Predicting and Analysing the Remaining Useful Life of EV Batteries

By Ayamullah Khan

***Abstract*—With the rise of Electric Vehicles (EV) and hence Lithium-ion batteries, estimating the Remaining Useful Life (RUL) of batteries has become critical to avoid a series of safety-related problems caused by continual battery use after its service life threshold. Battery capacity is used to describe battery State-of-Health (SoC) however measurement of capacity during the operation of EVs is difficult. We propose a method that uses measurable features such as discharge time and battery temperature to estimate RUL.**

Several learning algorithms are implemented which include Support Vector Regressors, Random Forests, Artificial Neural Networks and Boosting methods. Test results show that the proposed method estimates capacity accurately with little to no hyperparameter tuning. The model predictions are seen to hold for other batteries as well.

Model explainability is discussed in the context of the trained models. Insights drawn from Exploratory Data Analysis allow us to provide explanations for the models' working. Some measures for extending RUL are also suggested.

***Index Terms*—Li-ion batteries, RUL, State-of-Health (SoC), XAI**

I. Introduction

The rapid increase in automobiles worldwide has led to serious concerns regarding air pollution, global warming and depletion of natural resources. Such concerns have resulted in the need to develop safe, efficient and environmental-friendly modes of transportation. As the world embraces the rise of electric vehicles, Li-ion batteries are at the helm of fast-paced development. Among all the state-of-the-art storage technologies, Li-ion batteries have the highest level of energy storage density along with additional advantages such as low self-discharge, very long lifetimes and cycling performances.

For these reasons, it is essential that we have an efficient algorithm for predicting an EV battery's **Remaining Useful Life (RUL)** with high accuracy using features such as Voltage, Current and Temperature measured over a few cycles.

One charge cycle is the period of use from fully charged to discharged and fully recharged again. An average Li-ion can typically last for 300-500 charge cycles with each charge cycle reducing battery capacity [1]. A battery becomes practically unusable when its capacity drops below a certain percentage of the total capacity, depending on the manufacturer and the usage of the given battery. 70% is used as the threshold in the following study.

II. Background and Prior Work

Researchers have been looking for various methods to predict the RUL of a Li-ion battery based on measurable features. Each method aims to minimize the loss associated with the predictions made by the regression model used.

Saha and Goebel came up with an approach to predict End of Discharge (EoD) time of Li-ion batteries [2]. Two regression algorithms were selected: a low complexity Polynomial Regression Model, and an Artificial Neural Network representing a more complex approach. To compare the results of the above models, a **Particle Filter** based benchmark algorithm was used which relies on empirical models and measurement data to predict battery EoD.

Saha, Goebel and Saxena, in their paper presented at the MFPT 2008 [3], discussed two new approaches for predicting the RUL of a battery. The first is a **Relevance Vector Machine** (RVM), a Bayesian form of the Support Vector Machine (SVM). RVMs attempt to address the issue of the lack of probabilistic output of SVMs in a Bayesian framework and use a lot fewer kernel functions. The second is a Gaussian Progress Regression model, which works well on small datasets and can provide uncertainty measurements on the predictions.

III. Exploratory Data Analysis

The dataset used for the project was downloaded from this link [4] and was in MATLAB format. Python's `pymatreader` module was used to help load data for further processing. The dataset consisted of degradation data for 4 18650 Li-ion batteries discharged at a constant current of 2A until the voltage fell to 2.7V, 2.5V, 2.2V and 2.5V for each battery. The batteries were subjected to 3 operational profiles - charge, discharge and impedance at room temperature.

type	ambient_temp	data
charge	24	{'Voltage_measured': [3.70, 3.33...
discharge	24	{'Voltage_measured': [4.20, 4.20...
impedance	24	{'Sense_current': [(839.74-31.55j..

Table I: Few rows from the dataset

A. Data and Datatypes

The dataset contained an equal amount of data on charging and discharging operational profiles (Fig. 1). This was consistent with the fact that charging and discharging was done one after another. There was almost double the amount of impedance data. This was because impedance measurements were performed after both charge and discharge profiles.

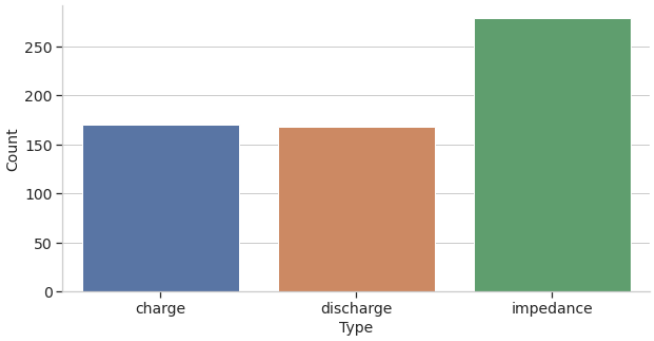

Fig. 1: Cycle Types

1) Charging Mode

During the charging cycle, the battery was charged in constant current (CC) mode till the voltage reached 4.2V, and subsequently in constant voltage (CV) mode till the current dropped to 20mA (see Fig. 2).

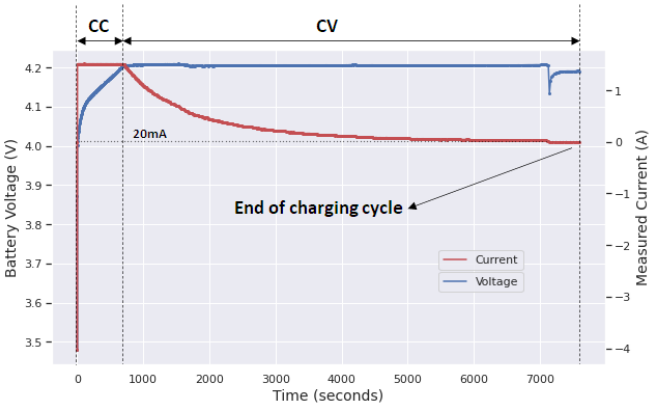

Fig. 2: Comparing Voltage and Current Profiles for Cycle 0

The data present under the charging cycle label included voltage and current at the battery, battery temperature, voltage and current at the charger as time-series data and an array labelled **Time** which contained time (in seconds) at which the measurements were made. Voltage and current at the load were analysed since measurements at the charger would be similar. As the battery deteriorated due to repeated charging-discharging, the time taken in constant current mode increased. Battery temperature did not show a strong correlation with battery degradation although the general trend was similar: rise in temperature during constant current phase and fall in constant voltage phase (Fig. 3).

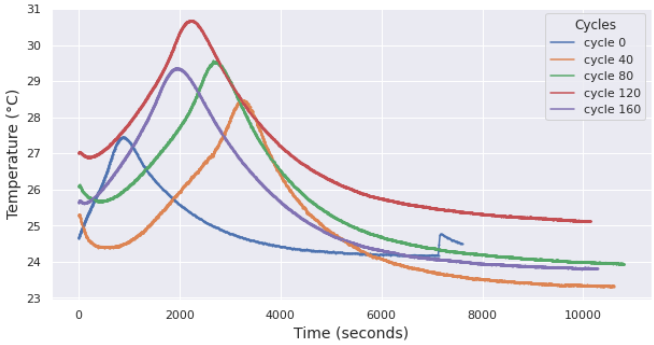

Fig. 3: Temperature profile for a few charging cycles

2) Impedance Mode

Impedance measurements were performed using **Electrochemical Impedance Spectroscopy (EIS)** frequency sweeps between 0.1Hz and 5kHz. In EIS, the battery is excited using a small AC perturbation and its response characterises battery impedance. Impedance measurements included sense current, battery current, current ratio (ratio of sense current and battery current), battery impedance, rectified impedance, estimated electrolytic and charge-transfer resistance (Re and Rct respectively). Out of all these features, **Re** and **Rct** were float values reported once per impedance cycle while the remaining were an array of complex values at select frequencies. The electrolytic and charge-transfer resistances were seen to increase as the battery degrades (Fig. 4 and Fig. 5). This is due to electrolytic depletion.

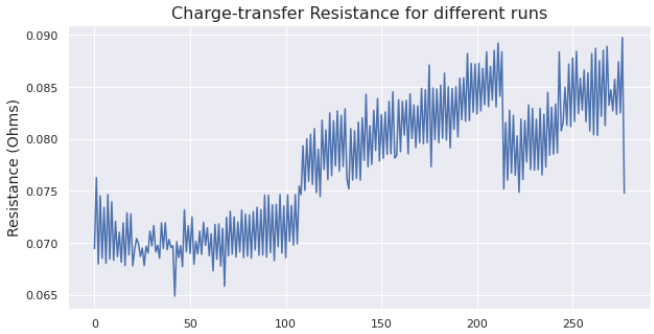

Fig. 4: Increasing trend in Electrolytic Resistance

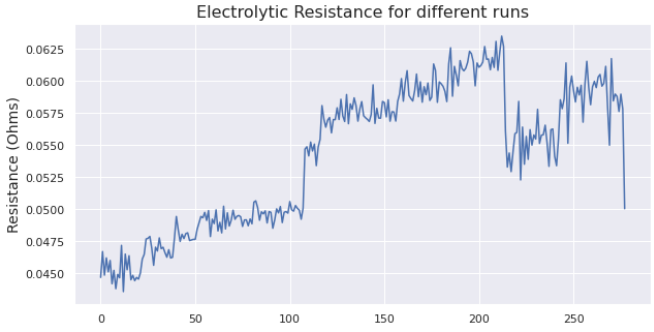

Fig. 5: Increasing trend in Charge-transfer Resistance

B. Discharge Mode

During discharge mode, measurements included discharge voltage and current at the battery and load, battery temperature, battery capacity (Ahr) and measurement times. Just like the charge cycle, measurements at the battery were analysed.

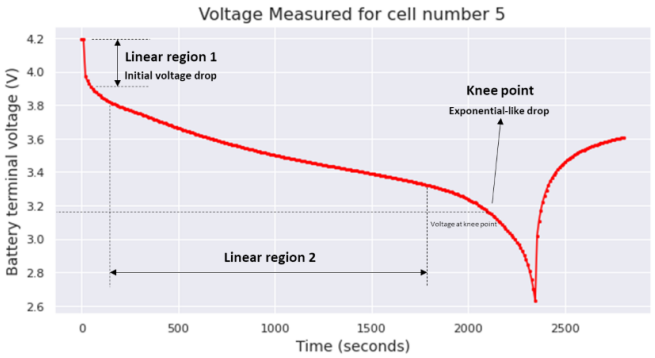

Fig. 6: Discharge Voltage measured at the Battery

The plot of voltage measured at the battery (Fig. 6) can be divided into 3 regions:

- Pseudo Linear Region I due to the sudden application of load current. (Fig. 7) shows that the slope of the region increases as the battery degrades. The slope of this region can also be connected to battery impedance
- Pseudo Linear Region II of reduced slope up to the knee point
- Steep exponential-like drop-off after the knee point

The load is switched off once the battery voltage drops to 2.6V. This is because the **Discharge Cut-off Voltage** of the Li-ion battery has been reached. The batteries under test were discharged beyond the discharge cut-off voltage to accelerate battery aging. Increase in battery terminal voltage beyond this point can be attributed to a diffusion of acid from the electrolyte to the plates. Fig. 7 indicates that total discharge time decreases as the battery degrades.

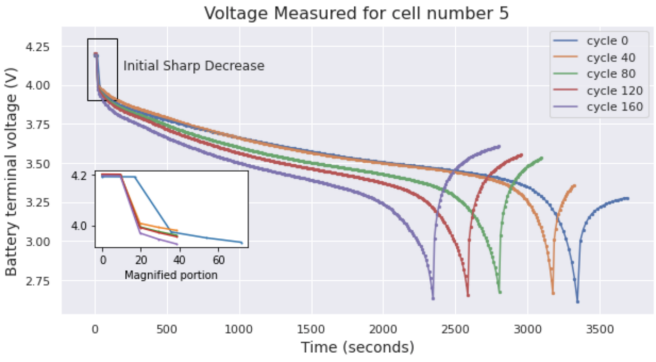

Fig. 7: Discharge Voltage measured at the Battery

Unlike the charging cycle, (Fig. 8) indicates that the maximum temperature attained by the battery during discharge increases due to degradation. Time taken to attain the maximum temperature is seen to decrease.

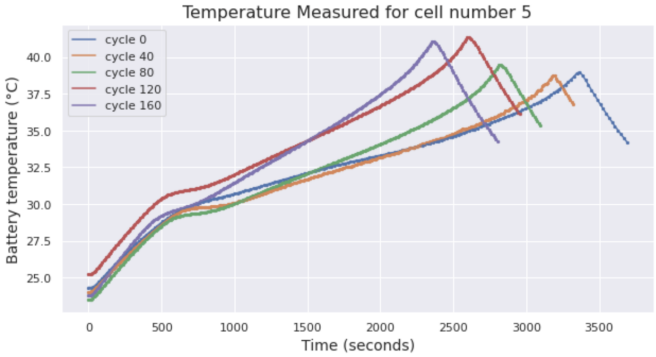

Fig. 8: Battery Temperature during discharge

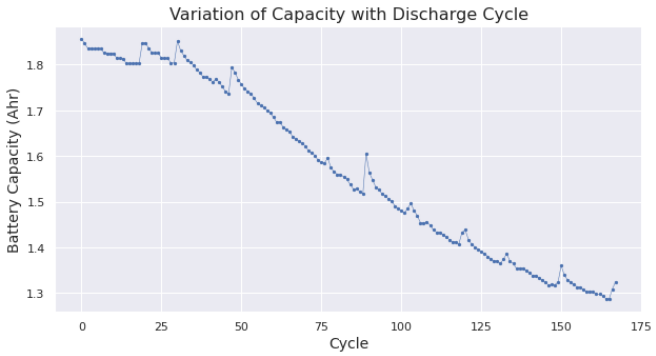

Fig. 9: Battery Capacity

Battery capacity is observed to decrease steadily as the battery deteriorates. The experiments are stopped at 30% fade in capacity beyond which the Li-ion battery is said to be unusable (Fig. 9).

IV. Machine Learning Workflow

From the previous section, it was evident that **Battery Capacity** could be used to estimate RUL. However, the measurement of battery capacity when an electric vehicle is running is difficult and in such cases, an indirect approach must be adopted. Features extracted from discharge mode data serve as a starting point to estimate capacity. Capacity is then used for RUL diagnosis.

A. Feature Extraction

The following features are extracted from discharge mode data after normalization:

- Time at which the battery is completely discharged
- Time at which the battery voltage hits some specified values between 2.7V and 4V. The number of time samples is a hyperparameter that can be tuned. 16 had been chosen for the purposes of analysis
- Time at which the battery reaches its maximum temperature
- Maximum battery temperature

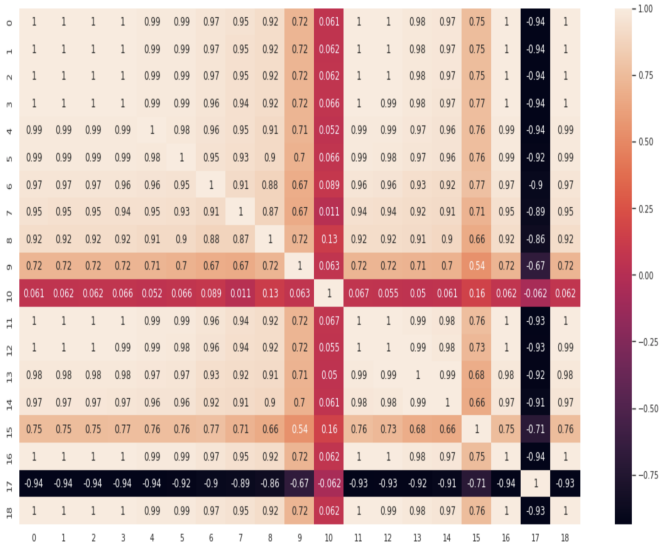

Fig. 10: Correlation Matrix of Extracted Features

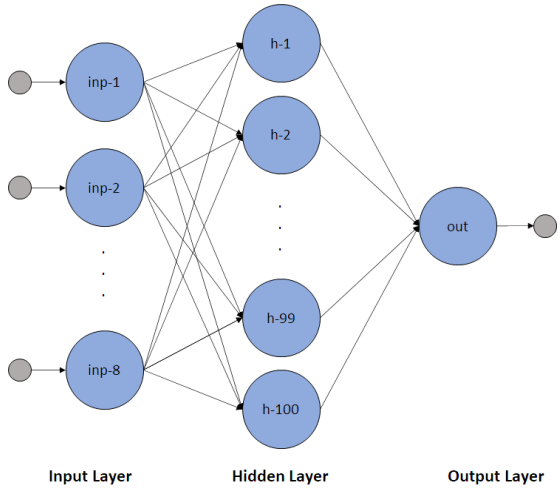

Fig. 11: The MLP model used, consisting of a single hidden layer with 100 neurons (h-1 to h-100)

There was high correlation between the extracted features (Fig. 10) and **Principal Component Analysis (PCA)** was used to reduce dimensionality of the data and extract uncorrelated features. PCA is performed by computing the eigenvectors of the data's Covariance Matrix. The eigenvectors corresponding to the largest eigenvalues are retained with the reconstruction error being proportional to the magnitude of the remaining eigenvalues.

1% of the standard deviation of L2-norm of the input features was used as the threshold for reconstruction. The reduced space consisted of only 8 features which can be used by learning models (Reconstruction error vs Number of retained features illustrated in Supplementary Fig. 1).

B. Learning Models

1) Support Vector Regression

Support Vector Machines are based on the concept of a maximum-margin classifier. Although the decision boundary for a vanilla SVM is linear, the **Kernel trick** allows for non-linearities by transforming to a higher dimensional feature space. Support Vector Regressors are the equivalents of SVMs for regression tasks. A **Radial Basis Kernel** was chosen for this task.

2) Multilayer Perceptron

A Multilayer Perceptron (MLP) is a feedforward network which uses the Backpropagation algorithm for training. The model consists of a network of interconnected **neurons** arranged in layers. There is a single input and output layer in an MLP but multiple possible *hidden layers*. The outputs of neurons in one layer are linearly combined and passed through a non-linear activation function before being fed to the neurons in the next layer. Python's `scikit-learn` module provides a prebuilt implementation of MLPs which can be integrated into the workflow directly (Fig. 11).

The inputs propagate through the model before arriving at the last layer. During training, the model's performance is scored using a *loss* metric. Backpropagation is used to improve model performance by updating neuron weights to optimise the loss function which was chosen to be **Mean Squared Error**.

3) Long Short-term Memory (LSTM)

An LSTM model is a Recurrent Neural Network architecture with feedback connections in addition to feedforward connections.

The model consists of an LSTM layer followed by a Dense layer. Hyperparameter tuning was performed using hyperparameter grids and the best results were obtained on using 8 neurons in the LSTM layer and 8 neurons in the Dense layer (Fig. 12). The LSTM layer had a *tanh* activation while the Recurrent layer had a *sigmoid* activation. Similar to the MLP, **Mean Squared Error** was chosen as the loss function.

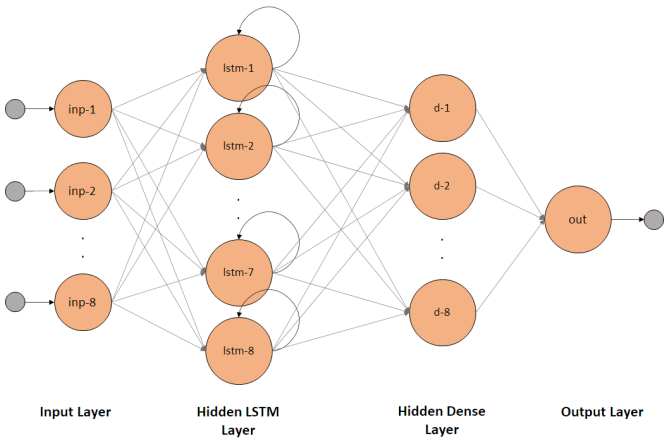

Fig. 12: The LSTM model architecture used, consisting of an LSTM layer and a Dense layer with 8 units each

4) Random Forest Regression

Random Forests are an Ensemble Learning decision-tree based method for classification and regression tasks. Random Forests work using the Bootstrap Aggregation, or *Bagging* ensemble technique.

Bootstrap: Several decision trees are generated and trained independently using rows sampled from the original dataset.

Aggregation: The outputs from all decision trees are then pooled to generate the final output.

Random Forests generally perform better than simple decision trees but are outperformed by gradient boosted trees, which are discussed in Section-IV-B5. This is also supported by the results in Table-II.

5) Boosting Methods

Boosting is another Ensemble Learning method that combines a set of weak learners into a strong learner to minimize training errors. Each model tries to compensate for the weaknesses of its predecessor. With each iteration, the weak rules from each classifier are combined to form one strong prediction rule.

Boosting algorithms differ in how they create and aggregate weak learners during the sequential process. Three types of boosting methods which were used for predicting Capacity using the selected features were:

1) Adaptive Boosting (AdaBoost): This method operates iteratively, identifying misclassified data points and adjusting their weights to minimize the training error. AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers.
2) Gradient Boosting: Works by sequentially adding predictors to an ensemble, with each one correcting for the errors of its predecessor. However, instead of changing weights of data points like AdaBoost, the gradient boosting trains on the residual errors of the previous predictor.
3) XGBoost: It is an implementation of gradient boosting designed for computational speed and scale. XGBoost leverages multiple cores on the CPU, allowing for learning to occur in parallel during training.

C. Model Explainability

Machine Learning models are often described as 'black boxes' because the mechanism by which they arrive at results is unknown. **Explainable AI** aims to come up with a set of tools to produce explainable models achieving high performance. **Partial Dependence Plots (PDPs)** are one such tool that show the effect of each feature on the output, assuming other features to be held constant. PDPs for the Support Vector Regressor were made for each of the 8 features. Two such PDPs are shown in Fig. 13 and Fig. 14. Feature 2 can be associated with the maximum temperature achieved by the battery while feature 5 can be associated with the total discharge time. This is consistent with observed trends in Fig. 7 and Fig. 8. Similar plots were also made for other models. Supplementary Fig. 2 compares the PDPs for the Random Forest, XGBoost and LightGBM Regression models.

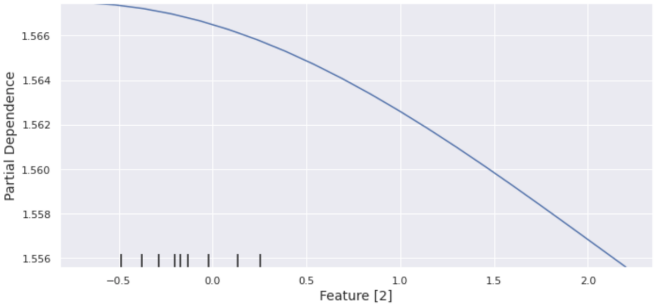

Fig. 13: SVR Partial Dependence Plot for Feature 2

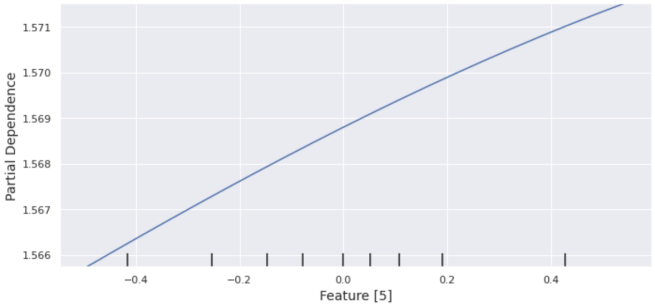

Fig. 14: SVR Partial Dependence Plot for Feature 5

V. Experiments and Results

A total of **7** regression models were trained on data from cell number 5 and tested on data from cells numbered **6**, **7** and **18**. Model performance was characterised in terms of Root Mean Squared Error and Coefficient of Determination (R2 Score) (Table-II).

Model	RMSE	R2 Score
XGBoost Regressor	0.0222	0.9809
Random Forest Regressor	0.0235	0.9785
Artificial Neural Network	0.0263	0.9731
LightGBM Regressor	0.0272	0.9712
AdaBoost Regressor	0.0292	0.9668
CatBoost Regressor	0.0364	0.9485
Support Vector Regressor	0.0692	0.8140

Table II: Model results on Cell Number 7

When run on other cells, it was observed that the XGBoost and Random Forest models achieved the best results. Evaluation results on cell numbers 6, 7 and 18 are described in Supplementary Tables 1, 2 and 3. Supplementary Fig. 3 shows the predicted vs. true labels plots for the Random Forest, XGBoost and LightGBM models on the data from cell number 7.

General Loss Metrics

- $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - predicted)^2}$
- $SS_{total} = \sum_i (y_i - \bar{y})^2$
- $SS_{residual} = \sum_i (y_i - predicted)^2$
- $R^2 Score = 1 - \frac{SS_{residual}}{SS_{total}}$