# Major Project: Sales Prediction Project

## Table of Contents

1. [Project Overview](#project-overview)
2. [Installation & Setup](#installation--setup)
3. [Data Requirements](#data-requirements)
4. [Technical Implementation](#technical-implementation)
5. [Model Optimization Methods](#model-optimization-methods)
6. [Usage Instructions](#usage-instructions)
7. [Output Interpretation](#output-interpretation)
8. [Troubleshooting](#troubleshooting)
9. [Technical Dependencies](#technical-dependencies)

## Project Overview

### Purpose

This project implements a machine learning solution for predicting sales based on data spending across different media channels (TV, Radio, and Newspaper). It includes multiple optimization techniques and model evaluation methods to achieve the best possible predictions.

### Key Features

* Multiple regression algorithms implementation
* Automated feature engineering
* Hyperparameter optimization
* Comprehensive model evaluation
* Performance visualization
* Feature importance analysis

### Target Users

* Data Scientists
* Marketing Analysts
* Business Intelligence Teams
* Sales Forecasting Teams

## Installation & Setup

### Prerequisites

* Python 3.8 or higher
* pip (Python package manager)

### Required Libraries

pip install pandas numpy scikit-learn matplotlib seaborn

## Data Requirements

### Input Data Format

The input dataset should be a CSV file named ‘data.csv’ with the following columns: - TV: Data spending on TV (numerical) - Radio: Data spending on Radio (numerical) - Newspaper: Data spending on Newspaper (numerical) - Sales: Actual sales numbers (target variable, numerical)

### Data Format Example

TV,Radio,Newspaper,Sales  
230.1,37.8,69.2,22.1  
44.5,39.3,45.1,10.4  
17.2,45.9,69.3,9.3

### Data Quality Requirements

* No missing values
* Numerical values only
* Positive numbers for all columns
* Sufficient data points (minimum 100 recommended)

## Technical Implementation

### Core Components

1. **Data Preprocessing**

* - StandardScaler for feature scaling  
  - PolynomialFeatures for feature engineering  
  - SelectKBest for feature selection

1. **Model Implementations**
   * Base Models:
     + Linear Regression
     + Ridge Regression
     + Lasso Regression
     + ElasticNet
     + Random Forest
2. **Optimization Pipeline**

* Data Loading → Preprocessing → Feature Engineering → Model Selection → Hyperparameter Tuning → Evaluation

## Model Optimization Methods

### 1. Feature Scaling

* Standardizes features to mean=0 and variance=1
* Improves model convergence
* Essential for distance-based algorithms

### 2. Polynomial Features

* Creates interaction terms
* Captures non-linear relationships
* Degree=2 polynomial transformations

### 3. Feature Selection

* Uses f\_regression for scoring
* Selects top K features
* Reduces model complexity

### 4. Algorithm Selection

| Algorithm | Strength | Use Case |
| --- | --- | --- |
| Linear Regression | Baseline | Simple linear relationships |

### 5. Hyperparameter Tuning

* Grid Search Cross-validation
* Optimized parameters per algorithm
* K-fold validation (k=5)

## Usage Instructions

### Basic Usage

1. Prepare your data:

* # Ensure your CSV file is in the correct format  
  df = pd.read\_csv('data.csv')

1. Run the model:

* python main.py

### Advanced Usage

param\_grid = {  
 'n\_estimators': [100, 200, 300],  
 'max\_depth': [None, 10, 20],  
 'min\_samples\_split': [2, 5, 10]  
}  
  
  
cv\_scores = cross\_val\_score(model, X, y, cv=5)

## Output Interpretation

### Model Performance Metrics

* RMSE (Root Mean Square Error)
  + Lower is better
  + Same units as target variable
* R² Score
  + Range: 0 to 1
  + Higher is better

### Visualization Outputs

1. **Performance Comparison Plot**
   * Bar chart of RMSE values
   * Compare different models
2. **Feature Importance Plot**
   * Bar chart of feature weights
   * Identifies key predictors

## Troubleshooting

### Common Issues

1. **FileNotFoundError**
   * Solution: Ensure ‘data.csv’ is in the correct directory
2. **ValueError: Missing required columns**
   * Solution: Check CSV file format matches requirements
3. **Memory Issues**
   * Solution: Reduce polynomial feature degree
   * Solution: Decrease cross-validation folds

### Error Messages

try:  
 # Your code  
except FileNotFoundError:  
 print("Error: CSV file not found")  
except ValueError as e:  
 print(f"Error in data: {e}")

## Technical Dependencies

### Core Dependencies

pandas==1.5.3  
numpy==1.23.5  
scikit-learn==1.2.2  
matplotlib==3.7.1  
seaborn==0.12.2

### System Requirements

* RAM: Minimum 8GB recommended
* CPU: Multi-core processor recommended
* Storage: 1GB free space
* OS: Windows/Linux/MacOS

### Version Control

git init  
git add .  
git commit -m "Initial commit"

**Conclusion**

**Summary**

This Sales Prediction Project provides a robust, production-ready machine learning solution for predicting sales based on media channel spending. The implementation combines industry-standard best practices in data preprocessing, model selection, and optimization techniques to deliver reliable sales forecasts.

**Key Achievements**

* Comprehensive implementation of multiple regression algorithms
* Automated feature engineering and selection pipeline
* Robust error handling and troubleshooting mechanisms
* Flexible architecture supporting both basic and advanced usage patterns
* Thorough documentation and usage instructions

**Technical Highlights**

* Scalable data preprocessing pipeline
* Advanced optimization techniques including polynomial feature generation
* Cross-validated model selection and hyperparameter tuning
* Extensive performance visualization capabilities
* Well-defined error handling and debugging support