Data 1030 Final Presentation

Repository Link

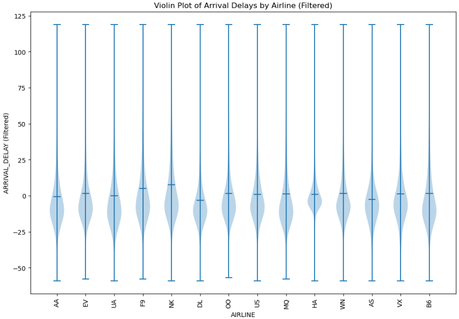
Chai Harsha

Brown University

December 9, 2025

Recap

- Predict flight delays using historical data → better allocate airport resources
- USDOT Bureau of Transportation Statistics collects on-time performance of major carriers
- Kaggle dataset - 2015 flight delays
- Preprocessing: Dropped rows with missing target values, removed obviously leaky columns and columns that have too many missing values
- Train-Val-Test split: 60-20-20
- EDA: Some variance in delays across airlines, airports, time of day



Models

- Large dataset resulted in restricted model choice
- Linear Regression (ElasticNet) - SGDRegressor because faster training on large data
- Support Vector Machine (Linear SVM) - LinearSVR for faster training + RAM efficiency
- Decision Tree Regressor - chose random splits for faster training
- XGBoost Regressor - has efficient enough implementation

Cross Validation

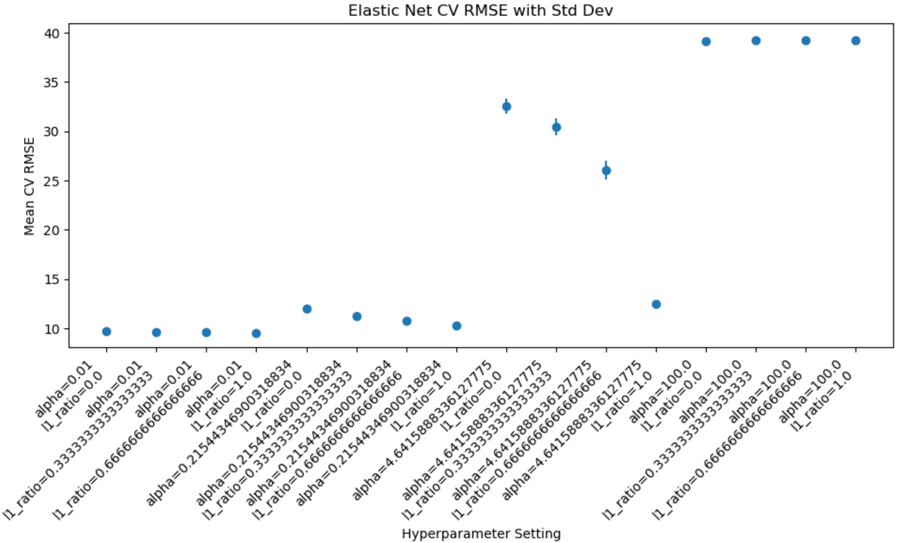
- Recall Train-Val-Test split was 60-20-20
- Used 4-Fold Cross Validation (with rs=42) on Train set for hyperparam tuning
- Selected hyperparams with lowest average RMSE across folds
- Tested best model on Val set to estimate performance

ElasticNet	Linear SVM	Decision Tree	XGBoost
$\alpha \in [0.01, 100]$, l_1 ratio $\in [0, 1]$, 4 values each	$C = 10^n, n \in \{-1, 0, 1, 2\}$, $\varepsilon = 0$	max_depth $\in \{3, 7, 10\}$	max_depth$\in \{6, 9, 12\}$ lr$\in \{0.01, 0.1, 1\}$ n_trees$\in \{300, 400, 500\}$

Results

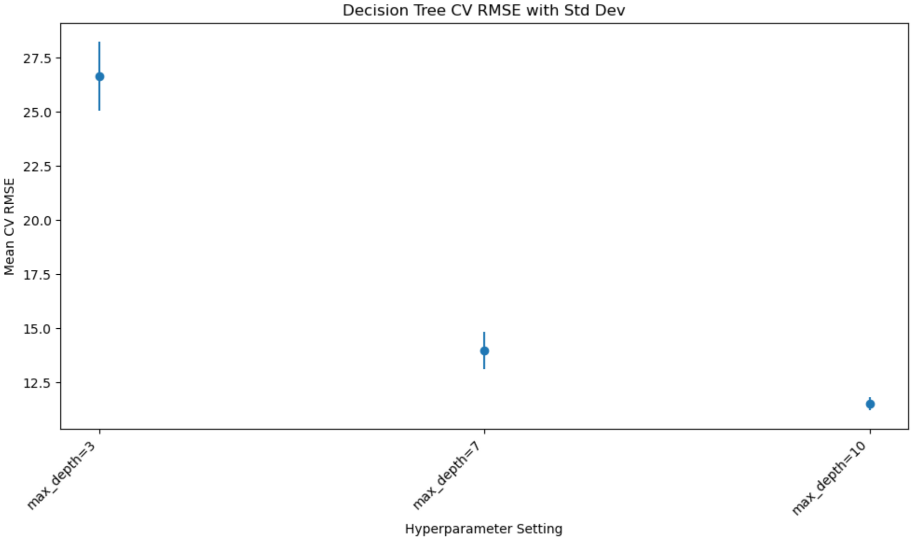
- Baseline prediction (mean of test set): $RMSE = 39.2402, R^2 = -0.0000$
- ElasticNet: $RMSE = 0.1498, R^2 = 1.0000$
- Linear SVM:$RMSE = 0.0000, R^2 = 1.0000$
- Decision Tree $RMSE = 11.7675, R^2 = 0.9101$
- XGBoost: $RMSE = 2.9834, R^2 = 0.9942$
- This is suspicious - there is some data leakage, some columns I should have left out
- If you have scheduled arrival and departure and actual arrival and departure, delay is trivial for linear models

Mean + Std Dev

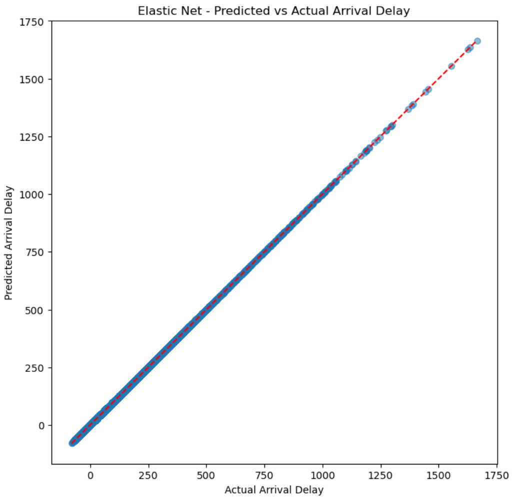


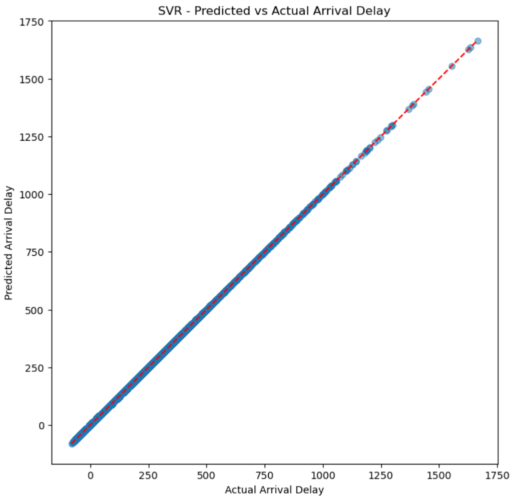
Elastic Net CV RMSE with Std Dev
Mean CV RMSE
40
35
30
25
20
15
10
alpha=0.01
l1_ratio=0.0
alpha=0.01
l1_ratio=0.3333333333333333
alpha=0.01
l1_ratio=0.6666666666666666
alpha=0.01
l1_ratio=1.0
alpha=0.21544346900318834
l1_ratio=0.0
alpha=0.21544346900318834
l1_ratio=0.3333333333333333
alpha=0.21544346900318834
l1_ratio=0.6666666666666666
alpha=0.21544346900318834
l1_ratio=1.0
alpha=4.641588833612775
l1_ratio=0.0
alpha=4.641588833612775
l1_ratio=0.3333333333333333
alpha=4.641588833612775
l1_ratio=0.6666666666666666
alpha=4.641588833612775
l1_ratio=1.0
alpha=100.0
l1_ratio=0.0
alpha=100.0
l1_ratio=0.3333333333333333
alpha=100.0
l1_ratio=0.6666666666666666
alpha=100.0
l1_ratio=1.0
Hyperparameter Setting

Mean + Std Dev

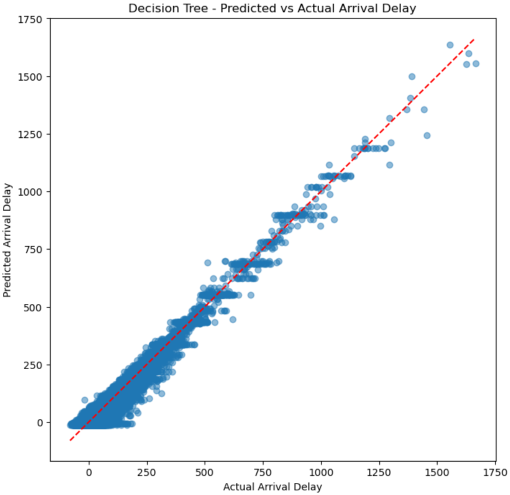

Decision Tree CV RMSE with Std Dev
Mean CV RMSE
27.5
25.0
22.5
20.0
17.5
15.0
12.5
max_depth=3
max_depth=7
max_depth=10
Hyperparameter Setting

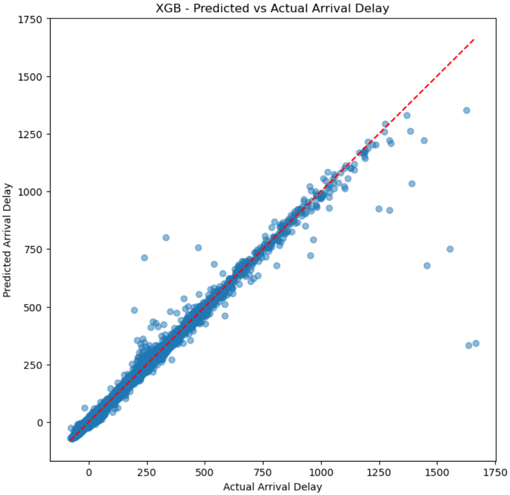
Scatter Plots



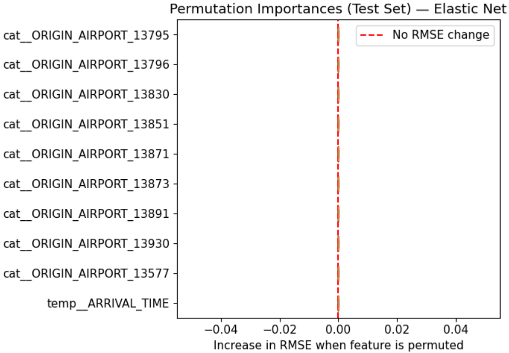


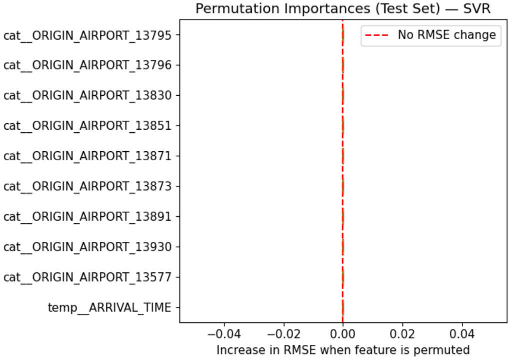
Scatter Plots



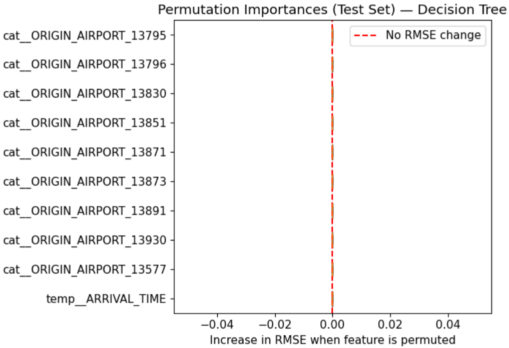


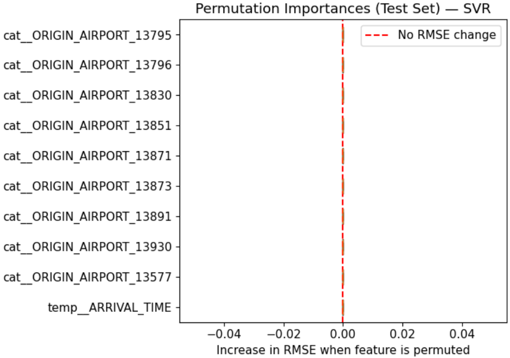
Perturbation Importance





Perturbation Importance





Interpretability

- Ran perturbation importance on best models (SHAP was too computationally expensive)
- Used linear-type models and a decision tree - easier to interpret
- These results seem suspicious - I think there is some implicit data leakage
- I left in some features, like ‘WHEELS_ON’, and ‘ARRIVAL_TIME’, that won’t be known at inference time
- In hindsight, an easy fix for the report