PREDICTING AUTOMOBILE FUEL EFFICIENCY USING SVR AND ANN

Wyatt Laughner and Azaria Reed (Group 9)

CS 430: Survey of Artificial Intelligence

Dr. Chaity Banerjee Mukherjee

December 2025

PROJECT DESCRIPTION

Goals of the Project:

- Predict **miles per gallon (mpg)** from the UCI Auto MPG dataset.
- Apply two machine learning approaches:
  - *Support Vector Regressor (SVR)*
  - *Artificial Neural Network (ANN)*
- Compare their performance using the mean squared error (MSE) metric.

What We Did:

- Loaded and cleaned the dataset
- Preprocessed the features
- Trained both models using **existing libraries and pipelines**
- Evaluated and compared accuracy

DATASET DESCRIPTION

Dataset: UCI Auto MPG Dataset

Instances: 398

ID: Car name

Features:

- Displacement
- Cylinders
- Horsepower
- Weight
- Acceleration
- Model year
- Origin

Target:

- MPG (automobile fuel efficiency)

Issues Identified:

- Missing values for horsepower

DATA PREPROCESSING

Steps We Took:

1. **Read the dataset** using pandas.read_csv() with whitespace delimiter.
2. **Converted "?" to NaN** using na_values="?".
3. **Dropped missing rows** using dropna().
4. **Selected the 7 numeric features**.
5. **Performed 80/20 train-test split** using train_test_split().

Why This Matters:

- Removes invalid entries
- Matches preprocessing from class homework

SVR APPROACH

- Learns nonlinear patterns
- Performs well on small datasets
- Widely used baseline regression model

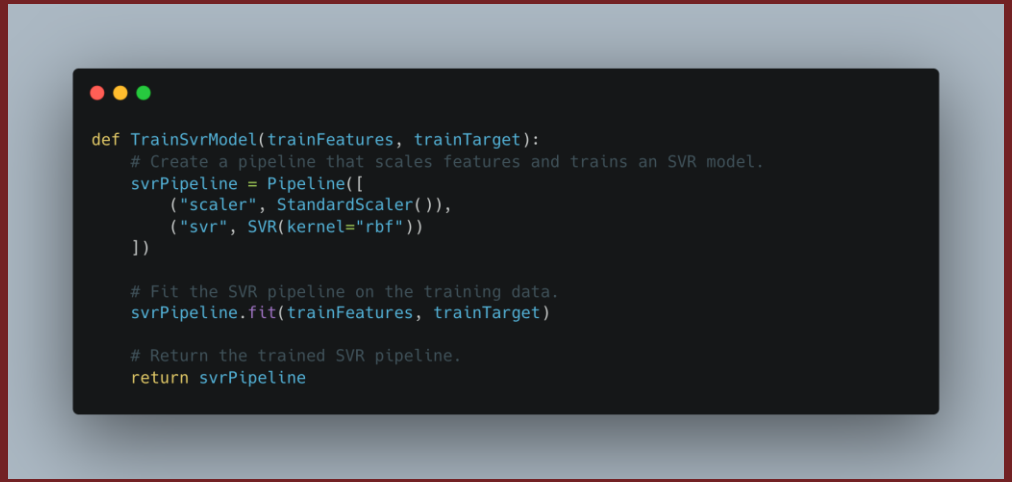
Library Used: scikit-learn (sklearn.svm.SVR)

Pipeline Steps:

- Scale features using StandardScaler()
- Train SVR with RBF kernel

Code Structure:

```
Pipeline([
   ("scaler", StandardScaler()),
   ("svr", SVR(kernel="rbf"))
])
```



```python
def TrainSvrModel(trainFeatures, trainTarget):
    # Create a pipeline that scales features and trains an SVR model.
    svrPipeline = Pipeline([
        ("scaler", StandardScaler()),
        ("svr", SVR(kernel="rbf"))
    ])

    # Fit the SVR pipeline on the training data.
    svrPipeline.fit(trainFeatures, trainTarget)

    # Return the trained SVR pipeline.
    return svrPipeline
```

ANN APPROACH

- Learns deeper nonlinear relationships
- Often outperforms SVR with enough training
- Works well with continuous prediction tasks

Library Used: scikit-learn (MLPRegressor)

Note: We used a small incremental ANN to generate the training vs. validation convergence curve, and a final ANN pipeline for computing the test MSE.

Architecture in Our Code:

- Hidden layers: (64, 32)
- Activation: ReLU
- Optimizer: Adam
- Max iterations: 1000

Pipeline:

```
Pipeline([
   ("scaler", StandardScaler()),
   ("mlp", MLPRegressor(...))
])
```

```python
def TrainAnnModel(trainFeatures, trainTarget):
    # Create a pipeline that scales features and trains an ANN model.
    annPipeline = Pipeline([
        ("scaler", StandardScaler()),
        ("mlp", MLPRegressor(
            hidden_layer_sizes=(64, 32),
            activation="relu",
            solver="adam",
            max_iter=1000,
            random_state=RANDOM_STATE
        ))
    ])

    # Fit the ANN pipeline on the training data, suppressing convergence warnings.
    with warnings.catch_warnings():
        warnings.filterwarnings("ignore", category=ConvergenceWarning)
        annPipeline.fit(trainFeatures, trainTarget)

    # Return the trained ANN pipeline.
    return annPipeline
```

EXPERIMENTAL SETUP

Train/Test Split:

- 80% Training
- 20% Testing
- random_state = 42 for reproducibility

Evaluation Metric:

- Mean Squared Error (MSE)
  - *Lower = better*

Reason for MSE:

- Used in homework
- Standard regression loss

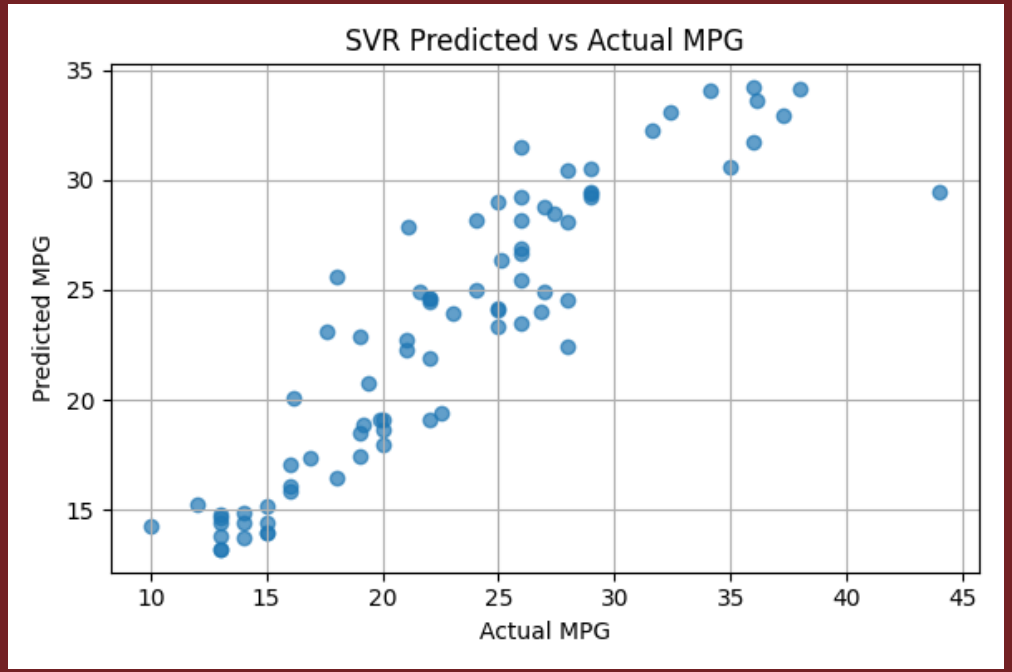
SVR RESULTS

Support Vector Regressor

- Test MSE: 9.27383

Interpretation:

- The model predicts mpg values with modest error
- SVR handles nonlinear relationships but sometimes underfits small datasets
- Reasonable performance given default hyperparameters



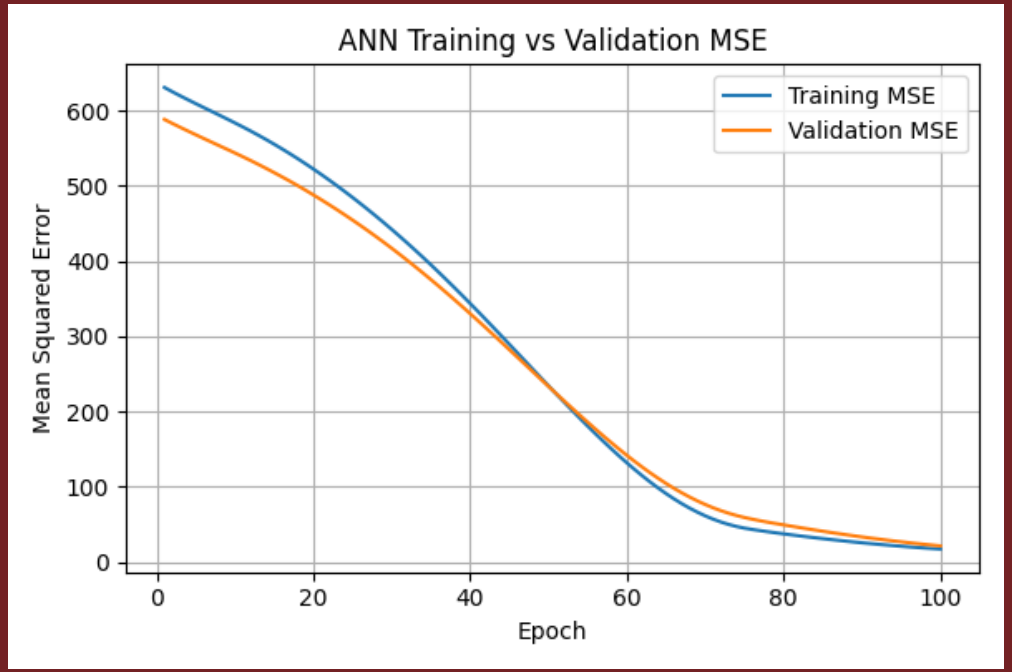
ANN RESULTS

Artificial Neural Network

- Test MSE: 7.38223

Interpretation:

- Lower error than SVR
- ANN captured more complex patterns
- Increasing hidden layers or epochs could improve accuracy further



MODEL COMPARISON

Conclusion:

- The ANN performed better on the Auto MPG prediction task due to its ability to learn more complex relationships among features.

Model	Test MSE	Notes
SVR	~9.27	Good baseline, simpler
ANN	~7.38	Best performance

KEY TAKEAWAYS

- Cleaning and scaling is crucial
- Both SVR and ANN can model mpg prediction
- ANN produced the lowest MSE
- Pipelines kept code simple, clean, and reproducible

REFERENCES AND ACKNOWLEDGMENTS

Dataset: UCI Machine Learning Repository (Auto MPG dataset)

Libraries Used:

- pandas
- scikit-learn

Acknowledgments:

- Professor: Dr. Chaity Banerjee Mukherjee
- Teaching Assistant: Shania Shakri