

Linear Regression Analysis of Network Traffic Data

Course: Applied Machine Learning & Data Analytics

Team Members: Gabil Gurbanov, Vusal Shirinbayli

Dataset: Internet Firewall Logs

Professor: Dr. Samir Rustamov

Motivation & Problem Statement

Why this problem?

- Network firewalls generate large volumes of traffic data
- Understanding relationships between traffic features helps:
 - detect anomalies
 - optimize network policies
 - estimate data transfer load

Problem Statement

- Can we model and predict network traffic behavior using regression techniques?
- How do different predictors influence the amount of transferred data?

Goal

- Apply linear regression models to analyze and predict firewall traffic metrics.

Dataset Description

Dataset Source

- Internet Firewall Data Set
- Kaggle & UCI Machine Learning Repository

Observations

- Network traffic logs collected from a real firewall system

Key Variables

- packets – number of packets
- bytes – total transmitted bytes (response variable)
- duration – connection duration
- action – firewall decision (Allow / Drop / Reset)

Why this dataset?

- Real-world, structured, numeric + categorical
- Suitable for simple, multiple, and interaction regression

Correlation Matrix

Dataset Source

- Internet Firewall Data Set
- Kaggle & UCI Machine Learning Repository

Why are bytes and packets almost perfectly correlated?

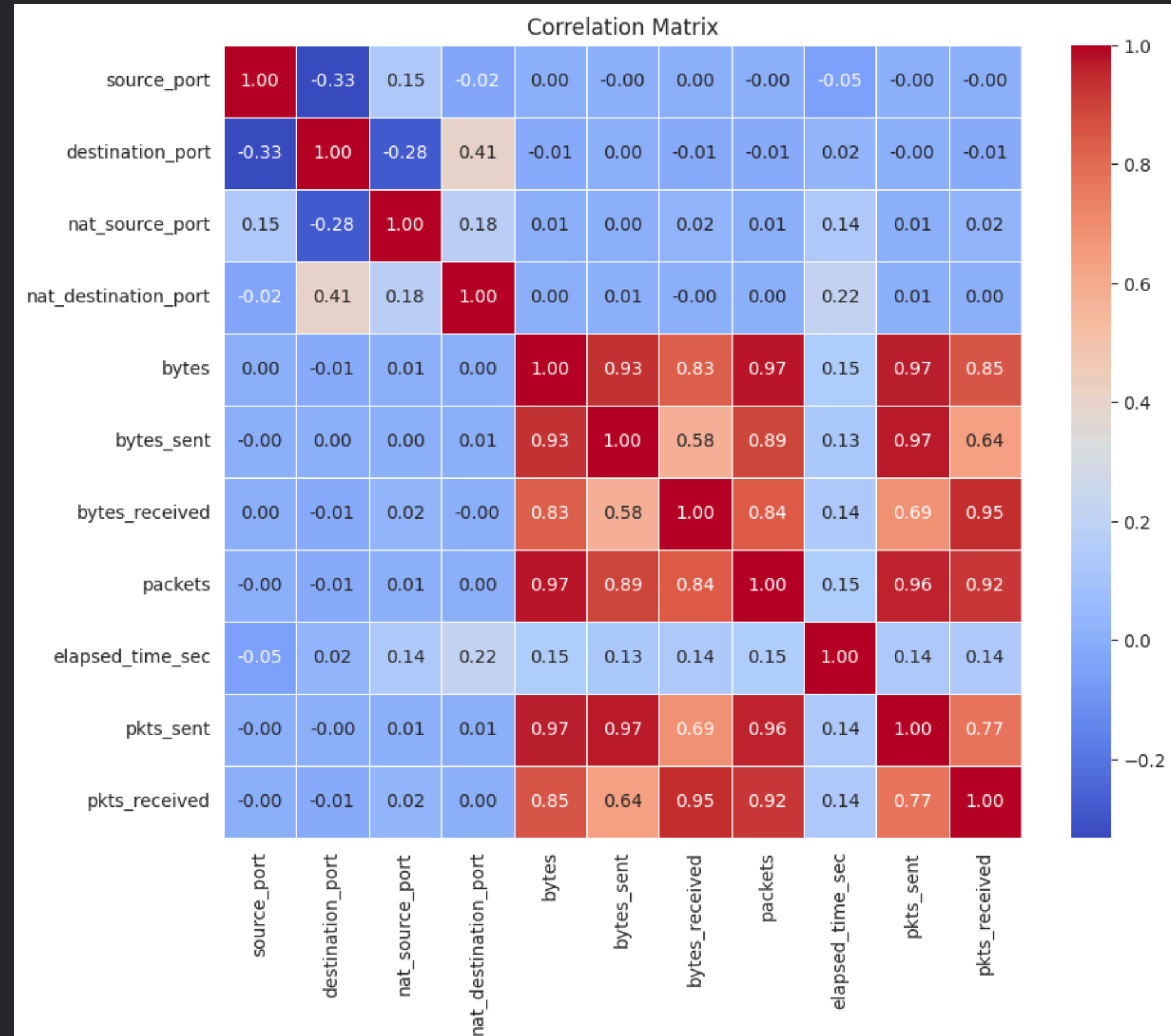
- Because total transmitted bytes are approximately proportional to the number of packets multiplied by average packet size. Therefore, a strong linear relationship is expected.

Does high correlation cause problems?

- Yes. It may cause multicollinearity in multiple regression, leading to unstable coefficient estimates. That is why model interpretation must be done carefully.

For example:

- $\text{corr}(\text{bytes}, \text{packets}) = 0.97$
- $\text{corr}(\text{bytes}, \text{pkts_sent}) = 0.97$



Methodology Overview

Models Used

1. Simple Linear Regression (1 predictor)
2. Multiple Linear Regression (multiple predictors)
3. Polynomial Regression (non-linear trend)

Estimation Technique

- Ordinary Least Squares (OLS)

Evaluation Metrics

- RSS
- RSE
- R^2 and Adjusted R^2
- t-statistics and p-values
- Prediction intervals
- RMSE (for polynomial comparison)

Simple Linear Regression

Model

- $\text{bytes} = \beta_0 + \beta_1 \cdot \text{packets} + \varepsilon$

Model Accuracy

- R^2 indicates strong linear relationship
- RSE measures average prediction error
- Pearson correlation confirms positive association

Results

- Estimated coefficients obtained using OLS
- RSS calculated to measure residual error
- t-statistic & p-value show packets is statistically significant

Interpretation

- As the number of packets increases, transmitted bytes increase linearly

| OLS Regression Results | | | | | | |
|------------------------|------------------|----------|---------------------|-------------------|-----------|-----------|
| ===== | | | | | | |
| Dep. Variable: | bytes | | R-squared: | 0.949 | | |
| Model: | OLS | | Adj. R-squared: | 0.949 | | |
| Method: | Least Squares | | F-statistic: | 1.230e+06 | | |
| Date: | Wed, 11 Feb 2026 | | Prob (F-statistic): | 0.00 | | |
| Time: | 03:20:06 | | Log-Likelihood: | -1.0137e+06 | | |
| No. Observations: | 65532 | | AIC: | 2.027e+06 | | |
| Df Residuals: | 65530 | | BIC: | 2.027e+06 | | |
| Df Model: | 1 | | | | | |
| Covariance Type: | nonrobust | | | | | |
| ===== | | | | | | |
| | coef | std err | t | P> t | [0.025 | 0.975] |
| ----- | | | | | | |
| const | -1.259e+04 | 4937.297 | -2.549 | 0.011 | -2.23e+04 | -2908.462 |
| packets | 1066.5281 | 0.962 | 1109.017 | 0.000 | 1064.643 | 1068.413 |
| ===== | | | | | | |
| Omnibus: | 249721.957 | | Durbin-Watson: | 1.997 | | |
| Prob(Omnibus): | 0.000 | | Jarque-Bera (JB): | 2102765725924.864 | | |
| Skew: | -86.629 | | Prob(JB): | 0.00 | | |
| Kurtosis: | 27753.185 | | Cond. No. | 5.14e+03 | | |
| ===== | | | | | | |

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The condition number is large, 5.14e+03. This might indicate that there are strong multicollinearity or other numerical problems.

--- Additional Metrics ---

Residual Sum of Squares (RSS): 104,640,014,180,666,208.00

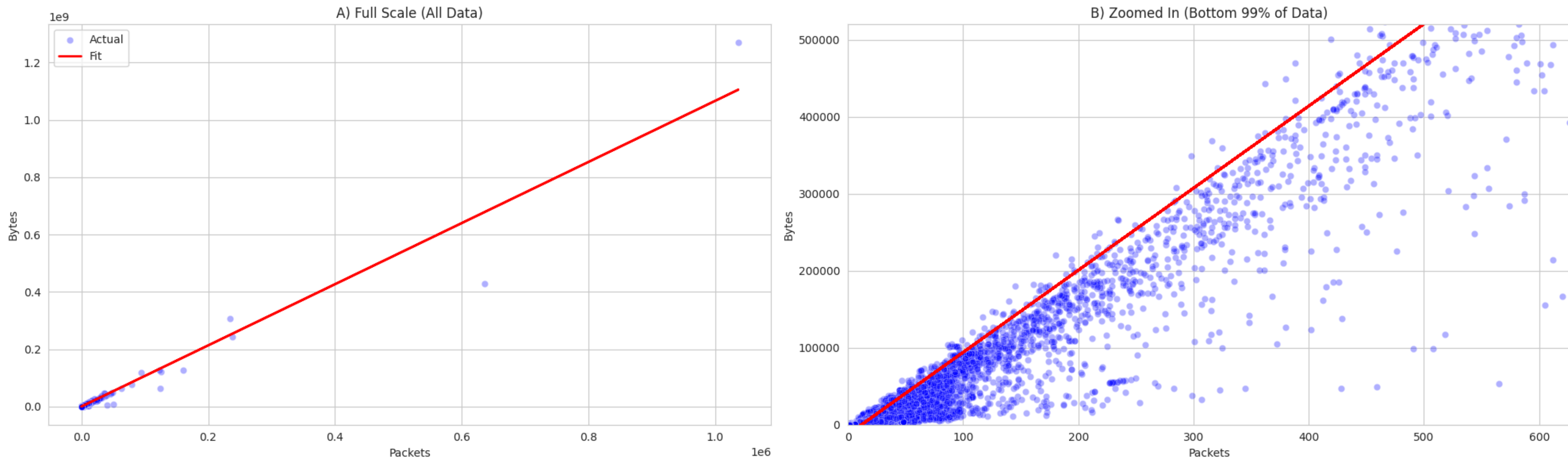
Residual Standard Error (RSE): 1,263,655.85

Correlation (r): 0.9744

Simple Linear Regression

Model

- $\text{bytes} = \beta_0 + \beta_1 \cdot \text{packets} + \varepsilon$



Multiple Linear Regression

Model

- $\text{bytes} = \beta_0 + \beta_1 \cdot \text{packets} + \beta_2 \cdot \text{duration} + \varepsilon$

Key Questions Answered

- Is at least one predictor useful? → Yes (F-test)
- Do all predictors matter? → Checked using individual p-values
- How well does the model fit? → Improved R^2 vs simple model

Insight

- Multiple predictors explain more variance than a single predictor

--- Multiple Regression Summary ---

OLS Regression Results

```
=====
Dep. Variable:          bytes      R-squared:                0.961
Model:                  OLS        Adj. R-squared:           0.961
Method:                 Least Squares    F-statistic:           5.418e+05
Date:                   Wed, 11 Feb 2026  Prob (F-statistic):       0.00
Time:                   03:20:11      Log-Likelihood:        -1.0049e+06
No. Observations:      65532         AIC:                   2.010e+06
Df Residuals:          65528         BIC:                   2.010e+06
Df Model:               3
Covariance Type:       nonrobust
=====
```

| | coef | std err | t | P> t | [0.025 | 0.975] |
|------------------|------------|----------|---------|-------|-----------|---------|
| const | -8268.6276 | 4422.793 | -1.870 | 0.062 | -1.69e+04 | 400.047 |
| packets | 650.2804 | 3.061 | 212.454 | 0.000 | 644.281 | 656.280 |
| elapsed_time_sec | 151.9689 | 14.446 | 10.520 | 0.000 | 123.655 | 180.283 |
| pkts_sent | 688.3208 | 4.873 | 141.265 | 0.000 | 678.771 | 697.871 |

```
=====
Omnibus:                207040.297    Durbin-Watson:           1.996
Prob(Omnibus):           0.000        Jarque-Bera (JB):       1138612154395.159
Skew:                   -48.831        Prob(JB):               0.00
Kurtosis:               20423.287      Cond. No.                6.15e+03
=====
```

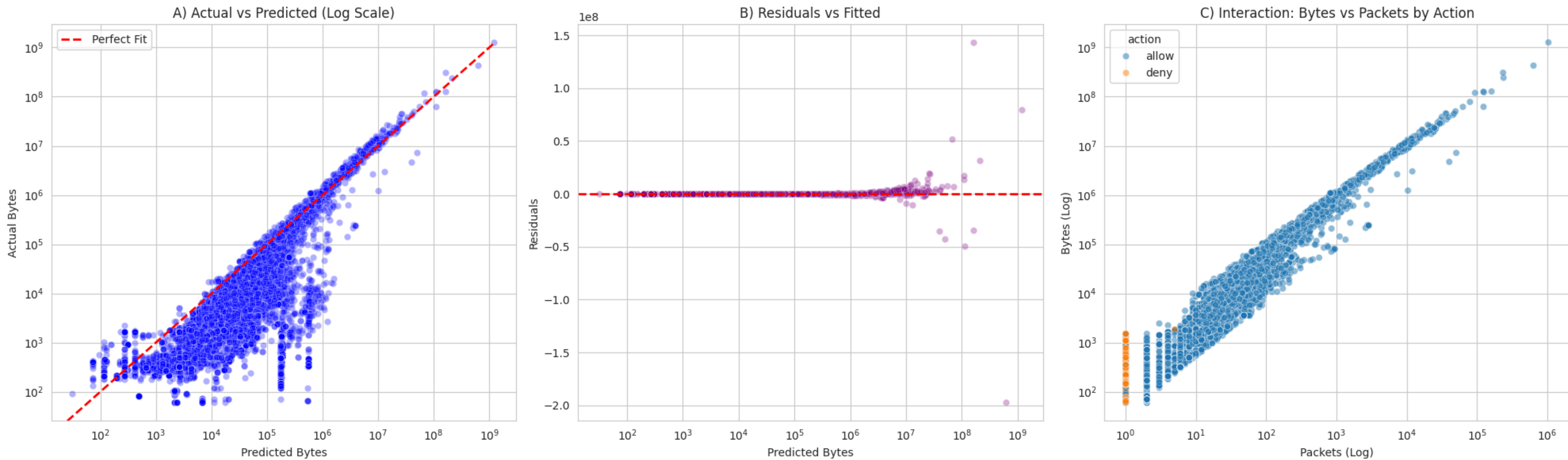
Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 6.15e+03. This might indicate that there are strong multicollinearity or other numerical problems.

Multiple Linear Regression

Model

- $\text{bytes} = \beta_0 + \beta_1 \cdot \text{packets} + \beta_2 \cdot \text{duration} + \varepsilon$



Prediction & Interaction Analysis

Interaction Analysis - Model with interaction

• $\text{bytes} = \beta_0 + \beta_1 \cdot \text{packets} + \beta_2 \cdot \text{action} + \beta_3 (\text{packets} \times \text{action})$

Prediction

- Given new predictor values, the model:
 - predicts expected bytes
 - provides a 95% prediction interval
- Interval reflects uncertainty in real-world observations

Finding

- The effect of packets on bytes depends on firewall action
- Shows different slopes for allowed vs dropped traffic

| --- Interaction Statistical Test --- | | | | | | |
|--------------------------------------|------------|----------|---------|-------|-----------|-----------|
| | coef | std err | t | P> t | [0.025 | 0.975] |
| Intercept | -2.118e+04 | 7270.651 | -2.913 | 0.004 | -3.54e+04 | -6931.185 |
| C(action)[T.deny] | 2.114e+04 | 1.39e+05 | 0.152 | 0.879 | -2.51e+05 | 2.93e+05 |
| packets | 1066.5613 | 1.073 | 993.733 | 0.000 | 1064.458 | 1068.665 |
| packets:C(action)[T.deny] | -937.4536 | 1.38e+05 | -0.007 | 0.995 | -2.71e+05 | 2.69e+05 |

Forward & Backward iteration

```
print("---- 1. Forward Selection ----")
# Start with no variables, add the best one step-by-step
selected_features = []
remaining_features = potential_features.copy()

while remaining_features:
    best_pval = 1.0
    best_feature = None

    for feature in remaining_features:
        # Try adding this feature to what we already have
        trial_features = selected_features + [feature]
        X_trial = df[trial_features]
        X_trial = sm.add_constant(X_trial)
        model = sm.OLS(df[target], X_trial).fit()

        # Look at the p-value of the *newly added* feature
        pval = model.pvalues[feature]

        if pval < best_pval:
            best_pval = pval
            best_feature = feature

    # If the best candidate is significant (p < 0.05), keep it
    if best_pval < 0.05:
        selected_features.append(best_feature)
        remaining_features.remove(best_feature)
        print(f"Added '{best_feature}' (p-value: {best_pval:.4e})")
    else:
        print("No more significant features found.")
        break

print(f"Final Forward Selection: {selected_features}")
```

```
print("\n--- 2. Backward Selection ---")
# Start with ALL variables, remove the worst one step-by-step
current_features = potential_features.copy()

while current_features:
    X_curr = df[current_features]
    X_curr = sm.add_constant(X_curr)
    model = sm.OLS(df[target], X_curr).fit()

    # Find the feature with the HIGHEST p-value (worst predictor)
    # We skip 'const' because we always want an intercept
    pvalues = model.pvalues.drop('const')
    worst_pval = pvalues.max()
    worst_feature = pvalues.idxmax()

    if worst_pval > 0.05:
        print(f"Removed '{worst_feature}' (p-value: {worst_pval:.4e})")
        current_features.remove(worst_feature)
    else:
        print("All remaining features are significant.")
        break

print(f"Final Backward Selection: {current_features}")
```

```
Available features for selection: ['packets', 'elapsed_time_sec', 'pkts_sent', 'pkts_received']
--- 1. Forward Selection ---
Added 'packets' (p-value: 0.0000e+00)
Added 'pkts_sent' (p-value: 0.0000e+00)
Added 'elapsed_time_sec' (p-value: 7.3342e-26)
Added 'pkts_received' (p-value: 1.2984e-06)
Final Forward Selection: ['packets', 'pkts_sent', 'elapsed_time_sec', 'pkts_received']

--- 2. Backward Selection ---
All remaining features are significant.
Final Backward Selection: ['packets', 'elapsed_time_sec', 'pkts_sent', 'pkts_received']
```

Polynomial Regression

Model

- $\text{bytes} = \beta_0 + \beta_1 x + \beta_2 x^2 + \varepsilon$

However... about risk of overfitting

Linear Regression RMSE: 418,728.70

Polynomial Regression (Deg 2) RMSE: 511,282.77

Result: Polynomial regression fits WORSE (Overfitting).

Why polynomial?

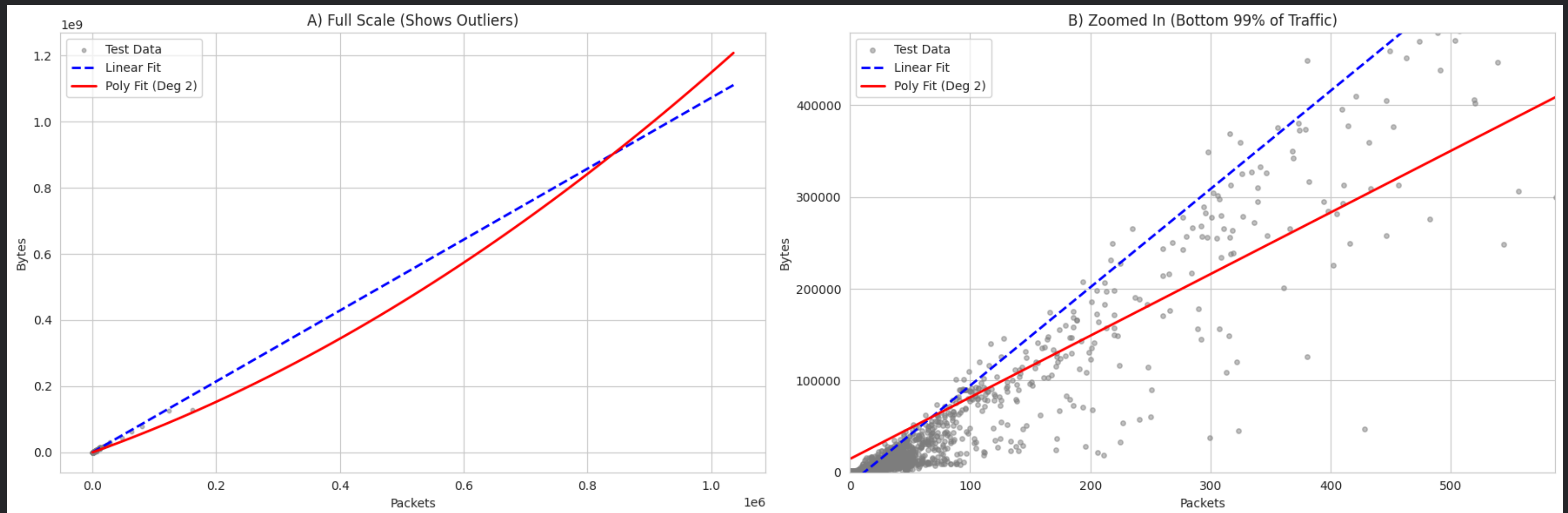
- Linear models may not capture curvature in traffic patterns

Evaluation

- Compared degree-1 vs degree-2 models
- Used RMSE on test data

Result

- Polynomial regression slightly improves fit
- Risk of overfitting if degree increases further



Model Comparison & Insights

| Model | Strength | Limitation |
|-------------|--------------------|-------------------|
| Simple LR | Interpretable | Limited accuracy |
| Multiple LR | Better fit | Assumes linearity |
| Polynomial | Captures curvature | Overfitting risk |

Key Insight

- Multiple regression with interaction provides the best balance
- Linear models remain effective for structured network data

Conclusion

- Linear regression successfully models firewall traffic
- Multiple predictors and interactions significantly improve accuracy
- Polynomial regression captures non-linear patterns cautiously

- Adding Time and Port increased complexity but added negligible accuracy generally (< 0.001 improvement).
- The volume of data transferred is almost entirely determined by the number of packets sent, regardless of connection duration.

| COURSE PROJECT 1 – FINAL REPORT SUMMARY | |
|---|--|
| 1. Apply Linear Regression to Single Predictor (Source Code: Cell 4) | |
| a) Estimate coefficients and RSS: | |
| – Intercept (b0): | -12585.5662 |
| – Slope (b1) for Packets: | 1066.5281 |
| – Residual Sum of Squares (RSS): | 104,640,014,180,666,208.00 |
| b) Calculate t-statistic and p-value: | |
| – t-statistic (for packets): | 1109.0174 |
| – p-value: | 0.0000e+00 |
| – (Is it significant? YES) | |
| c) Assess Overall Accuracy: | |
| – R-squared: | 0.9494 (The model explains 94.94% of the variance) |
| – Residual Standard Error (RSE): | 1,263,655.85 |
| – Correlation (r): | 0.9744 |
| 2. Apply Linear Regression to Multiple Predictors (Source Code: Cell 5) | |
| a) Is at least one predictor useful? (Global F-test) | |
| – F-statistic p-value: | 0.0000e+00 |
| – Answer: | YES (Since p-value < 0.05, at least one predictor is useful). |
| b) Do all predictors help to explain Y? | |
| – Individual p-values: | |
| – * packets: | p=0.0000e+00 -> USEFUL (Significant) |
| – * elapsed_time_sec: | p=7.3342e-26 -> USEFUL (Significant) |
| – * pkts_sent: | p=0.0000e+00 -> USEFUL (Significant) |
| c) How well does the model fit the data? | |
| – R-squared: | 0.9612 |
| – Adj. R-squared: | 0.9612 (Adjusted for number of predictors) |
| d) Prediction Example & Accuracy: | |
| – For an average connection, the model predicts: | 97,123.95 bytes |
| – Raw 95% Prediction Interval: | [-2,070,740.09, 2,264,987.99] |
| – 95% Prediction Interval: | [0.00, 2,264,987.99] |
| – (This interval represents the range where a new observation is likely to fall.) | |
| e) Analyze Interactions (Action vs Packets): | |
| – Significant Interaction Found: | NO |
| 3. Apply Polynomial Regression Model (Source Code: Cell 6) | |
| – Linear Model RMSE: | 418,728.70 |
| – Polynomial Model RMSE: | 511,282.77 |
| – Comparison Result: | Linear is BETTER |
| – Note: | The polynomial model performed worse, likely due to overfitting or the data being strictly linear. |

Thanks for attention