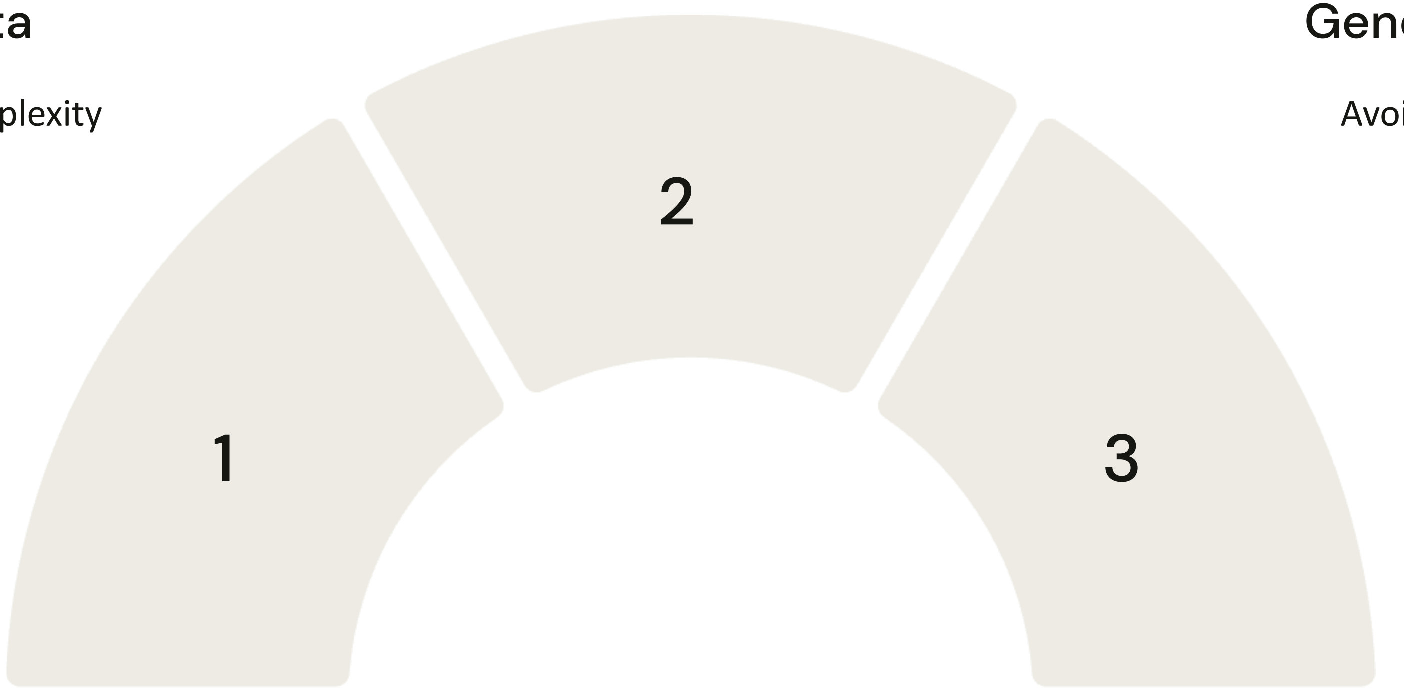
Achieve stability

Small Data

Limits model complexity

Generalization

Avoid overfitting



Decision Timeline: Key Milestones

1 Data Exploration

Assessed feature space

2 Model Screening

Tested Ridge & Forest

3 Framework Selection

Chose Flask backend

4 Final Evaluation

Validated model output

Learnings – Lessons from the Journey

Feature Scaling Impact

Regression performance boost

Model Selection

Ridge outperformed alternatives

Deployment Flexibility

Rapid iterations with Flask

Preprocessing/Segmentation

Higher accuracy via domain logic

Key Takeaways

1 **Prioritize Interpretability**

For stakeholder impact

2 **Optimize for Data Size**

Model must match scale

3 **Backend Flexibility**

Facilitates iteration

4 **Continuous Evaluation**

Refine as project grows

Future Scope – Looking Ahead

1. **Real-time Weather API**

   Automate data streams

2. **Satellite Features**

   Integrate remote sensing

3. **Enhanced Visual UI**

   Heatmaps, risk zones, charts

4. **Expand Regions**

   Global wildfire prediction

Solving Regional Model Performance Challenges

Building a predictive model across two distinct regions posed sharp drops in accuracy. Geographical variation led to feature influence inconsistencies. Addressing this challenge required careful technical analysis and targeted strategy. Our approach boosted reliability and set the app up for broader, more robust deployments.

STAR Solution: Regional Model Stabilization

S

Observed model performance loss when merging regional datasets; detected varying feature influence driven by geography.

T

Needed to maintain high accuracy and stability across both regions without bias or generalization loss.

A

- Explored data to reveal distinct patterns per region.
- Trained two region-specific models with robust individual evaluation.
- Deployed preprocessing that auto-detects and routes inputs.
- Applied scaling and Ridge Regression regularization.

R

- R^2 increased from 0.67 to 0.86.
- Elevated accuracy and seamless region-aware app deployment.

Thank You & Questions

Appreciate your attention

Thank you for joining today

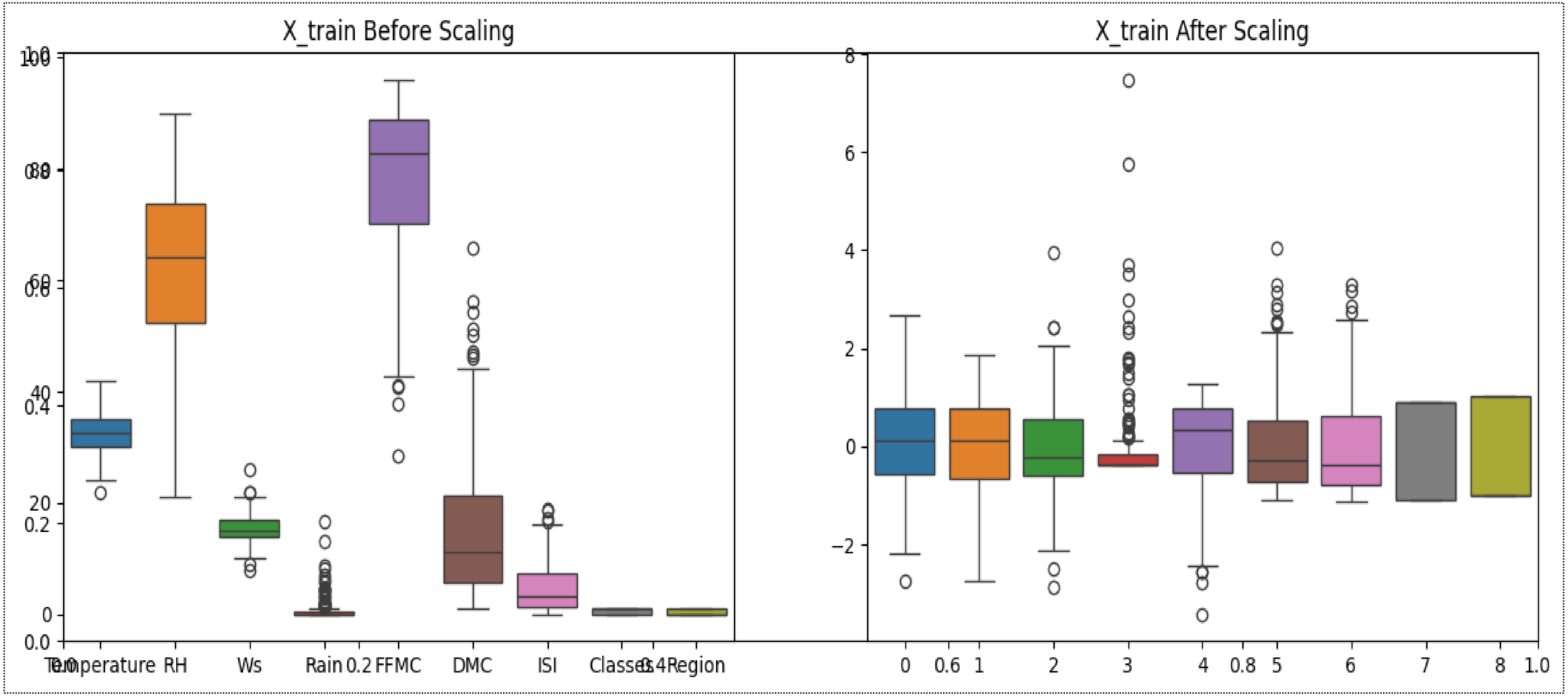
Open Q&A

Any questions or feedback?

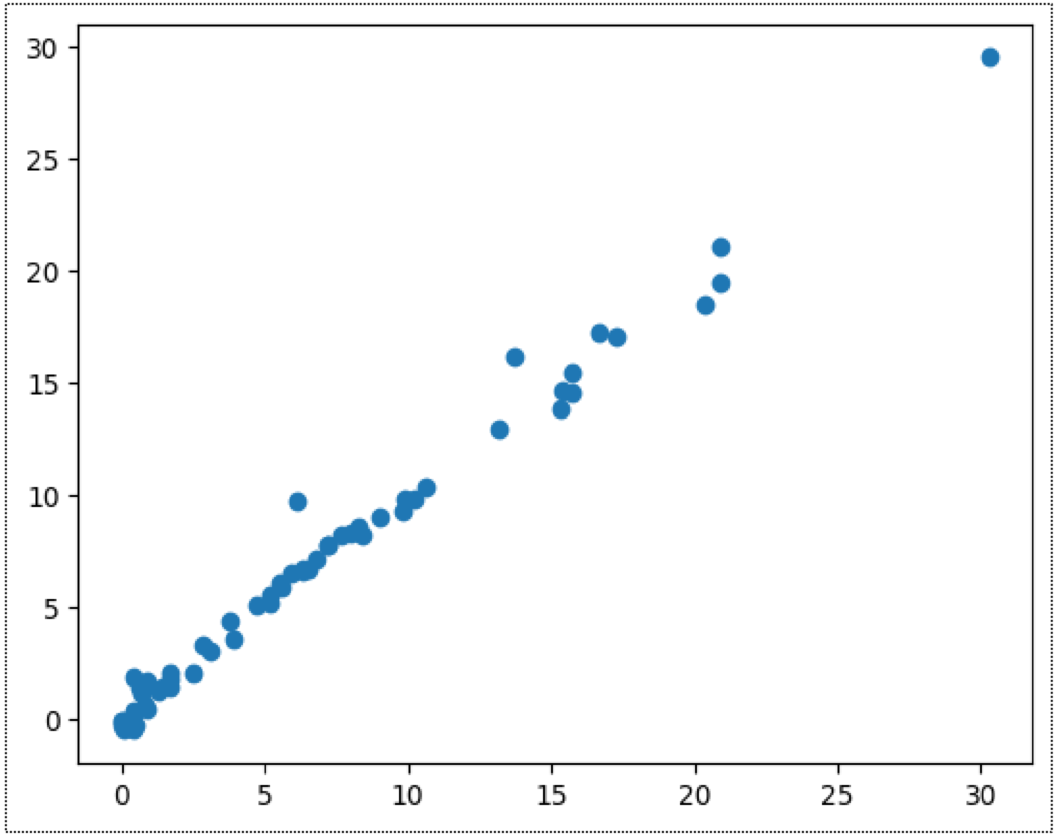
Appendix

Heatmap for Multicollinearity

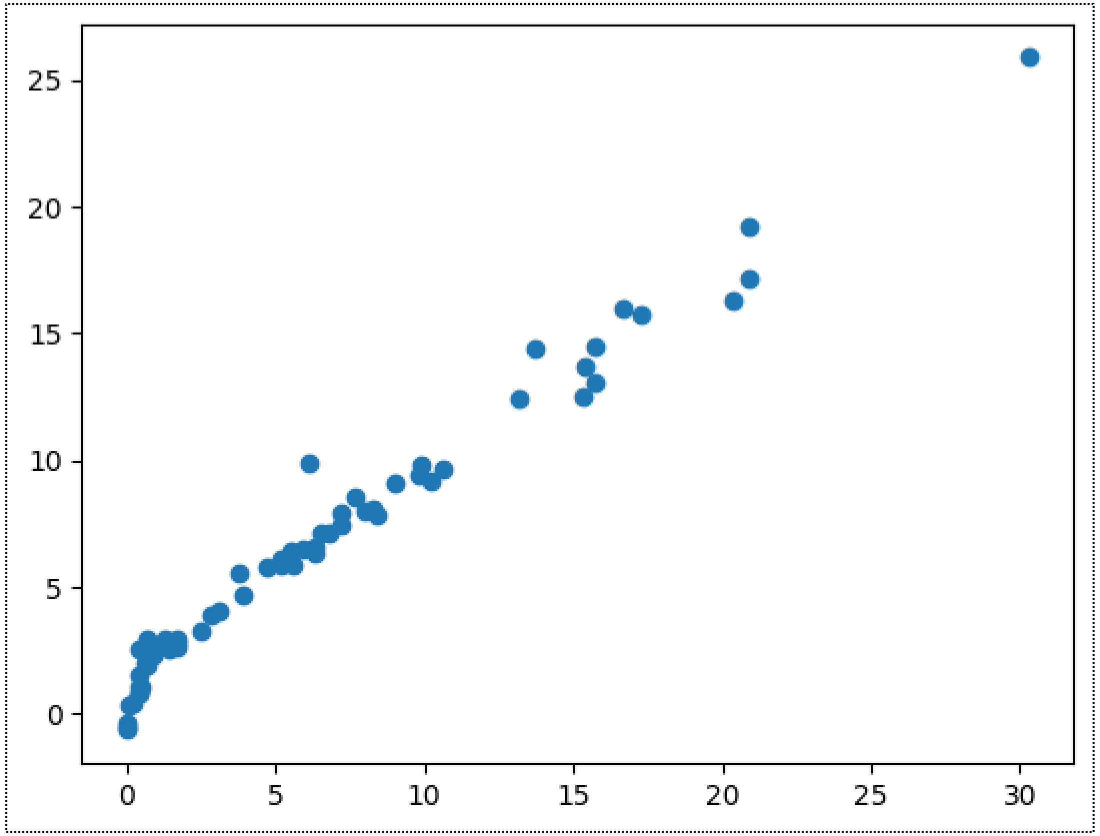
Box Plots to understand the effect of Standard Scalar



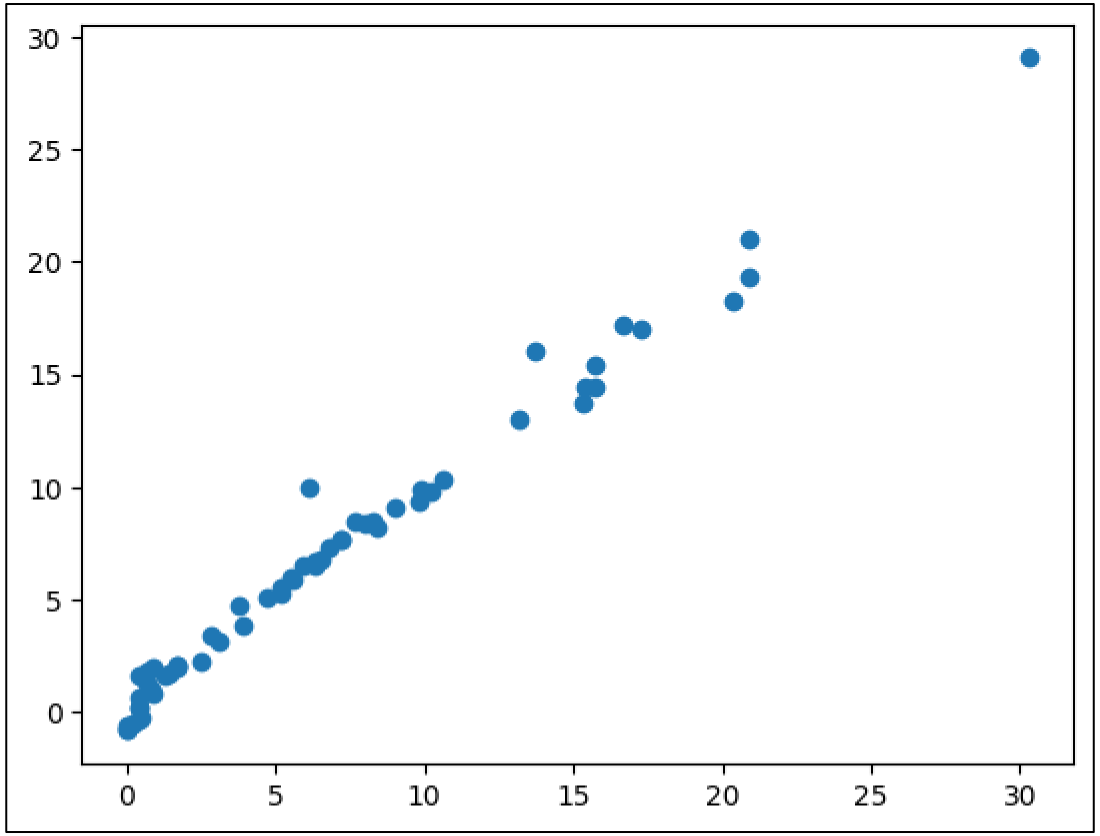
Linear Regression



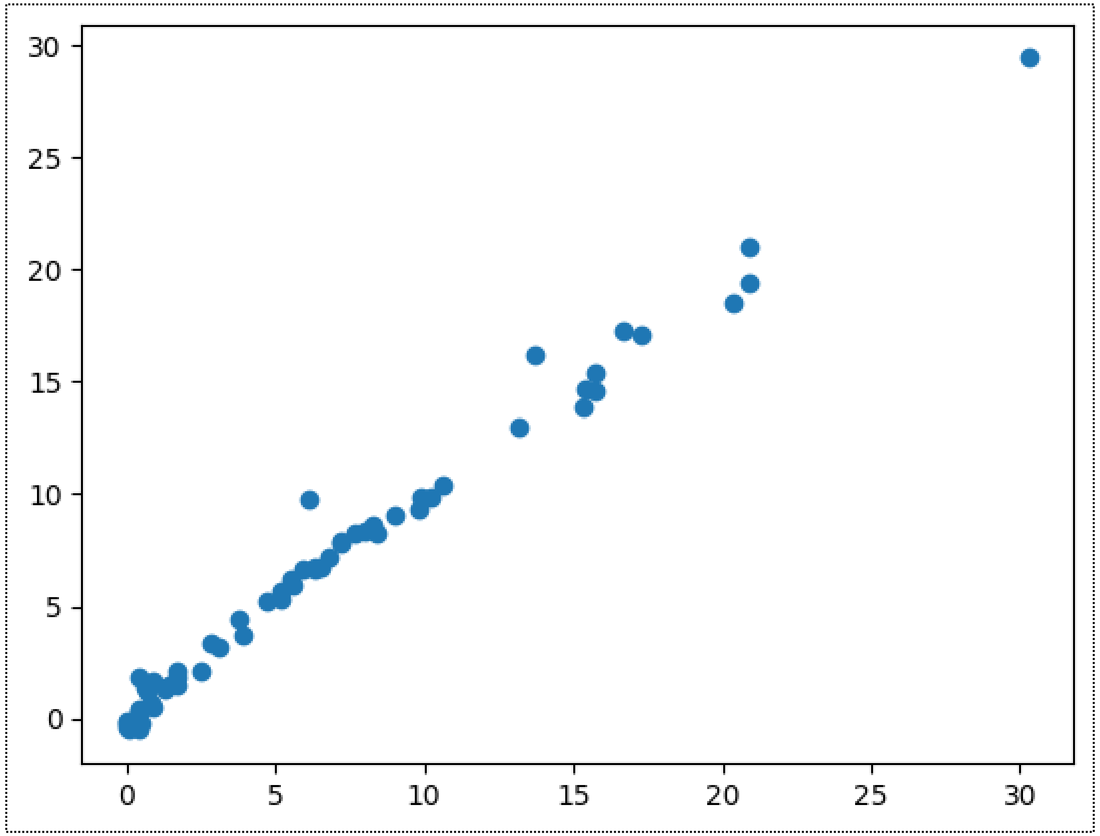
Lasso Regression

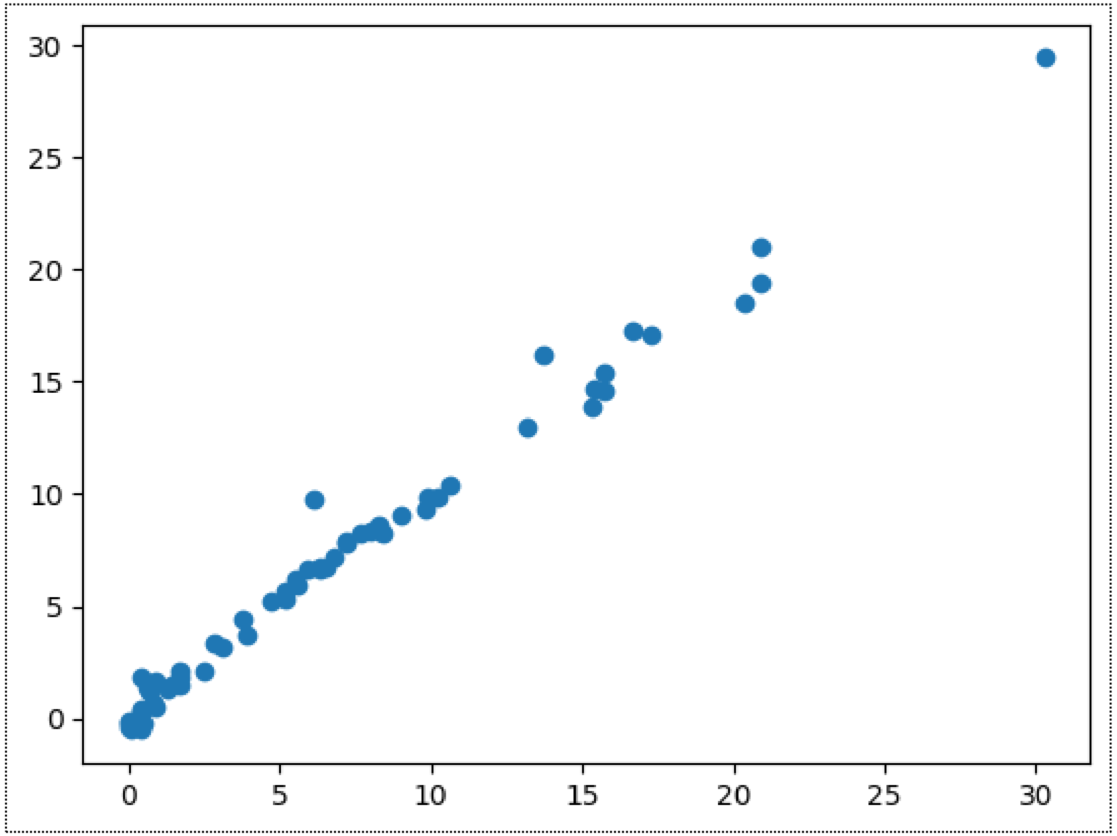


Cross Validation Lasso



Ridge Regression Model





Elasticnet Regression