iR SOLUTIONS

Name: Shayan Umar

Designation: AI intern

Department: Artificial Intelligence

Date: 7-11-2025

Table of Contents

1. Theoretical Understanding:

1.1 Machine Learning and Its types:

1.1.1 Supervised Machine Learning:

Supervised learning models can make predictions after seeing lots of data with the correct answers and then discovering the connections between the elements in the data that produce the correct answers. This is like a student learning new material by studying old exams that contain both questions and answers. Once the student has trained on enough old exams, the student is well prepared to take a new exam. These ML systems are "supervised" in the sense that a human gives the ML system data with the known correct results.

1.1.2 Regression

A **regression model** predicts a numeric value. For example, a weather model that predicts the amount of rain, in inches or millimeters, is a regression model.

1.1.3 Classification

Classification models predict the likelihood that something belongs to a category. Unlike regression models, whose output is a number, classification models output a value that states whether or not something belongs to a particular category. For example, classification models are used to predict if an email is spam or if a photo contains a cat.

1.1.4 Unsupervised learning

Unsupervised learning models make predictions by being given data that does not contain any correct answers. An unsupervised learning model's goal is to identify meaningful patterns among the data. In other words, the model has no hints on how to categorize each piece of data, but instead it must infer its own rules.

A commonly used unsupervised learning model employs a technique called **clustering.** The model finds data points that demarcate natural groupings.

Clustering differs from classification because the categories aren't defined by you. For example, an unsupervised model might cluster a weather dataset based on temperature, revealing segmentations that define the seasons. You might then attempt to name those clusters based on your understanding of the dataset.

1.1.5 Reinforcement learning

Reinforcement learning models make predictions by getting rewards or penalties based on actions performed within an environment. A reinforcement learning system generates a policy that defines the best strategy for getting the most rewards.

Reinforcement learning is used to train robots to perform tasks, like walking around a room.

1.2 Linear Regression

Linear regression is a statistical technique used to find the relationship between variables. In an ML context, linear regression finds the relationship between features and a label**.**

Linear regression equation

In algebraic terms, the model would be defined as $y=mx+b$, where

- y is miles per gallon—the value we want to predict.
- m is the slope of the line.
- x is pounds—our input value.
- b is the y-intercept.

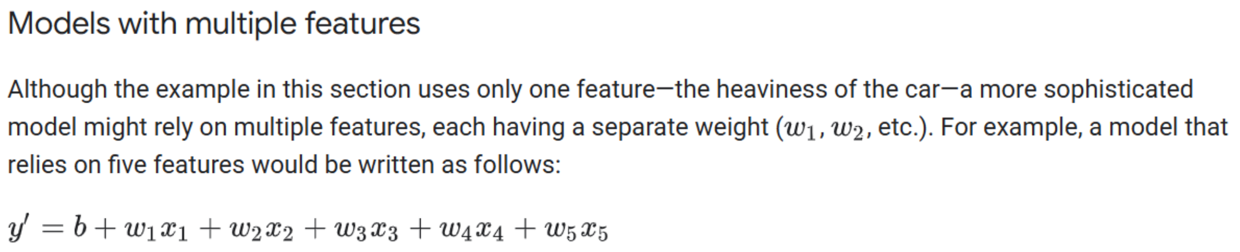
In ML, we write the equation for a linear regression model as follows:

$$y'=b+w1x1$$

where:

- y′ is the predicted label—the output.
- b is the **bias** of the model. Bias is the same concept as the y-intercept in the algebraic equation for a line. In ML, bias is sometimes referred to as w0. Bias is a **parameter** of the model and is calculated during training.
- w1 is the **weight** of the feature. Weight is the same concept as the slope m in the algebraic equation for a line. Weight is a **parameter** of the model and is calculated during training.
- x1 is a **feature**—the input.

During training, the model calculates the weight and bias that produce the best model.



Models with multiple features

Although the example in this section uses only one feature—the heaviness of the car—a more sophisticated model might rely on multiple features, each having a separate weight (w_1, w_2, etc.). For example, a model that relies on five features would be written as follows:

$$y' = b + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5$$

1.3 Gradient Descent:

Gradient descent is a mathematical technique that iteratively finds the weights and bias that produce the model with the lowest loss. Gradient descent finds the best weight and bias by repeating the following process for a number of user-defined iterations.

Basically, It is an optimization algorithm that is used to minimize the loss/error iteratively.

The model begins training with randomized weights and biases near zero, and then repeats the following steps:

- Calculate the loss with the current weight and bias.
- Determine the direction to move the weights and bias that reduce loss.
- Move the weight and bias values a small amount in the direction that reduces loss.
- Return to step one and repeat the process until the model can't reduce the loss any further.

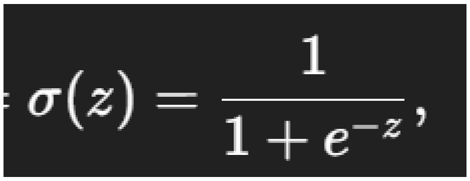
1.4 R2 Score:

The R-squared (R^2) or coefficient of determination is a statistical measure that represents the proportion of variance in the dependent variable that is explained by the independent variable(s) in a regression model. In simpler terms, it indicates how well the model fits the data or how much of variation in our output is explained by the model we have trained.

1.5 Logistic Regression and Classification

Logistic Regression is a classification algorithm used to predict the probability of a binary outcome (e.g., spam or not spam). It uses the **sigmoid function** to map predicted values between 0 and 1.

Z = b + w1x1+w2x2……wnxn (w = weights, b=bias, x=inputs/features)



$$\sigma(z) = \frac{1}{1+e^{-z}},$$

1.6 Classification model

A **model** whose prediction is a **class**. For example, the following are all classification models:

1. A model that predicts an input sentence's language (French? Spanish? Italian?).
2. A model that predicts tree species (Maple? Oak? Baobab?).
3. A model that predicts the positive or negative class for a particular medical condition.
4. In contrast, **regression models** predict numbers rather than classes.
5. Two common types of classification models are:

- **binary classification**
- **multi-class classification**

1.7 classification threshold

In a **binary classification,** a number between 0 and 1 that converts the raw output of a **logistic regression** model into a prediction of either the **positive class** or the negative class. Note that the classification threshold is a value that a human chooses, not a value chosen by model training.

1.8 Classification Metrics

1.8.1 Precision

Out of all the positive predictions, how many were actually correct?

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

1.8.2 Recall

Out of all actual positive cases, how many did we correctly identify?

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

1.8.3 Precision

The **harmonic mean** of Precision and Recall.

$$\textbf{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- Best when you want to **balance precision and recall**.

1.8.4 AUC-ROC (Area Under Curve – Receiver Operation Characteristics)

AUC measures the model's ability to distinguish between classes **across all thresholds**.

ROC curve plots:

- **True Positive Rate (Recall)** vs.
- **False Positive Rate (FPR = FP / (FP + TN))**

Interpretation:

- AUC = 1.0 → Perfect model
- AUC = 0.5 → Random guessing
- Higher AUC means better **class separability**

2. Project Work

2.1 Insurance Charges Prediction

Methodology: Predicted medical insurance charges based on features like sex, age, BMI, smoker status, children, and region.

Preprocessing: Checked linear relationship between inputs and output using scatter plots and other plots like hist/dist plots for analyzing distribution. Box Plots for detecting outliers. Encoded categorical variables (e.g., sex, smoker, region) using label/one-hot encoding.

Implementation: Trained a linear regression model using scikit-learn and PyTorch.

Results: Model performed well with metrics like MSE, MAE, and R^2 score showing good fit on the training and test sets.

Mean Squared Error: 30833038.61799048

Mean Absolute Error: 4035.549736789495

R2 Score: 0.7975077281232326

2.2 Spam-Ham Classification

Methodology: Classified messages as spam or ham using logistic regression.

Challenge: The dataset was imbalanced — ham messages were in the majority.

Preprocessing: Applied techniques to handle imbalance (e.g., class weighting or resampling), and converted text data to numerical format (e.g., using TF-IDF).

Implementation: Trained a logistic regression model using scikit-learn.

Results: Successfully built a classifier with good precision and recall for the minority (spam) class, and evaluated it using metrics like accuracy, precision, recall, F1-score, and AUC.

Validation Metrics:

Accuracy: 0.9797

Precision: 0.9220

Recall: 0.9286

AUC: 0.9930

---

3. Challenges and Approaches

Task 1: Insurance Charges Prediction

- **Challenge**: Presence of **outliers** and **skewed distributions** in numerical features like BMI and charges.
  Approach: Used scatter plots and boxplots to identify outliers and applied scaling to normalize features.

- **Challenge**: Handling **categorical variables** such as sex, smoker, and region.
  Approach: Used **label encoding** for binary categories and **one-hot encoding** for multi-class features like region.

- **Challenge**: Target variable (charges) had a wide range, making training unstable.
  Approach: **Standardized the target** values during training and reversed the transformation after prediction.

Task 2: Spam-Ham Classification

- **Challenge**: **Imbalanced dataset** — ham (non-spam) messages were the majority, which could bias the model.
  Approach: Handled imbalance by using **class weights** or **resampling techniques** to ensure the model learns to detect spam effectively.
- **Challenge**: Converting text data into numerical format.
  Approach: Used **TF-IDF vectorization** to transform text messages into feature vectors suitable for logistic regression.
- **Challenge**: Ensuring proper evaluation beyond accuracy due to imbalance.
  Approach: Evaluated using **precision, recall, F1-score, and AUC** to assess model performance on both classes.

4. Results and Analysis

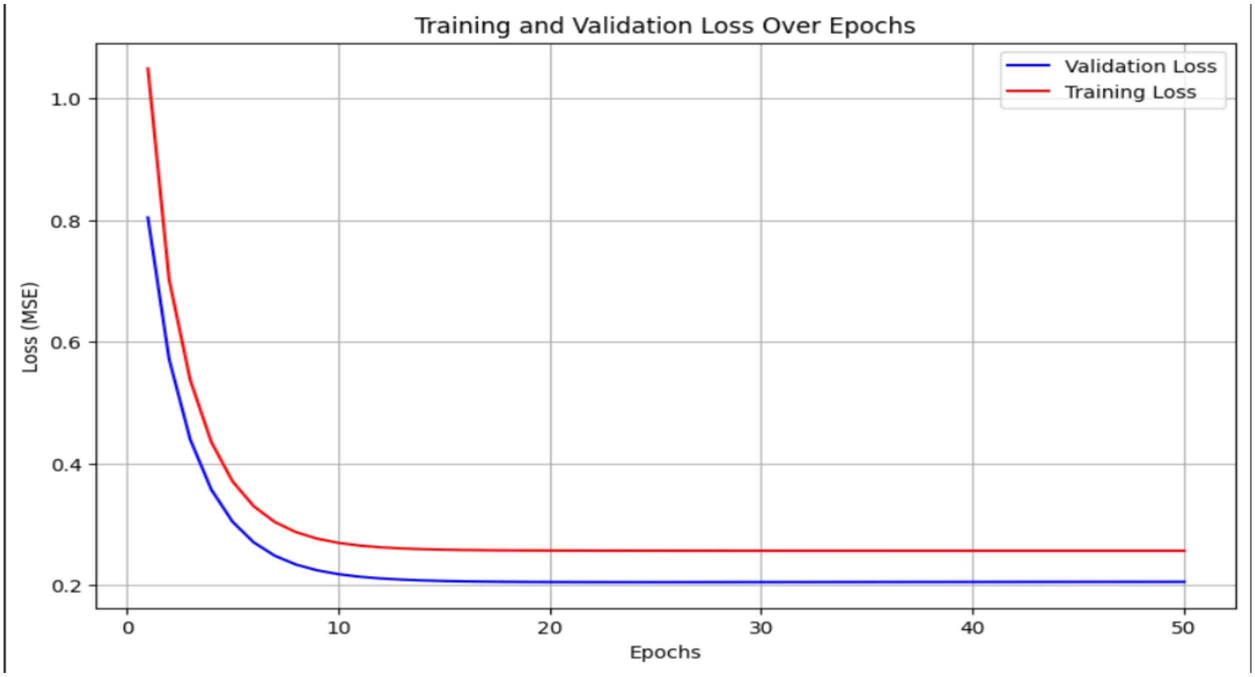
Task 01: Insurance Price Prediction

Evaluation Metrics

Mean Squared Error: 30664742.0000

Mean Absolute Error: 4039.1997

R² Score: 0.7986



Task 02: Spam-Ham Classification

Evaluation Metrics

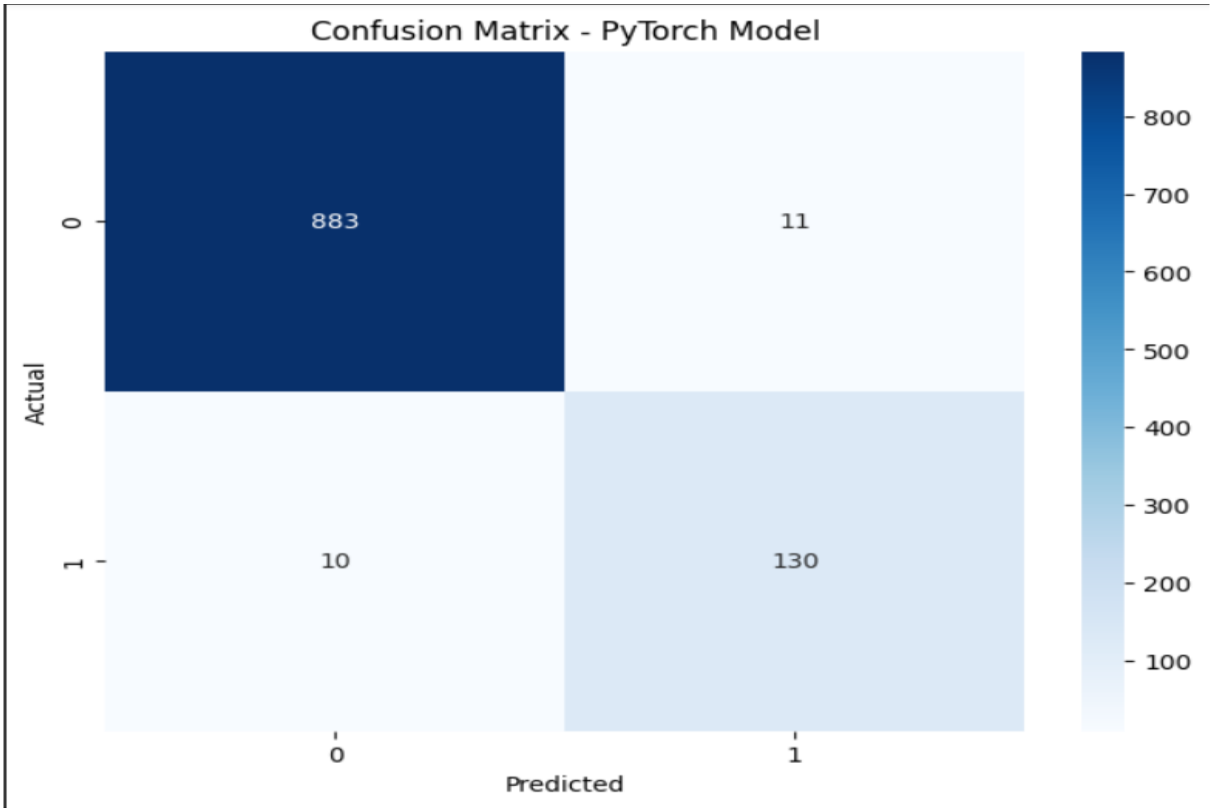
Validation Metrics:

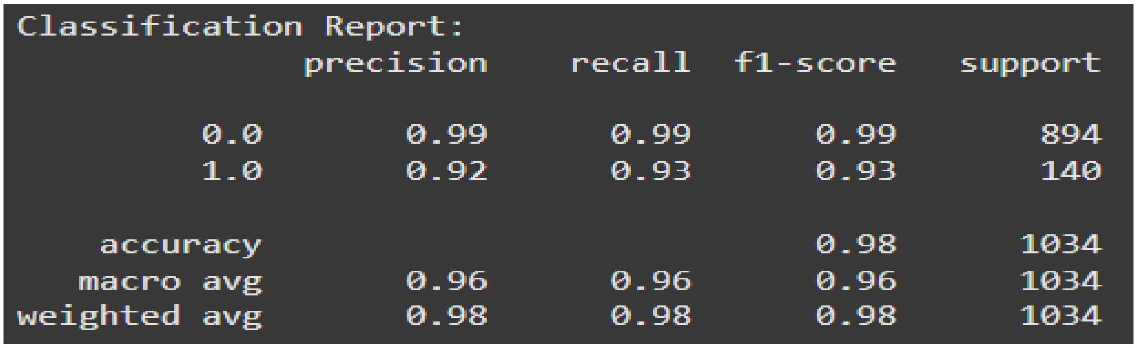
Accuracy: 0.9797

Precision: 0.9220

Recall: 0.9286

AUC: 0.9930





```
Classification Report:
              precision    recall  f1-score   support

         0.0       0.99      0.99      0.99       894
         1.0       0.92      0.93      0.93       140

    accuracy                           0.98      1034
   macro avg       0.96      0.96      0.96      1034
weighted avg       0.98      0.98      0.98      1034
```

5. Conclusion

Both tasks demonstrated effective use of regression and classification techniques. In Task 1, we accurately predicted insurance charges using proper preprocessing and linear modeling. In Task 2, we handled class imbalance and built a reliable spam-ham classifier using logistic regression.

Key Takeaways:

- Preprocessing and evaluation metrics are critical to model success.
- Handling imbalanced data improves fairness and accuracy.

Improvements:

- Try advanced models (e.g., XGBoost, LSTM).
- Use regularization and hyperparameter tuning for better performance.

6. References:

- https://pytorch.org/
- https://scikit-learn.org/stable/
- https://developers.google.com/machine-learning/crash-course/
- https://www.youtube.com/