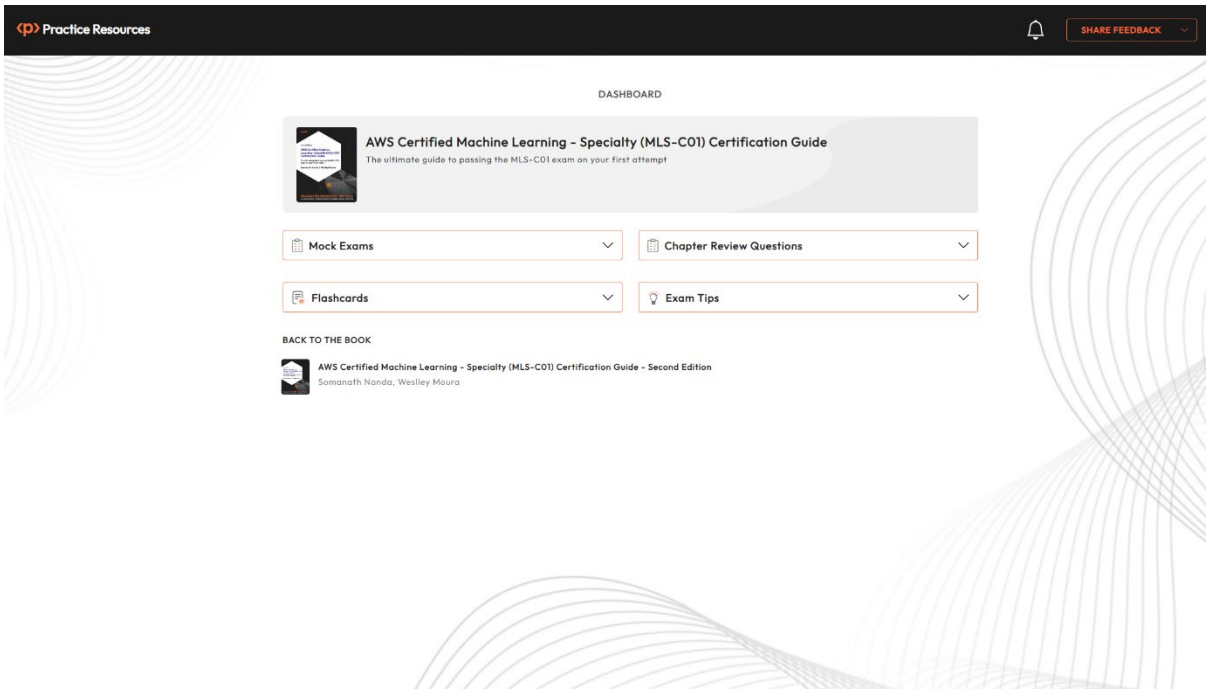
AWS Certified Machine Learning - Specialty (MLS-C01) Certification Guide, Second Edition

Preface:



Chapter 1: Machine Learning Fundamentals

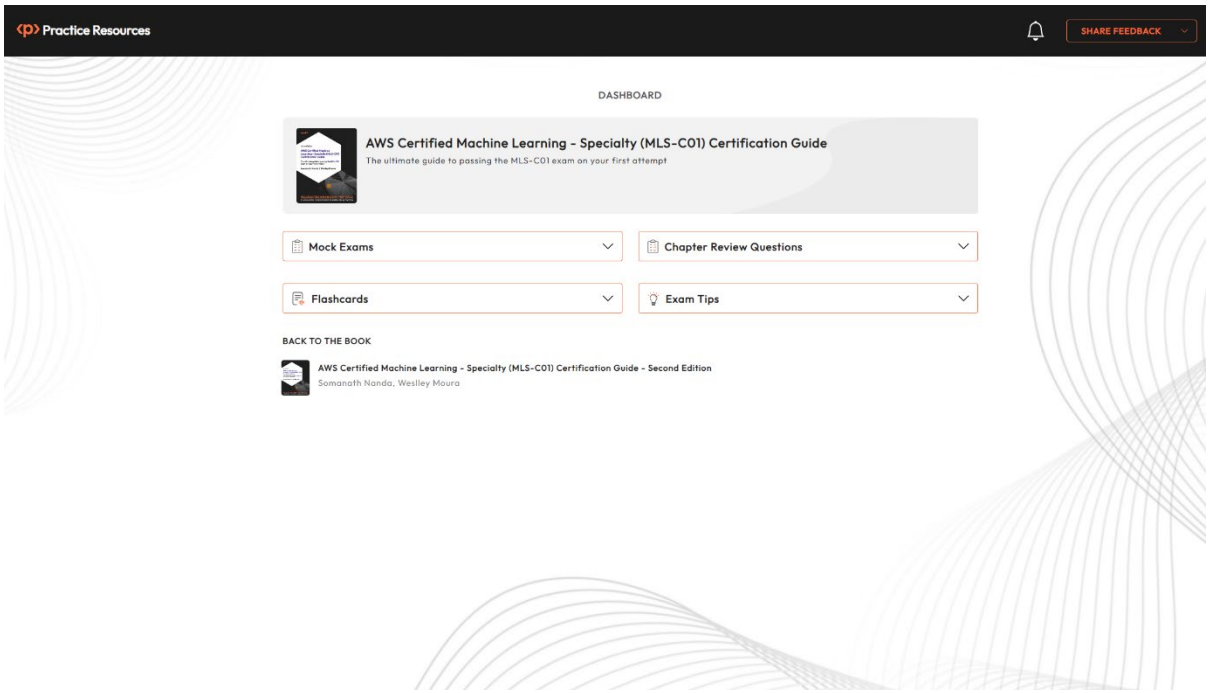


Figure 1.1 – Dashboard interface of the online practice resources

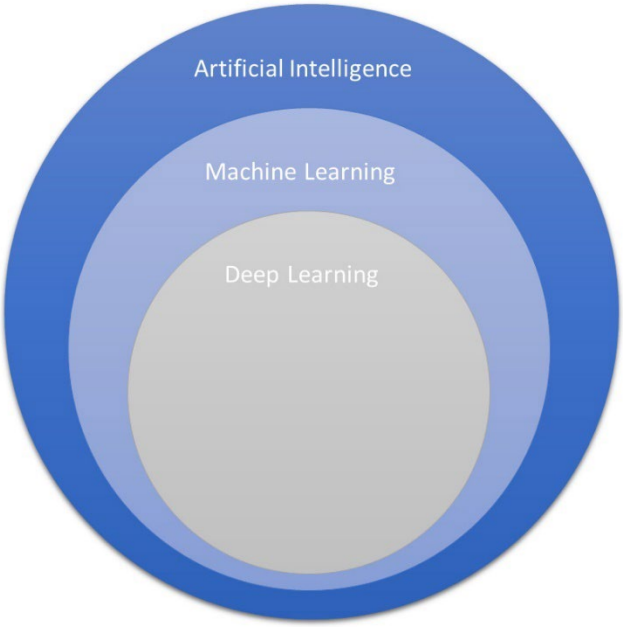


Figure 1.2 – Hierarchy of AI, ML, and DL

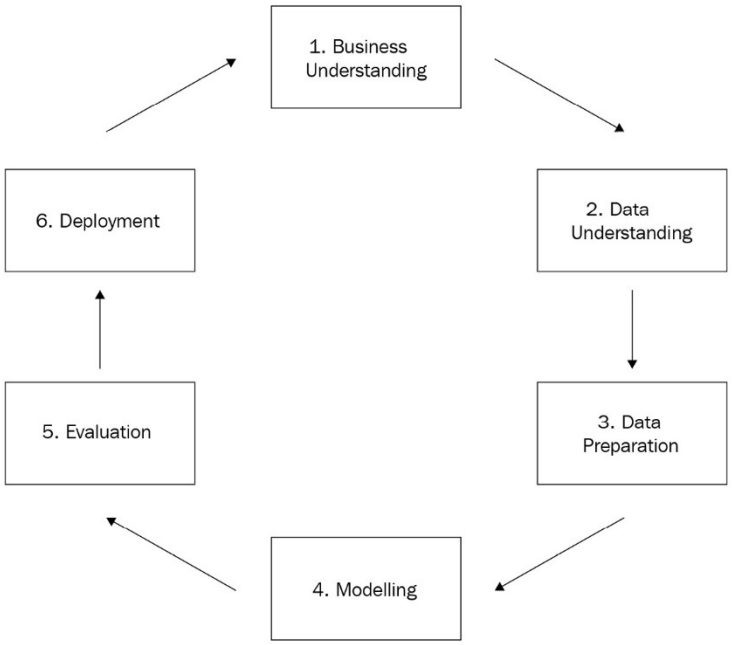


Figure 1.3 – CRISP-DM methodology

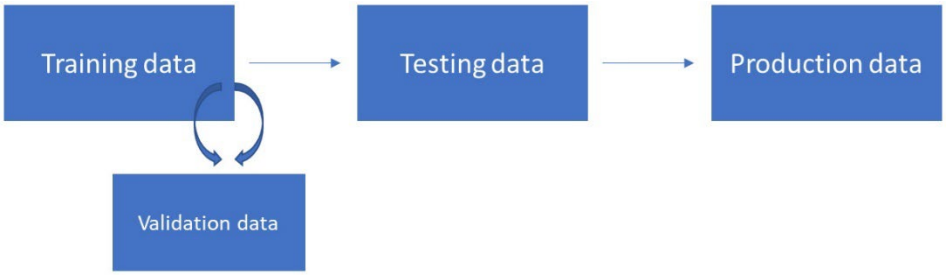


Figure 1.4 – Data splitting

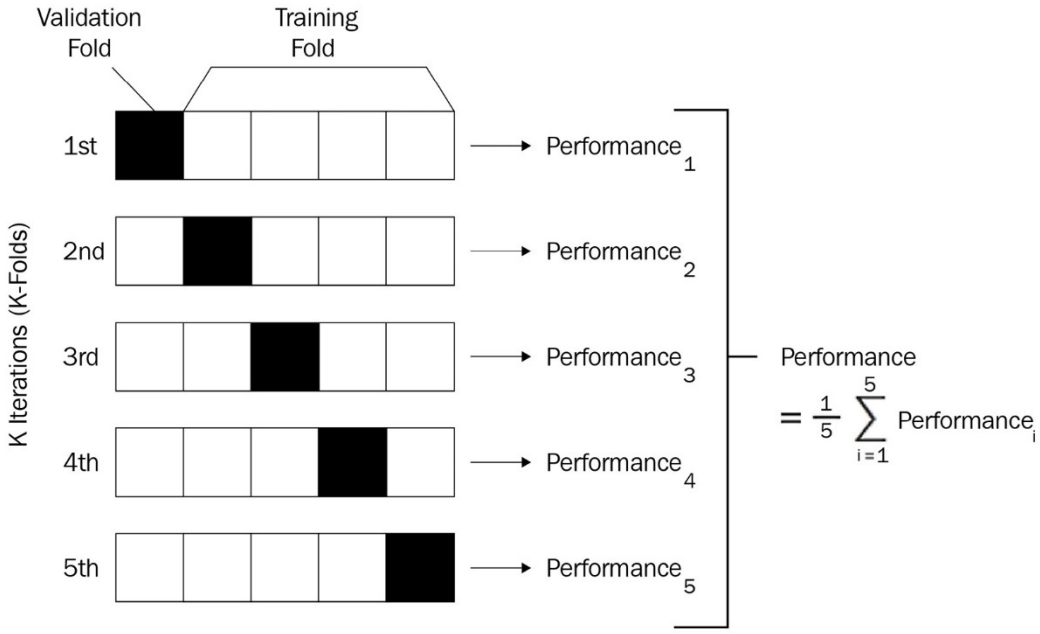


Figure 1.5 – Cross-validation in action

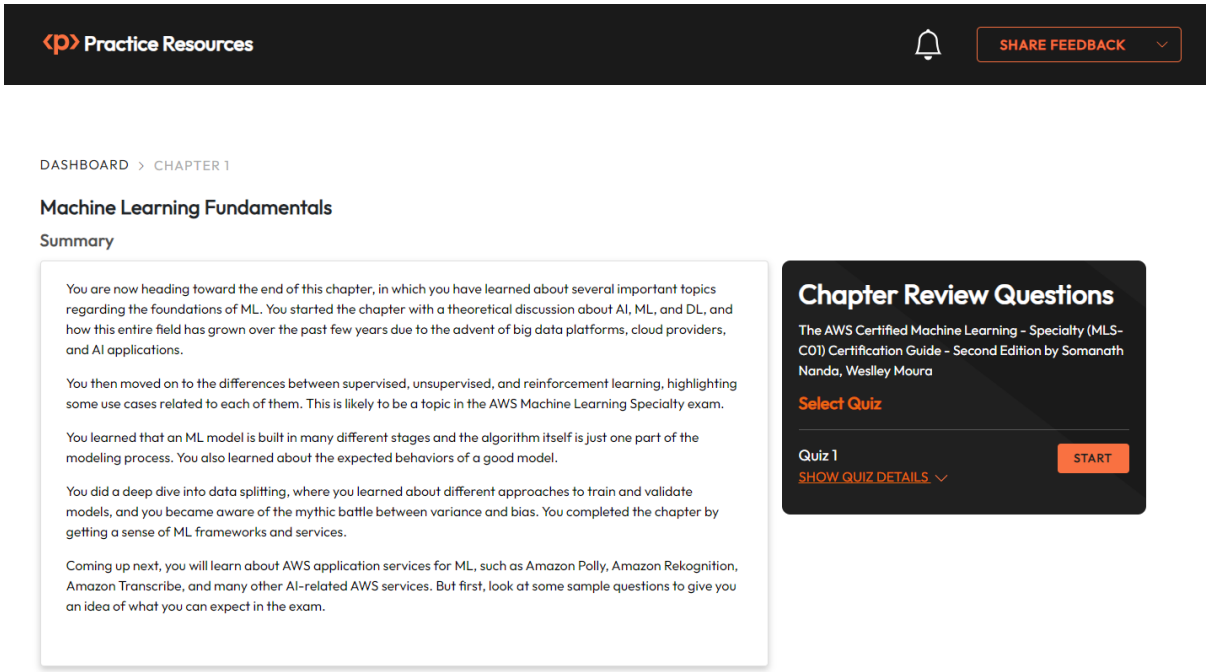


Figure 1.7 – Chapter Review Questions for Chapter 1

MOST EXPENSIVE PER GB STORED

LEAST EXPENSIVE PER GB STORED

STORAGE CLASS	STANDARD	STANDARD-IA	ONE ZONE-IA	AMAZON GLACIER
DURABILITY	99·999999999%	99·999999999%	99·999999999%	99·999999999%
AVAILABILITY	99·99%	99·9%	99·5%	99·99%

LEAST EXPENSIVE PER REQUEST

MORE EXPENSIVE PER REQUEST

Figure 2.1 – A comparison table of S3 Storage classes

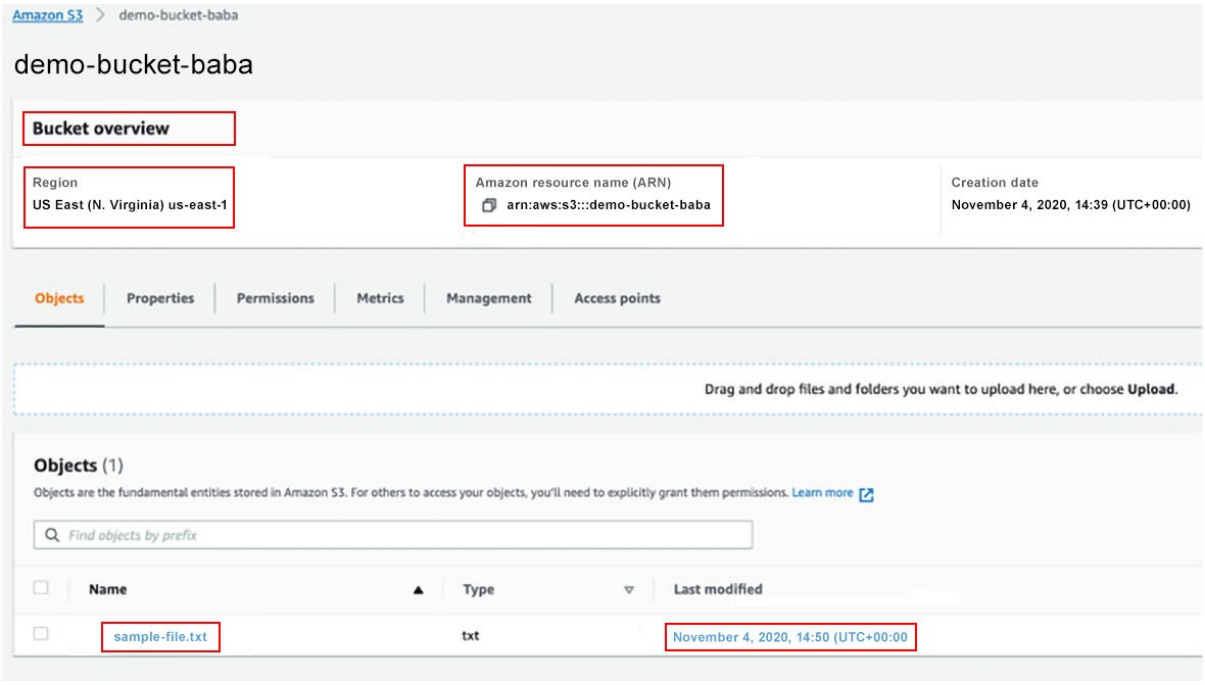


Figure 2.2 – AWS S3 listing your files

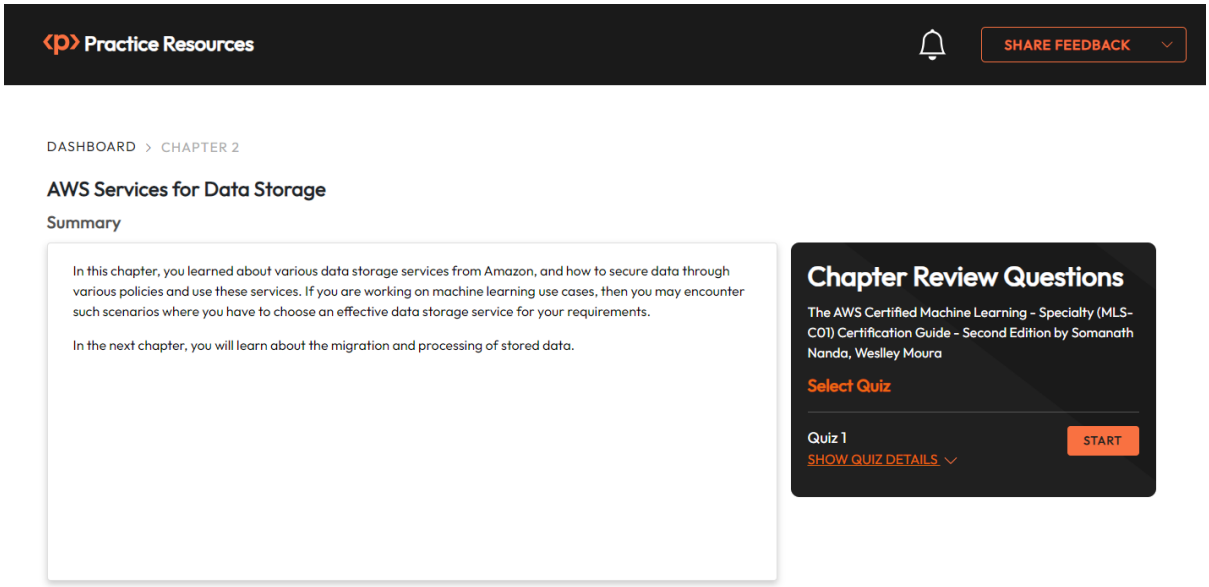


Figure 2.4 – Chapter Review Questions for Chapter 2

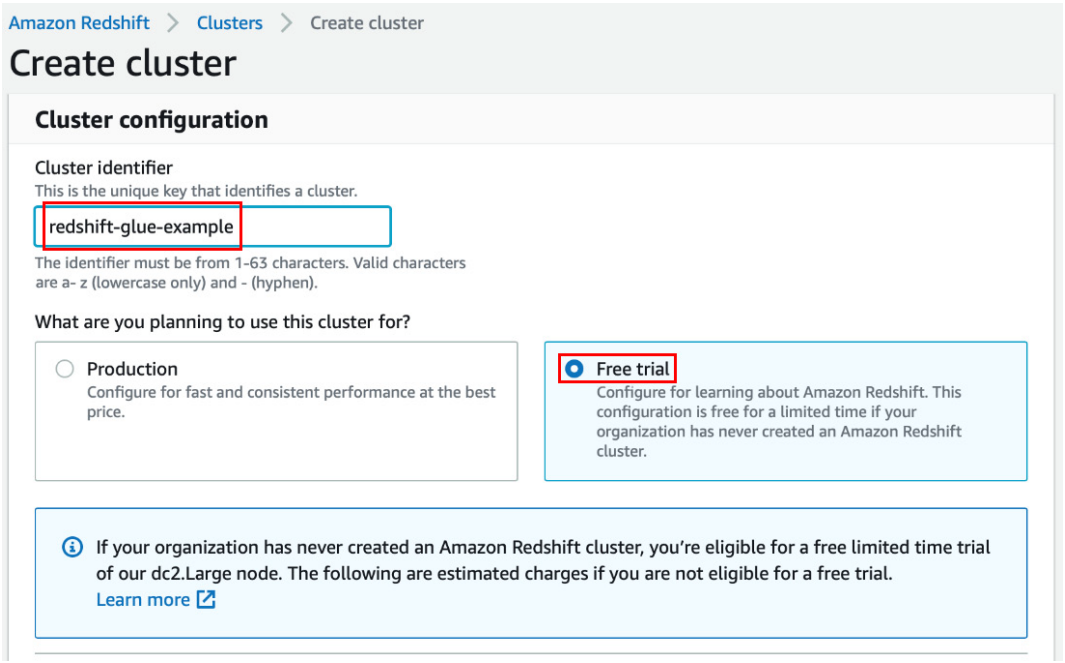


Figure 3.1 – A screenshot of Amazon Redshift's Create cluster screen

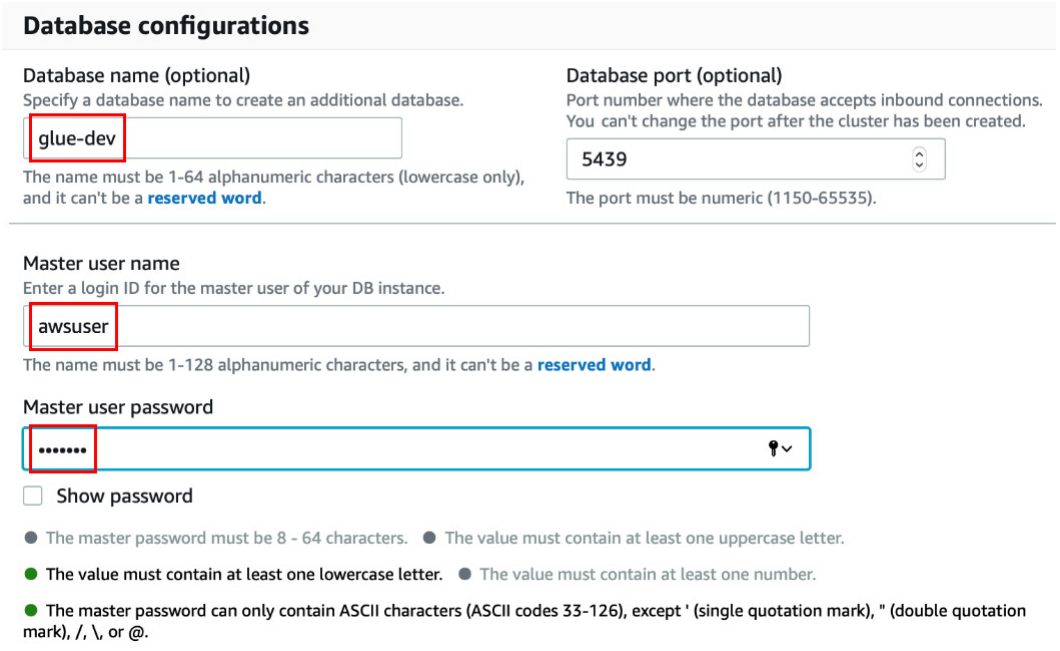


Figure 3.2 – A screenshot of an Amazon Redshift cluster's Database configurations section

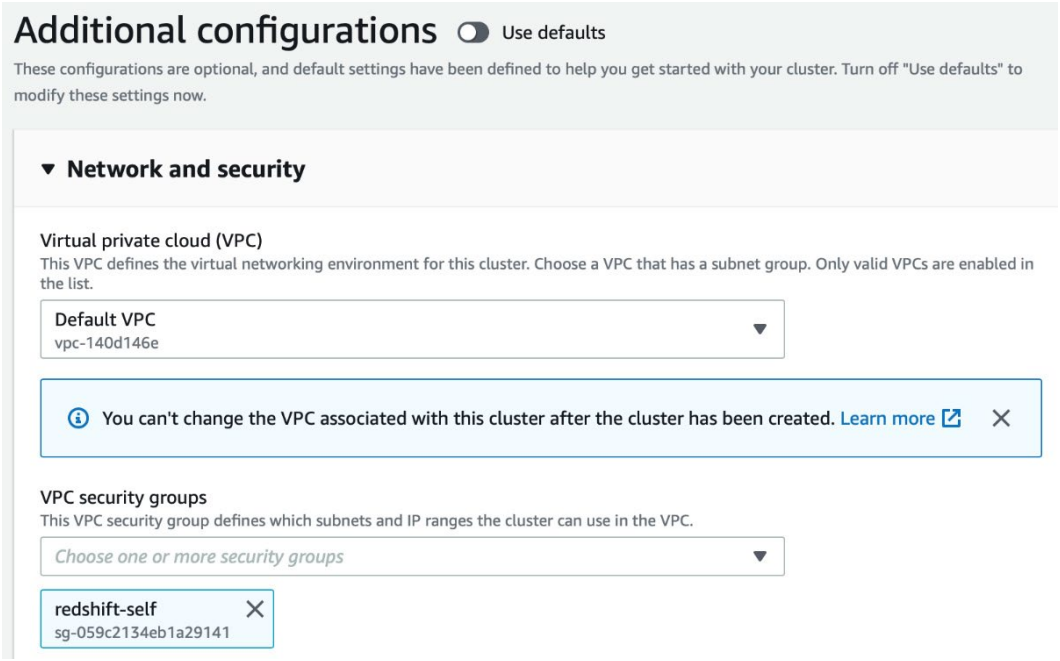


Figure 3.3 – A screenshot of an Amazon Redshift cluster's Additional configurations section

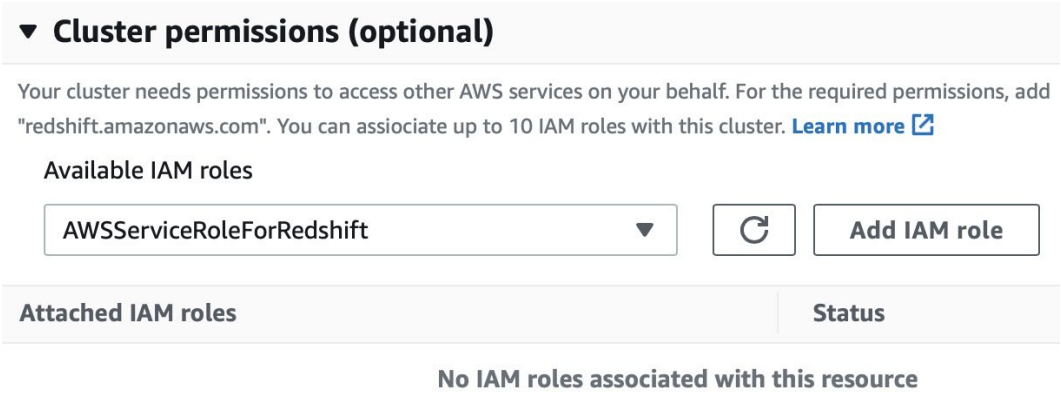


Figure 3.4 – A screenshot of an Amazon Redshift cluster's Cluster permissions section

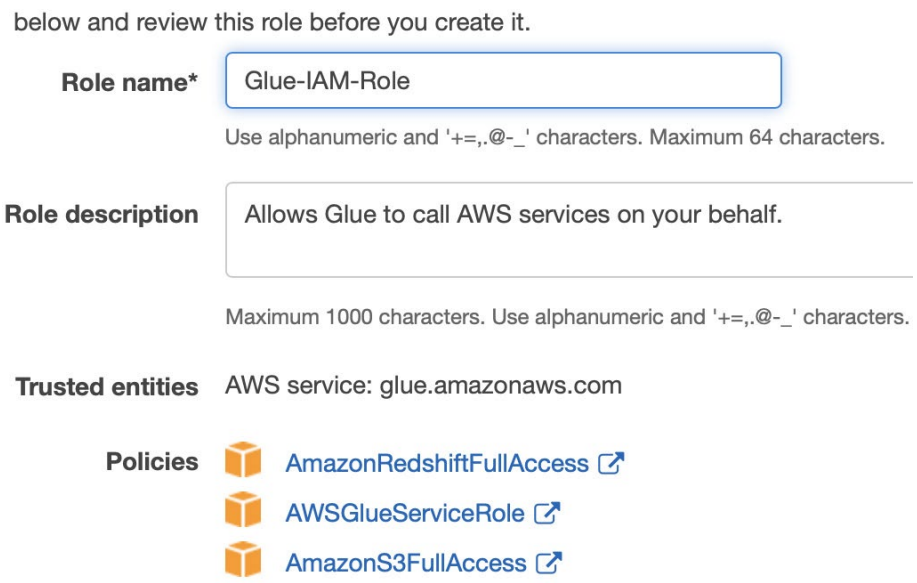


Figure 3.5 – A screenshot of the IAM role

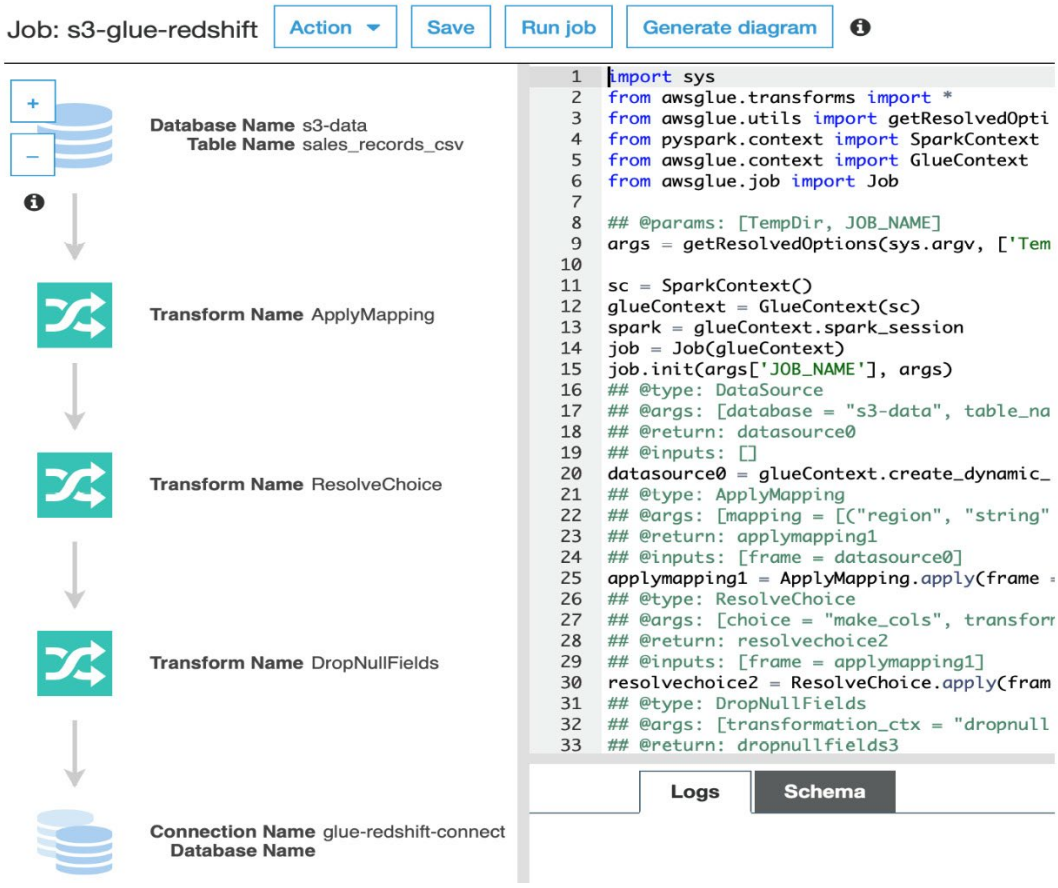


Figure 3.6 – A screenshot of the AWS Glue ETL job

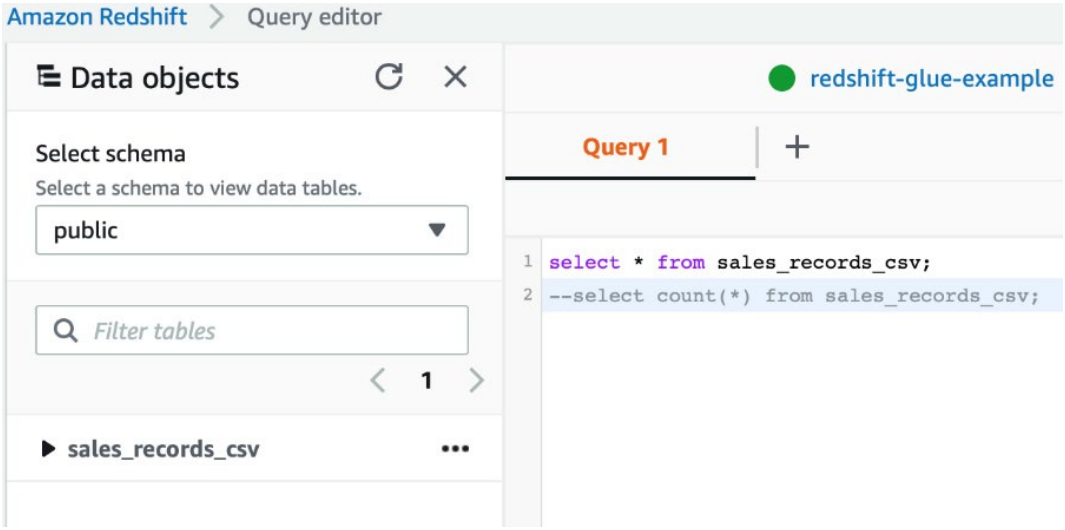


Figure 3.7 – A screenshot of Amazon Redshift's Query editor

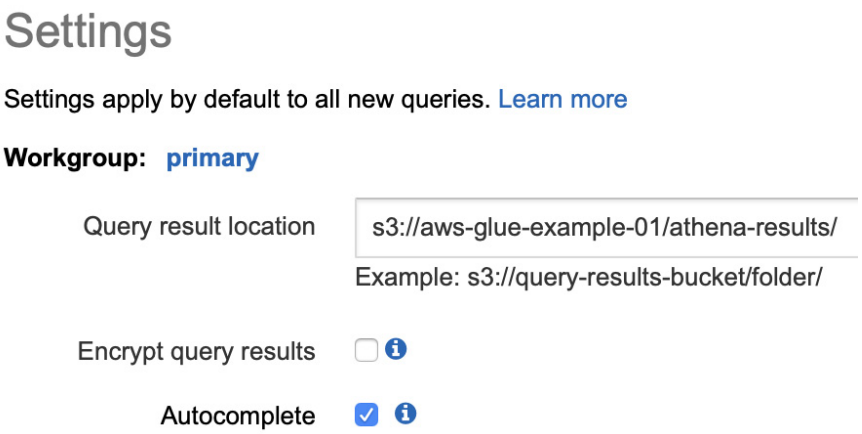


Figure 3.8 – A screenshot of Amazon Athena's settings

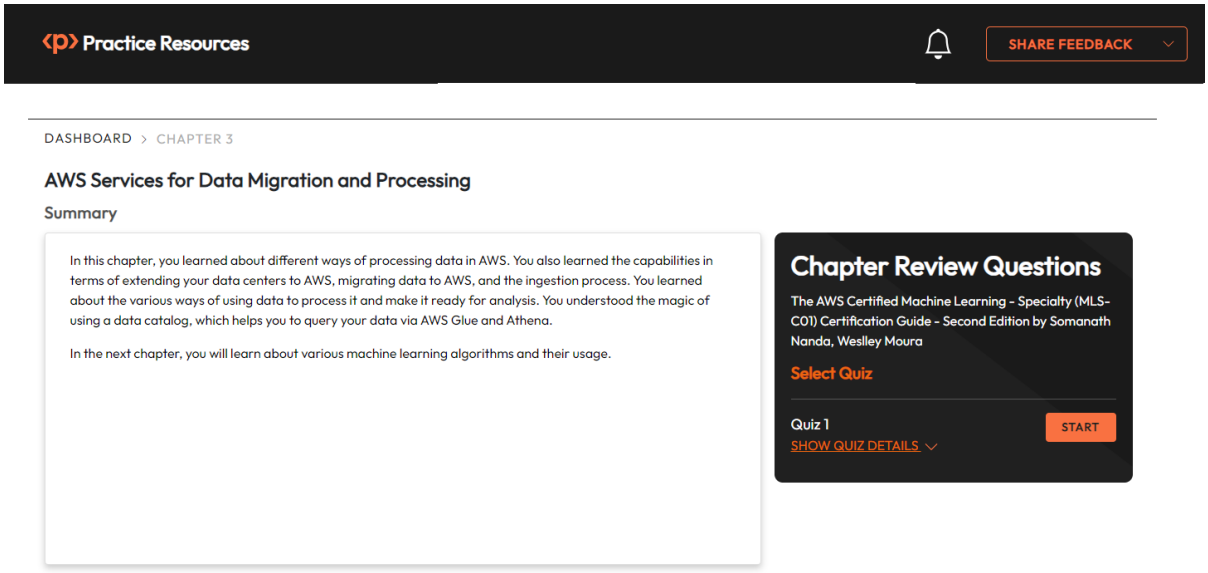


Figure 3.10 – Chapter Review Questions for Chapter 3

Chapter 4: Data Preparation and Transformation

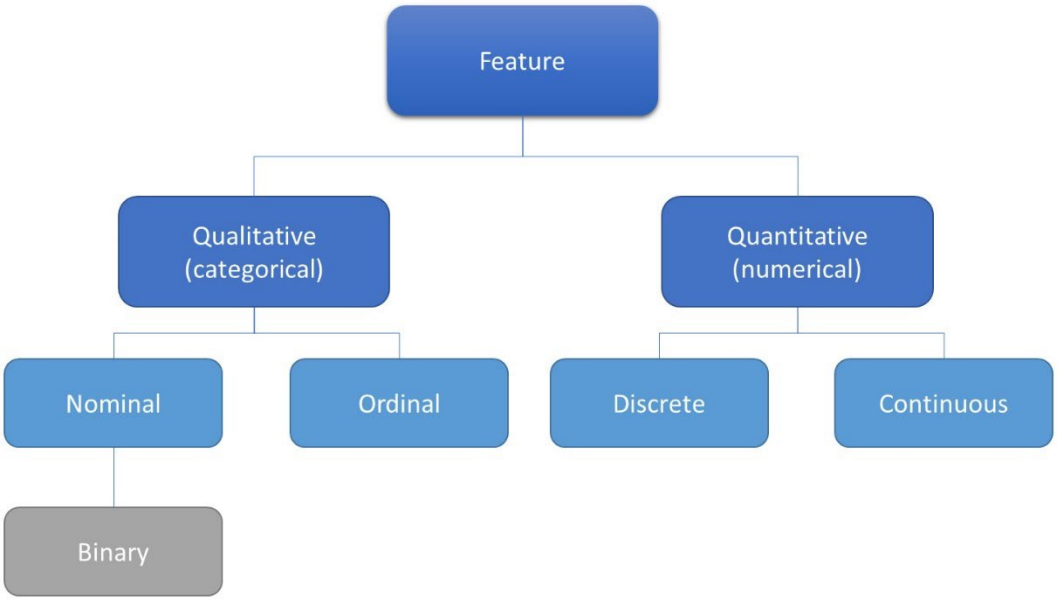


Figure 4.1 – Feature types

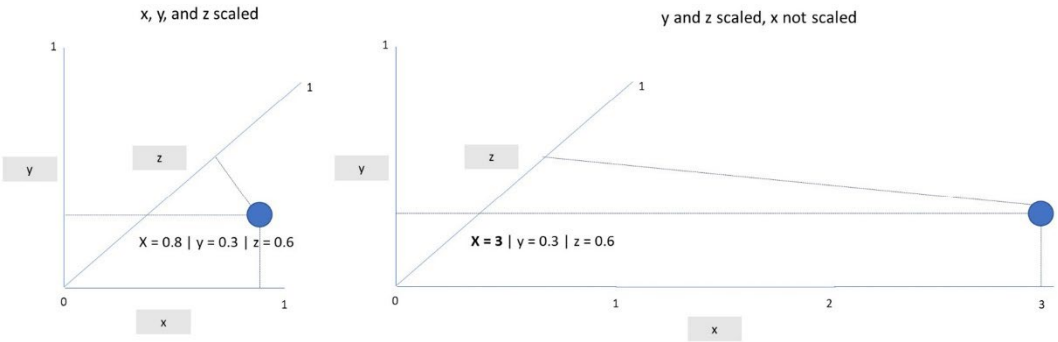


Figure 4.2 – Plotting data of different scales in a hyper plan

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Figure 4.3 – Normalization formula

$$X' = \frac{X - \mu}{\sigma}$$

Figure 4.4 – Standardization formula

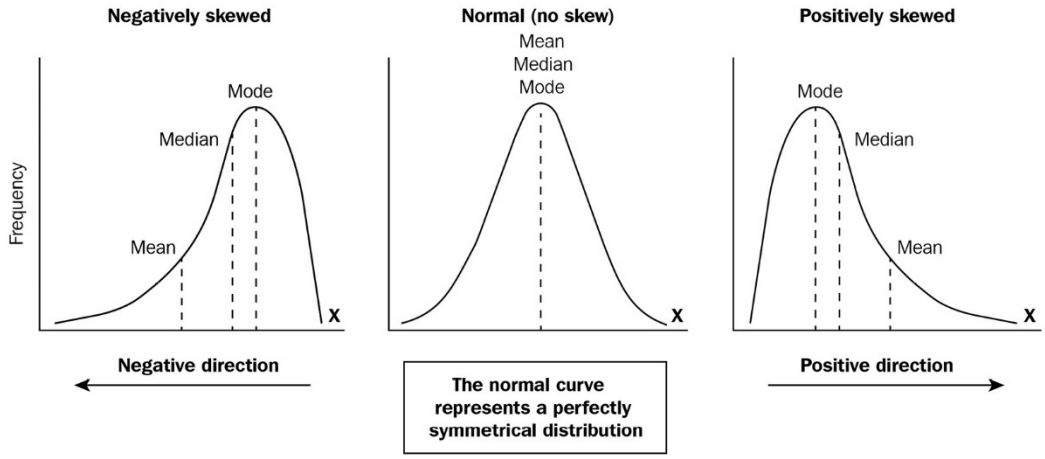


Figure 4.5 – Skewed distributions

$$\log(x^n) = n\log(x)$$

Figure 4.6 – Logarithmic properties

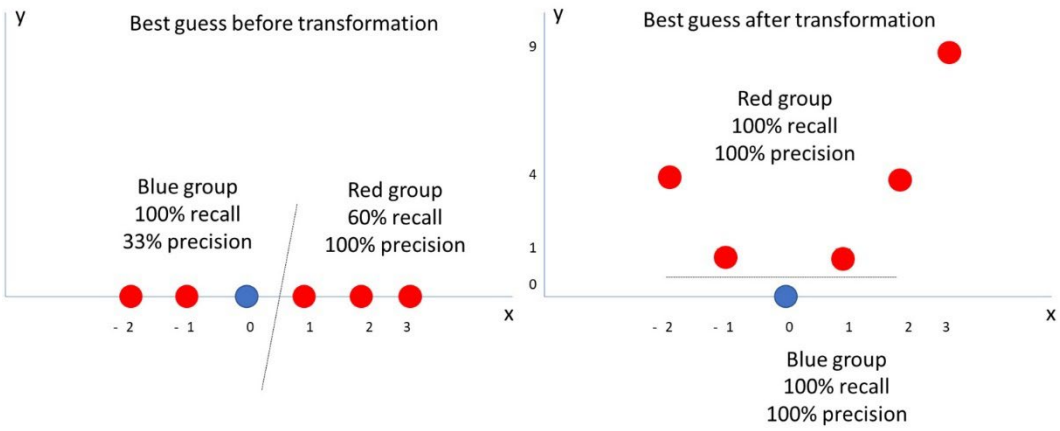


Figure 4.7 – Exponential transformation in action

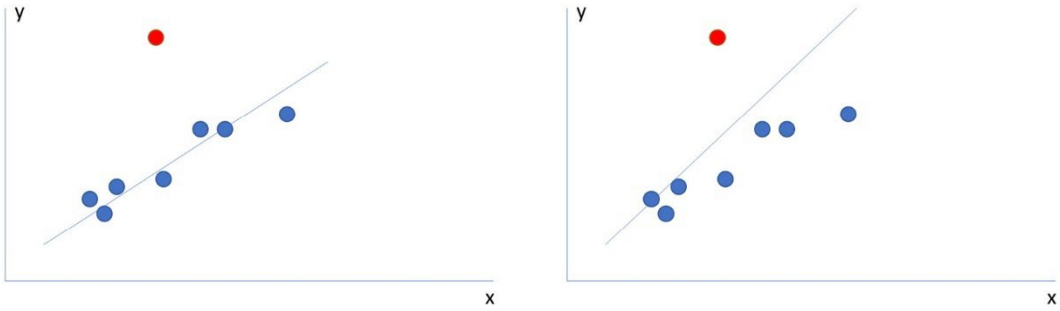


Figure 4.8 – Identifying an outlier

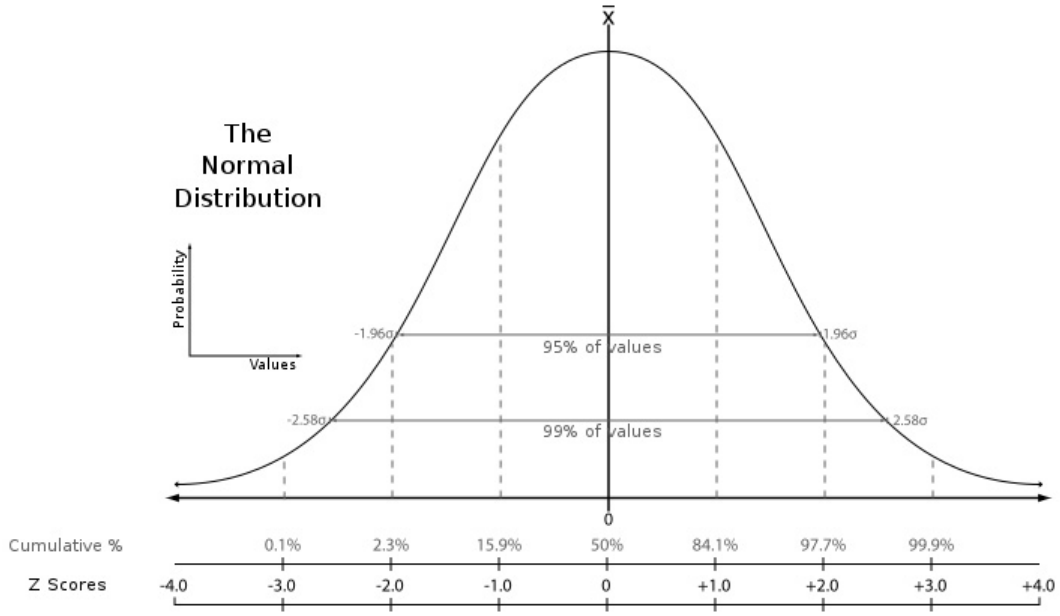


Figure 4.9 – Normal distribution properties. Image adapted from `https://pt.wikipedia.org/wiki/Ficheiro:The_Normal_Distribution.svg`

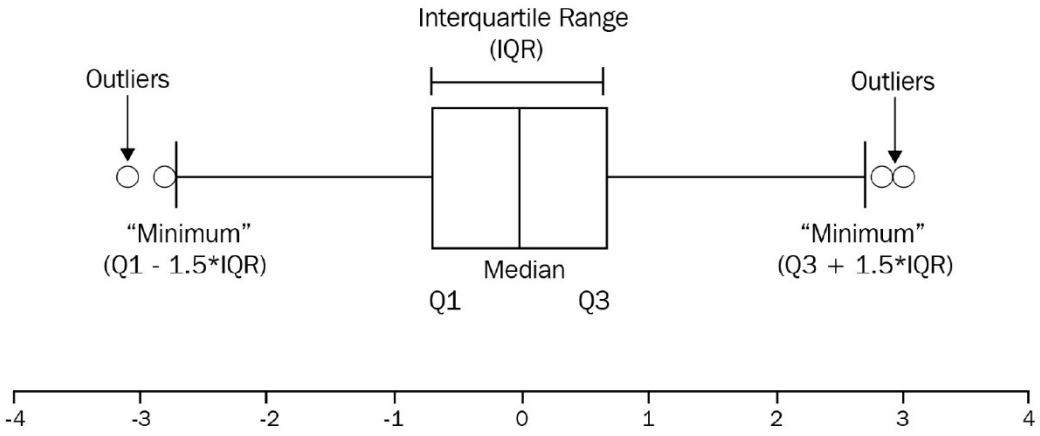
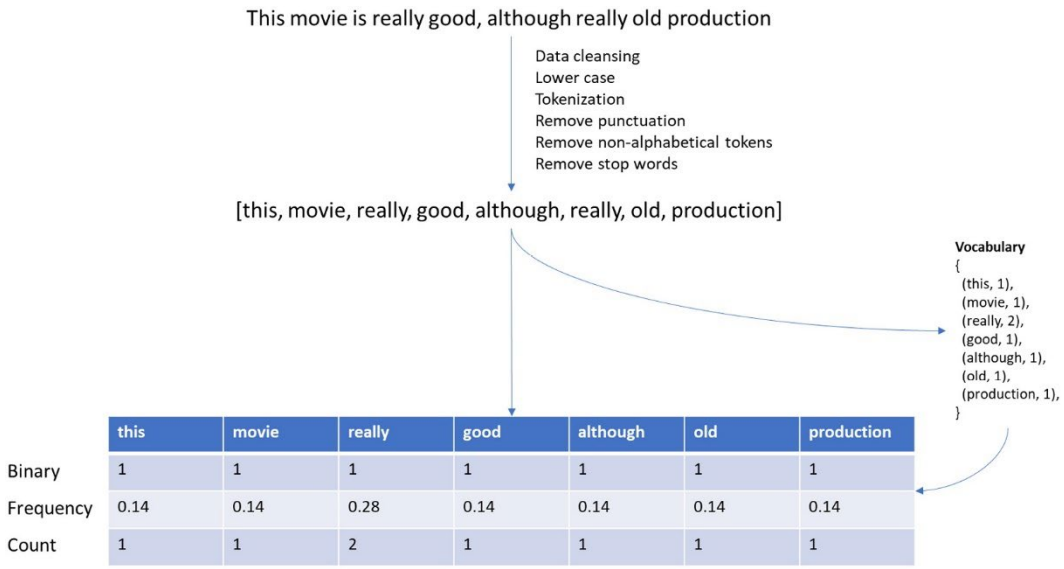


Figure 4.10 – Box plot definition



	this	movie	really	good	although	old	production
Binary	1	1	1	1	1	1	1
Frequency	0.14	0.14	0.28	0.14	0.14	0.14	0.14
Count	1	1	2	1	1	1	1

Figure 4.11 – BoW in action

	Dim 1	Dim 2	Dim 3	Dim 4	Dim 5
King	0.22	0.76	0.77	0.44	0.33
Queen	0.98	0.09	0.67	0.89	0.56
Live	0.13	0.99	0.88	0.01	0.55
Castle	0.01	0.89	0.34	0.02	0.90

	PC 1	PC2
King	0.9	0.3
Queen	0.8	0.5
Live	0.33	0.7
Castle	0.28	0.7

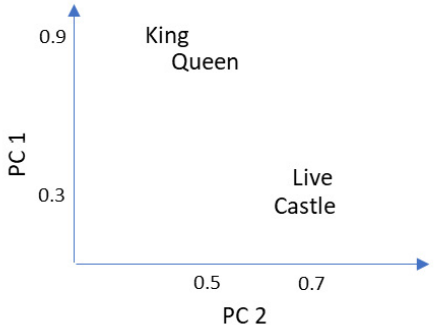


Figure 4.12 – Plotting words

	I	Will	Pass	This	Exam	You	See	It
I	0	3	...	...	...	...	...	...
Will	3	0	...	...	...	...	...	...
Pass	...	...	0	...	...	...	...	...
This	...	...	...	0	...	...	...	...
Exam	...	...	...	...	0	...	...	...
You	...	...	...	...	...	0	...	...
See	...	...	...	...	...	...	0	...
It	...	...	...	...	...	...	...	0

Figure 4.13 – Co-occurrence matrix

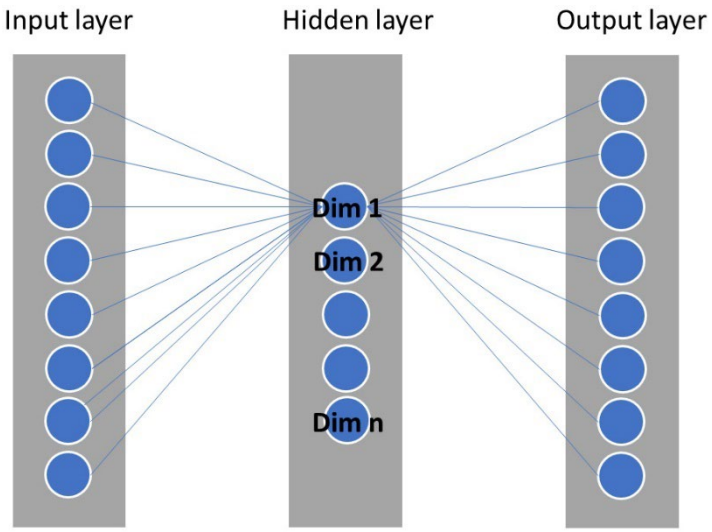


Figure 4.14 – Building embedding spaces with neural networks

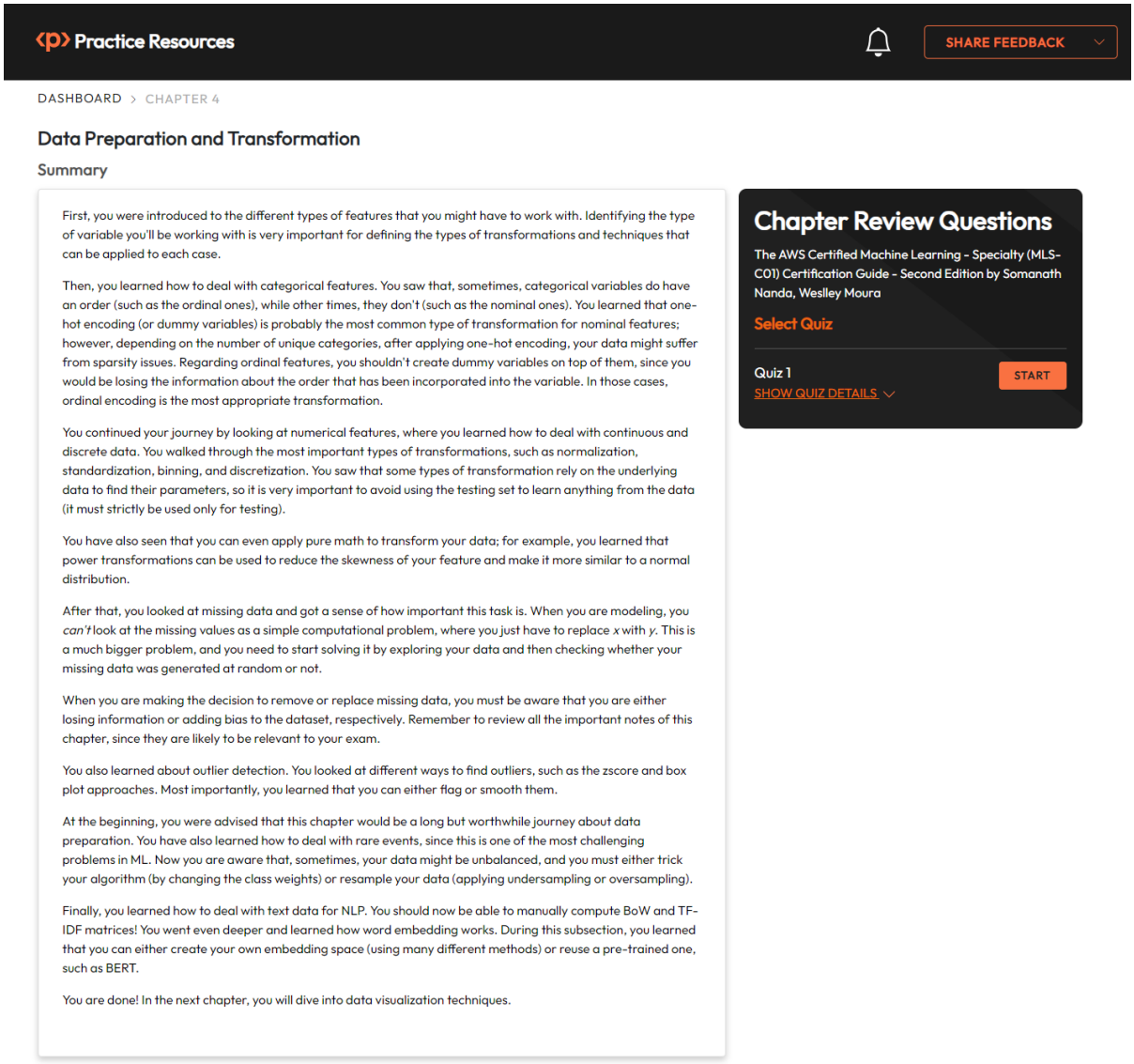


Figure 4.16 – Chapter Review Questions for Chapter 4

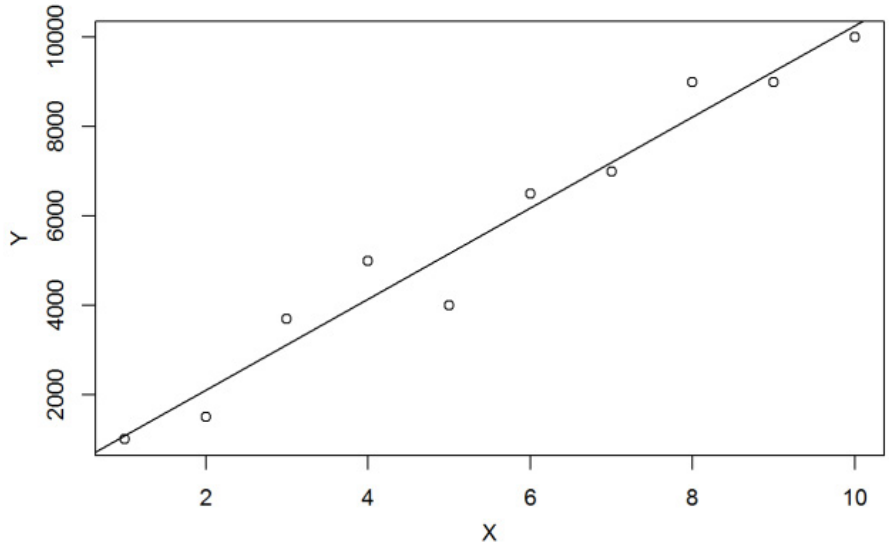


Figure 5.1 – Plotting relationships with a scatter plot

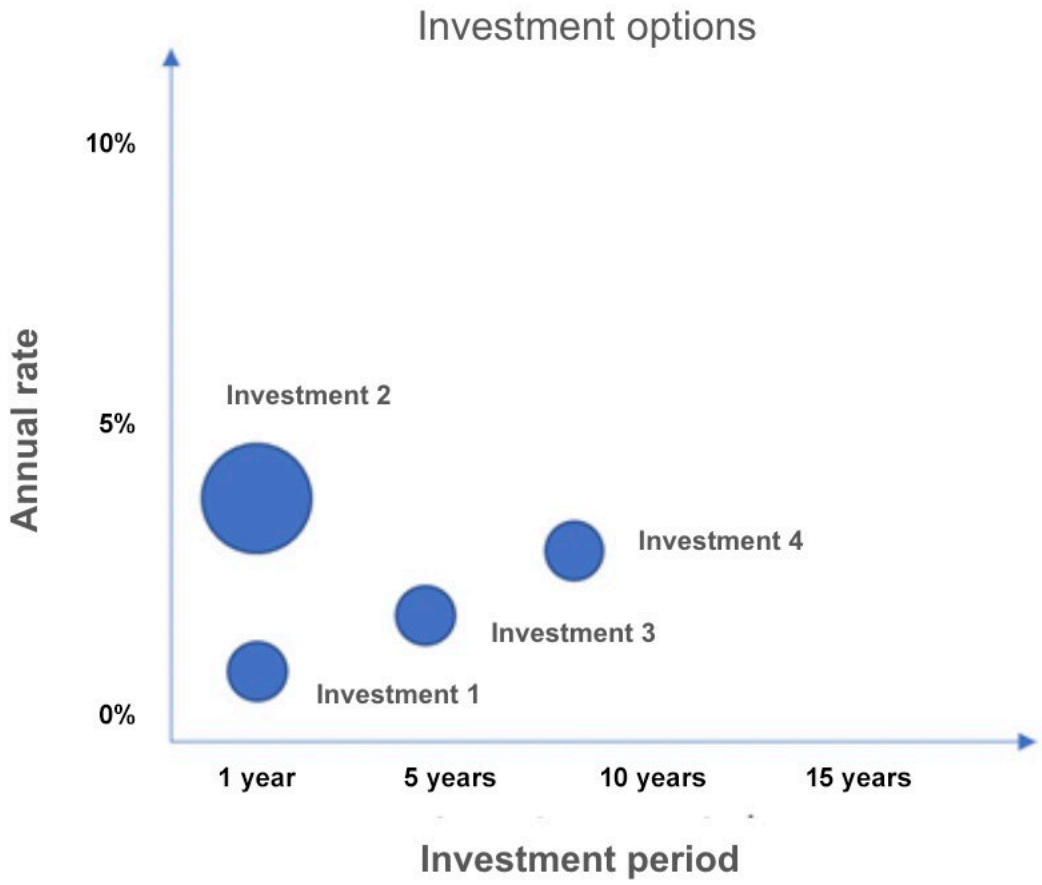


Figure 5.2 – Plotting relationships with a bubble chart

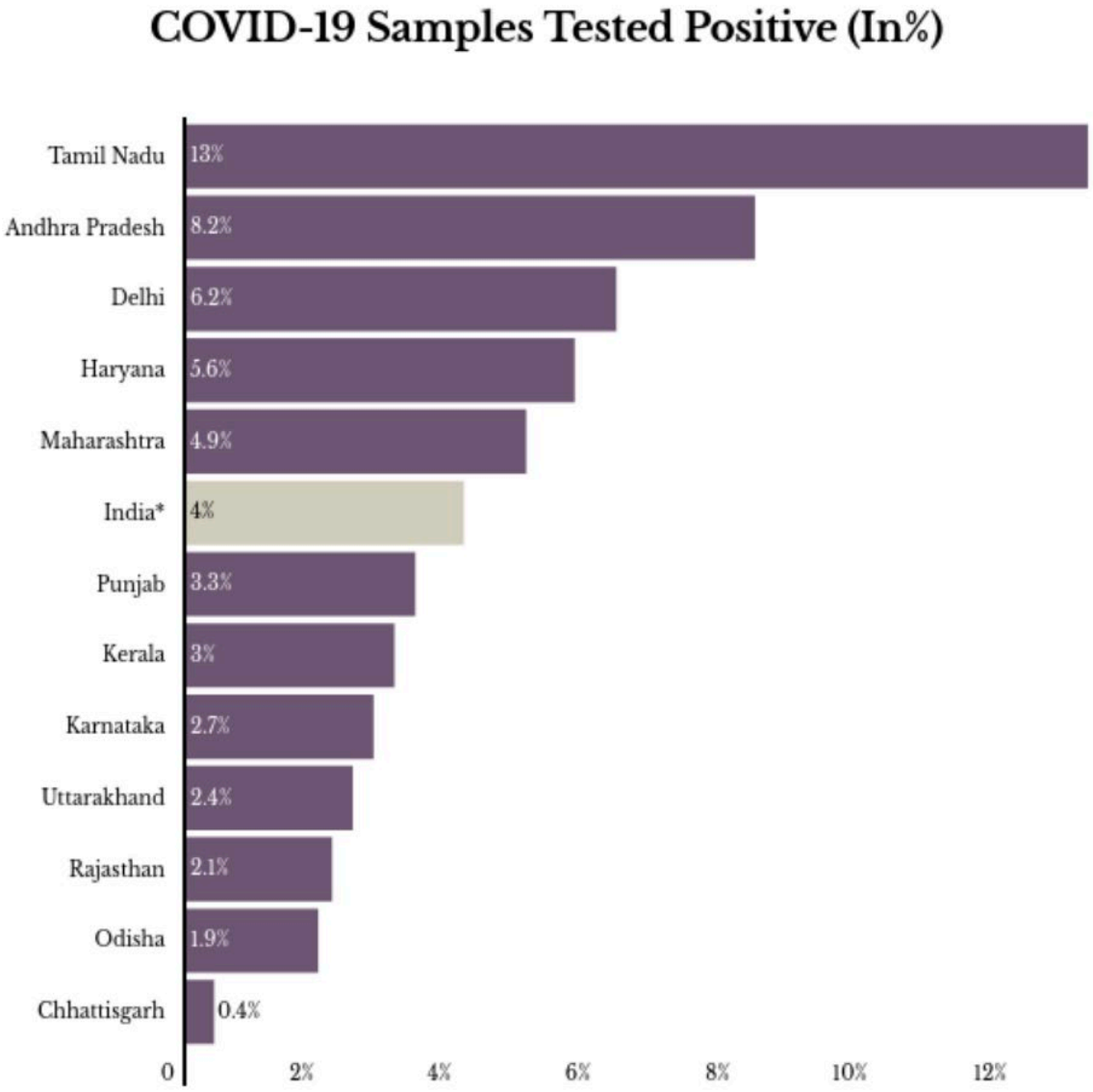


Figure 5.3 – Plotting comparisons with a bar chart (source: State Health Department of India)

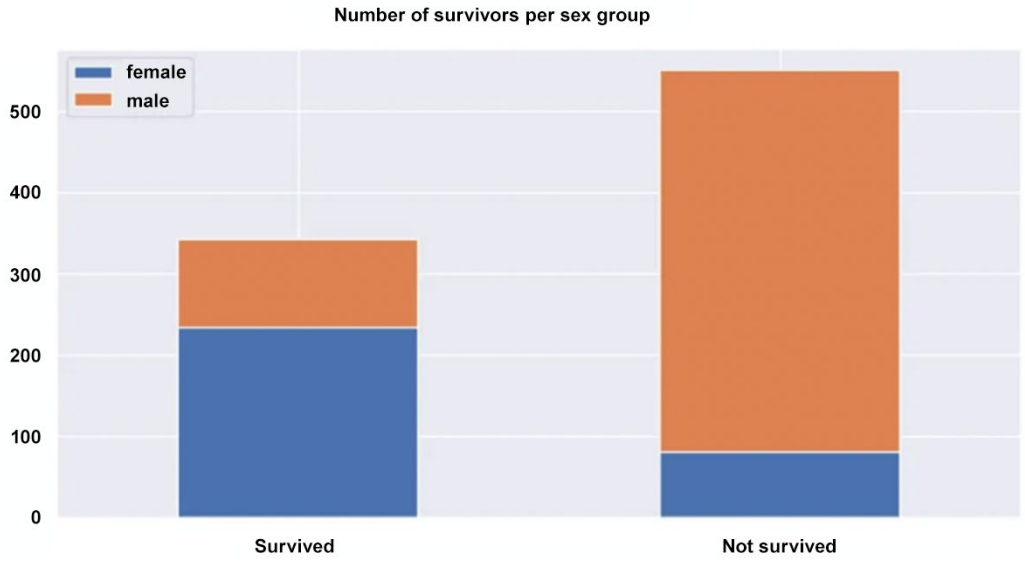


Figure 5.4 – Using a stacked bar chart to analyze the Titanic disaster dataset

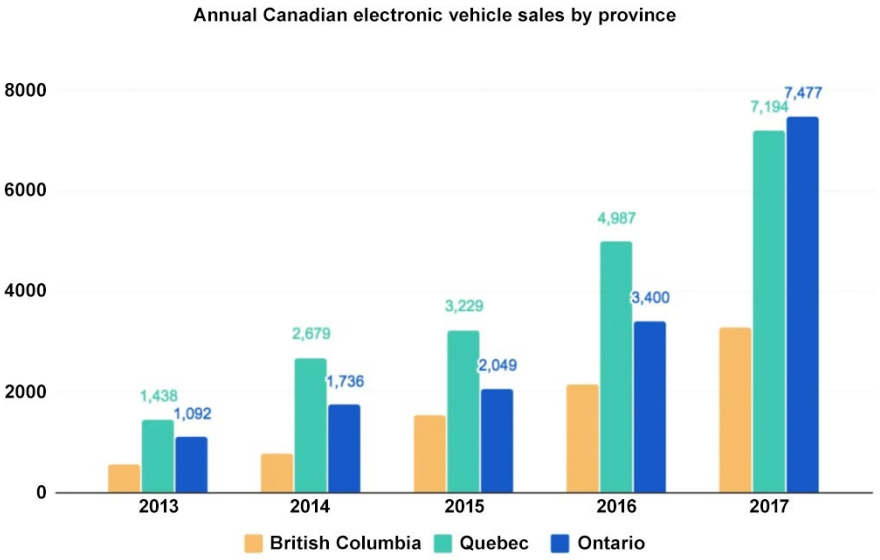


Figure 5.5 – Plotting comparisons with a column chart (source: https://electrek.co/)

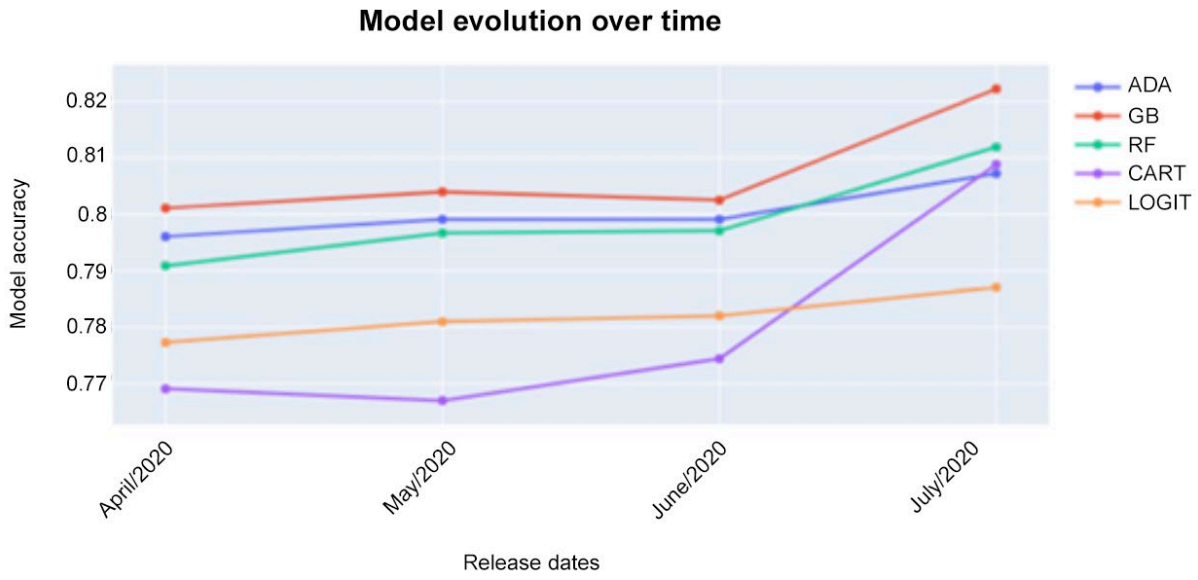


Figure 5.6 – Plotting comparisons with a line chart

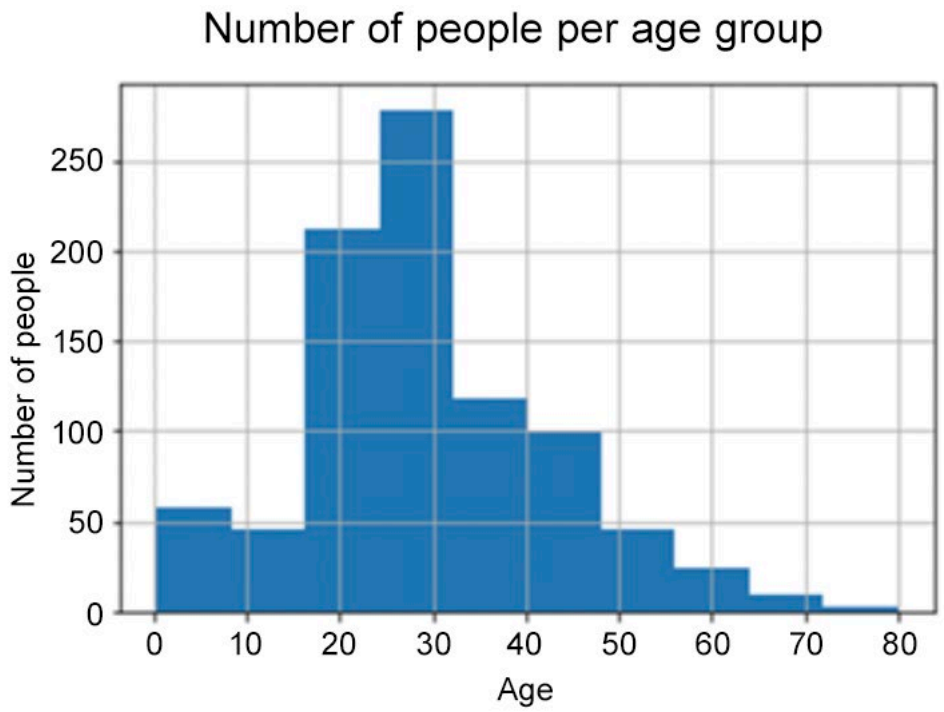


Figure 5.7 – Plotting distributions with a histogram

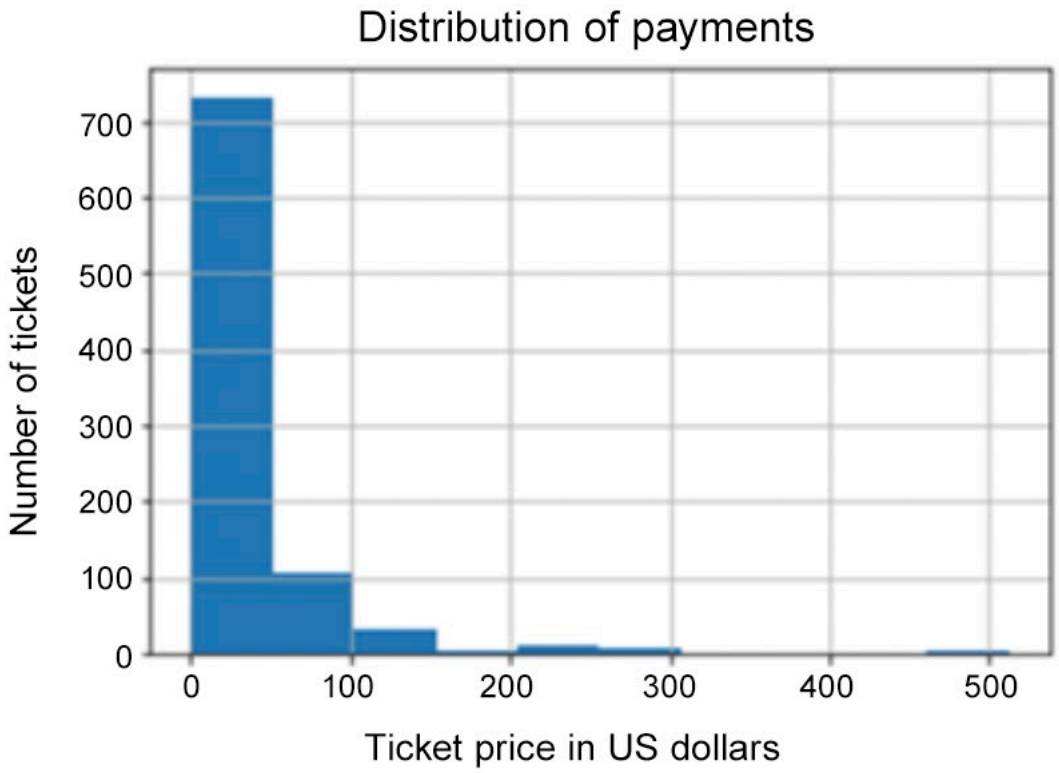


Figure 5.8 – Checking skewness with a histogram

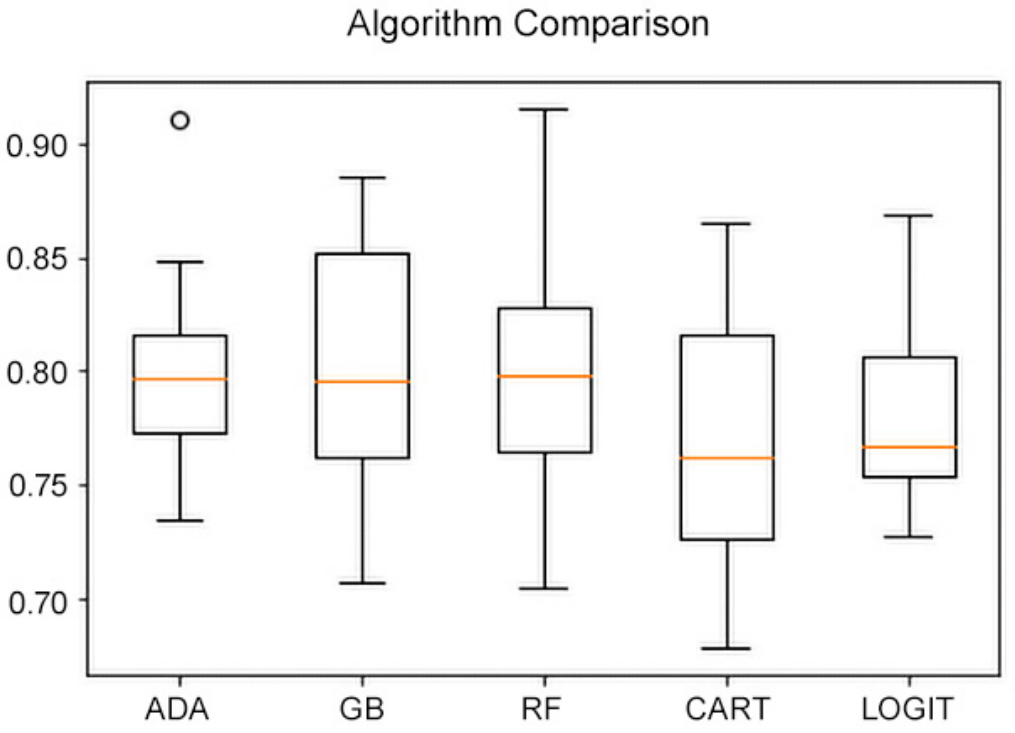


Figure 5.9 – Plotting distributions with a box plot

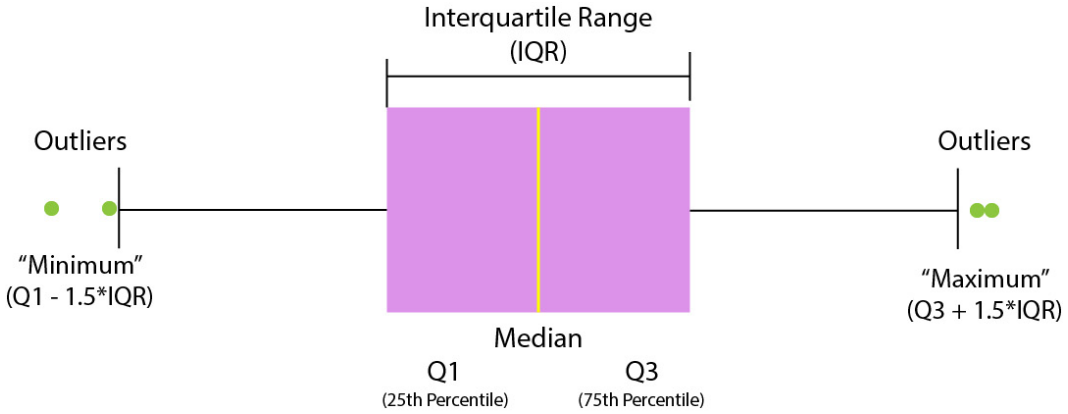


Figure 5.10 – Box plot elements

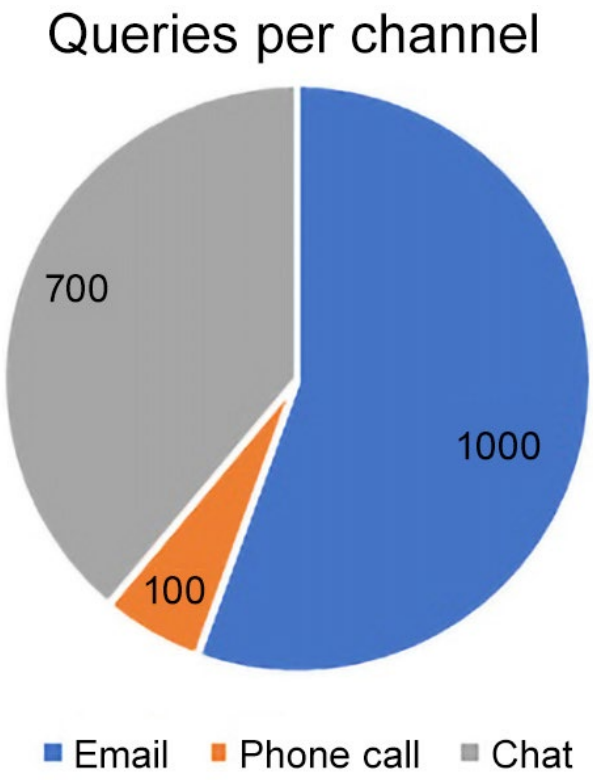


Figure 5.11 – Plotting compositions with a pie chart

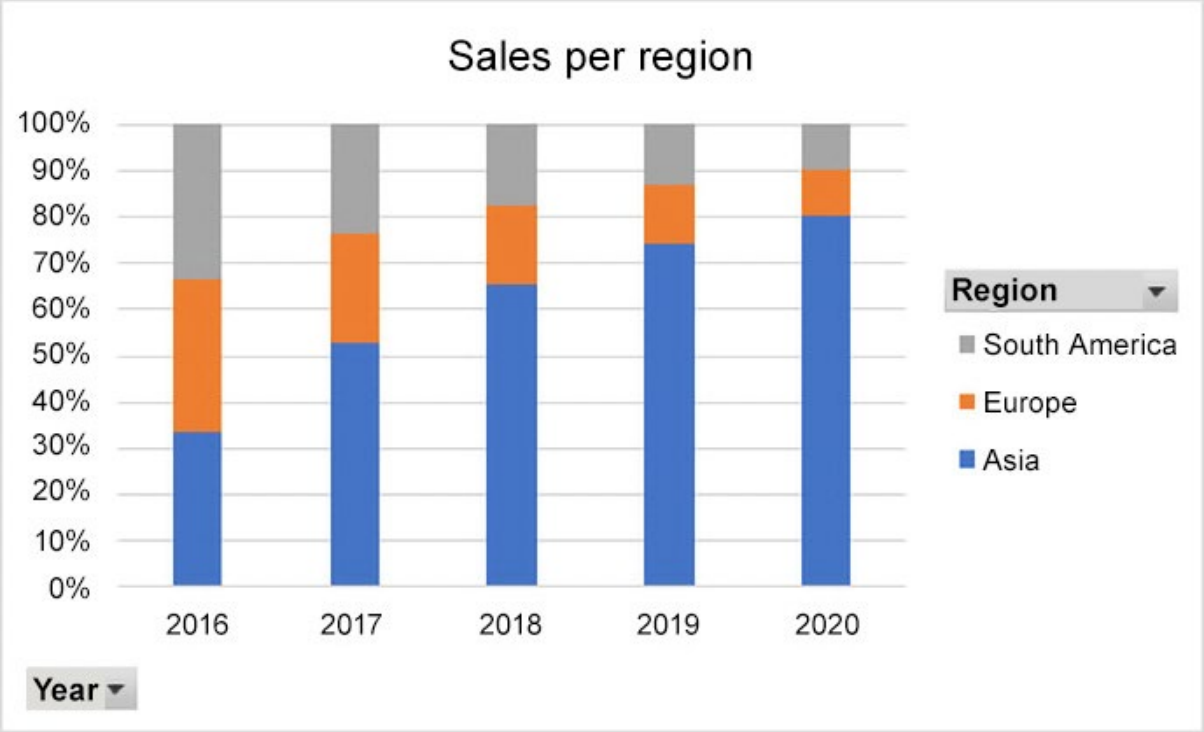


Figure 5.12 – Plotting compositions with a stacked 100% column chart

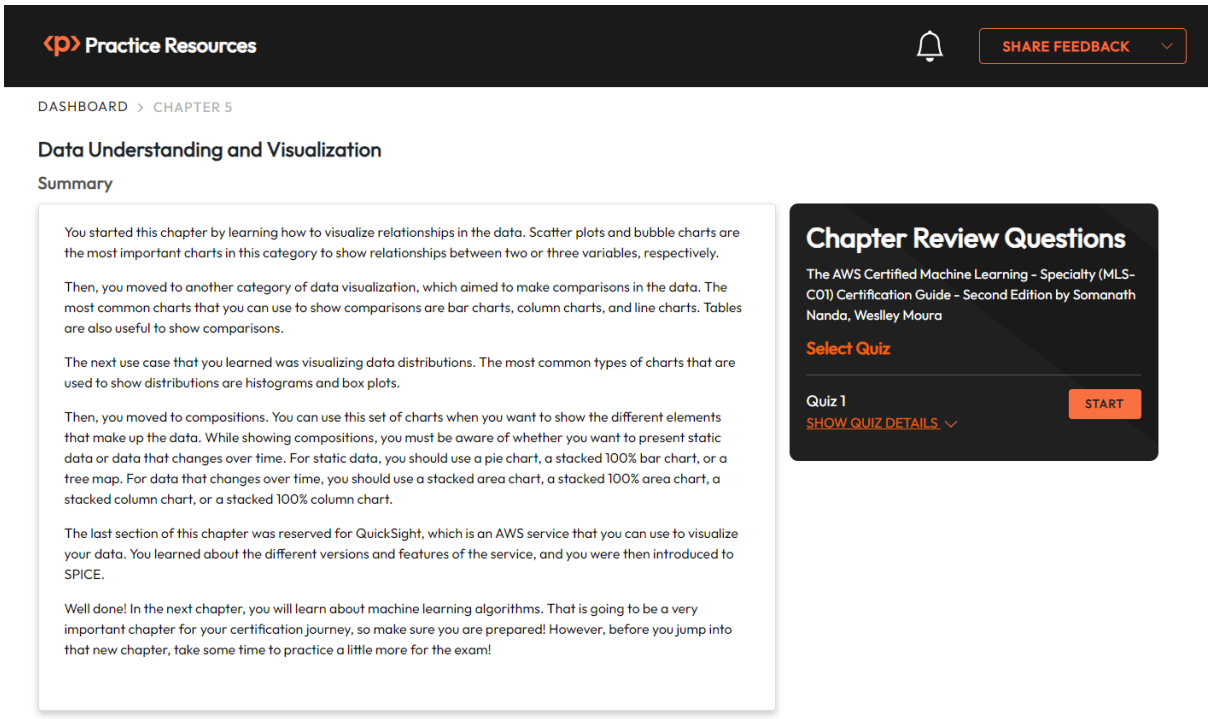


Figure 5.14 – Chapter Review Questions for Chapter 5

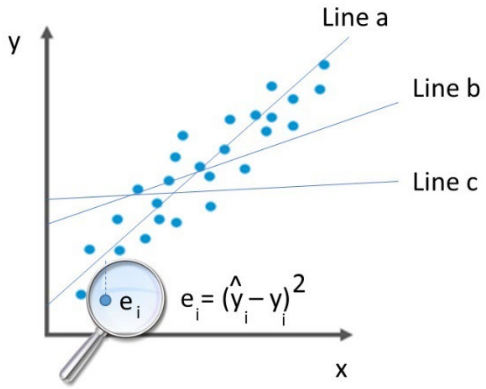


Figure 6.1 – Visualizing the principle of the least squares method

$$S^2 = \frac{\Sigma\left(x_i - \overline{x}\right)}{n}$$

$$Cov\ (X,\ Y) = \frac{\sum_{i=1}^{n}\left(x_i - \overline{x}\right)\left(y_i - \overline{y}\right)}{n}$$

Figure 6.2 – Mathematical representation of variance and covariance respectively

Line inclination | Angular coefficient (A, Alpha)

$$A = cov(x, y) / var(x)$$

Interceptor (B, Beta)

$$B = mean\ Y - (Alpha * mean\ X)$$

Correlation coefficient R

$$R = cov(x, y) / sqrt(variance\ x * variance\ y)$$

Adjusted Correlation coefficient (R adjusted)

$$R\ adjusted = R \wedge 2$$

Error (variability not explained)

$$Ei = Yi - yi$$

Equation

$$Y = A.X + B + Ei$$

Figure 6.3 – Equations to calculate coefficients for simple linear regression

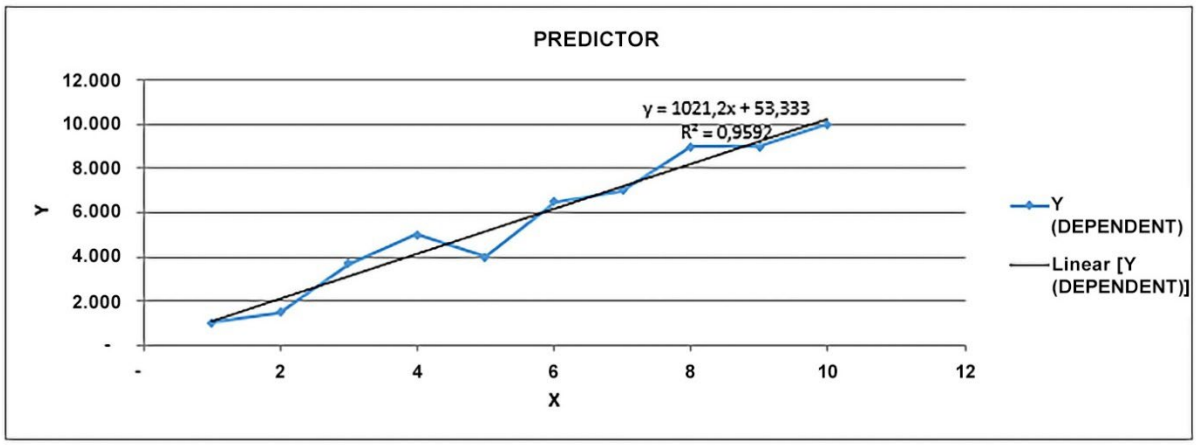


Figure 6.4 – Fitting data in the regression equation

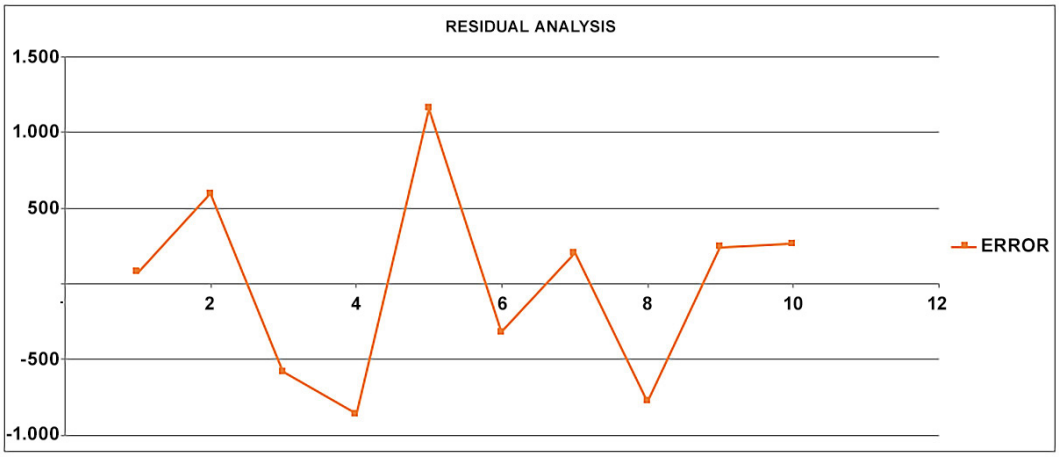


Figure 6.5 – Residual analysis

$$L_1 = (ax + b - y)^2 + \lambda|a|$$

$$L_2 = (ax + b - y)^2 + \lambda a^2$$

Figure 6.6 – L1 and L2 regularization

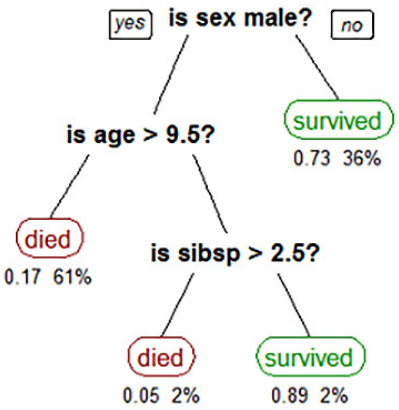


Figure 6.7 – Example of what a decision tree model looks like

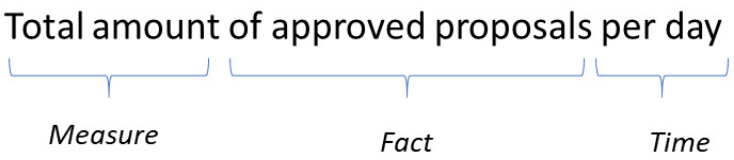


Figure 6.8 – Time series statement

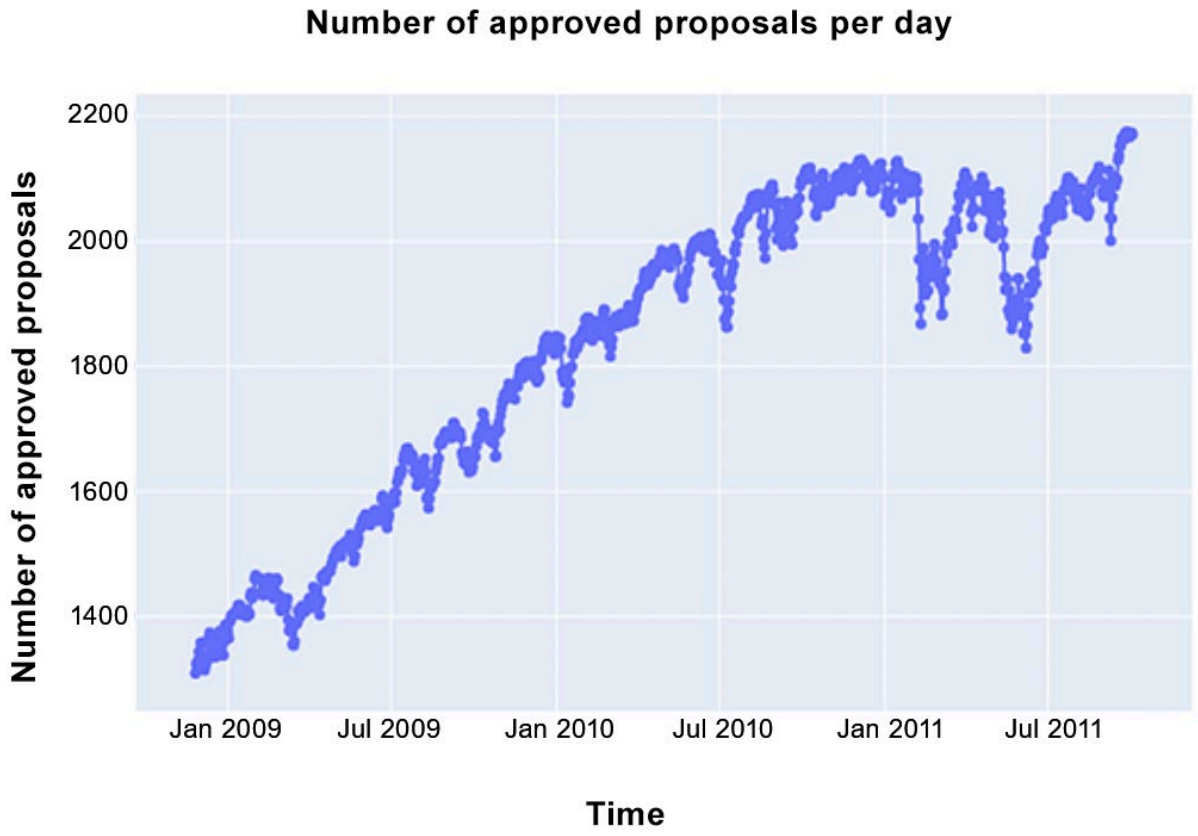


Figure 6.9 – Time series example

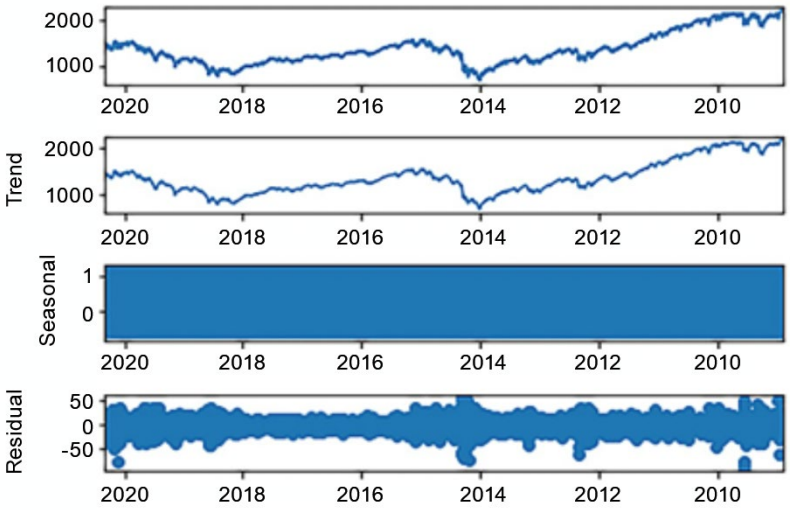


Figure 6.10 – Time series decomposition

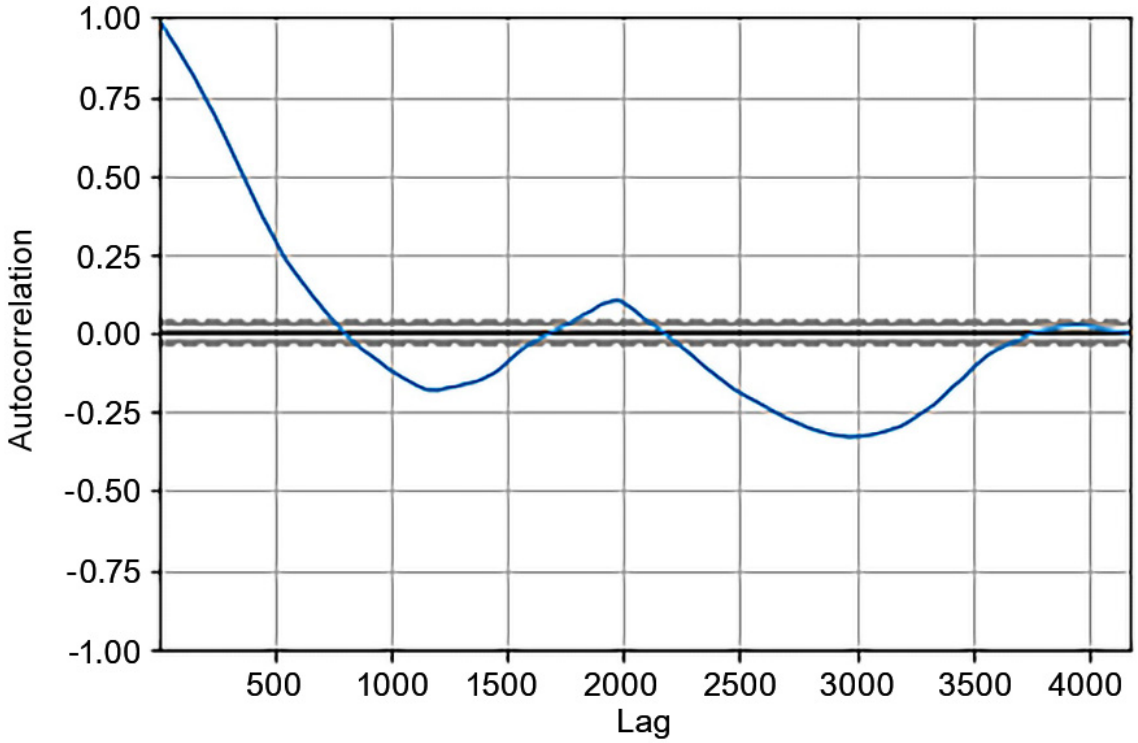


Figure 6.11 – Autocorrelation plot

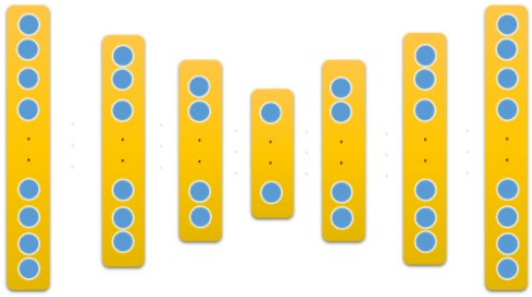


Figure 6.12 – A visual example of an embedding space

$$\sqrt{((x1 - x2)^2 + (y1 - y2)^2)}$$

$$\sqrt{\sum_{i=1}^{n}(y_i - x_i)^2}$$

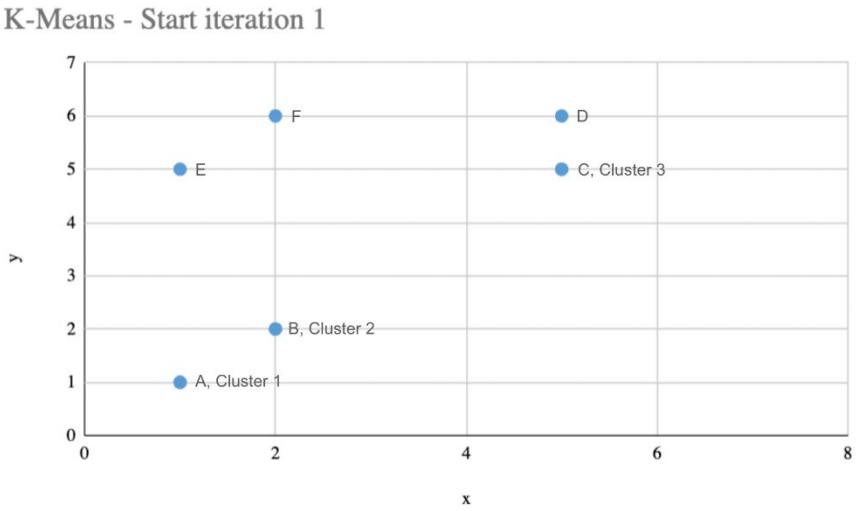


Figure 6.13 – Plotting the K-Means results before completing the first iteration

$$\sqrt{((x1 - x2)^2 + (y1 - y2)^2)}$$

$$\sqrt{((1-5)^2+(1-5)^2)}$$

$$\sqrt{((4)^2+(4)^2)}$$

$$\sqrt{16+16}$$

$$\sqrt{32}$$

5,656 or ~5,7

Figure 6.14 – Computing the Euclidian distance step by step

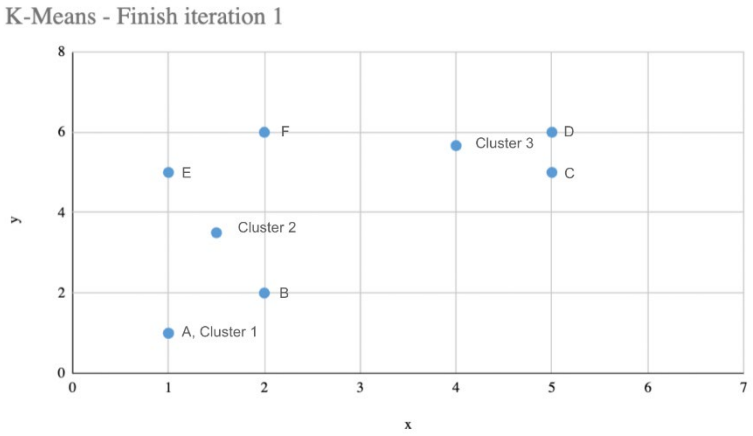


Figure 6.15 – Plotting the K-Means results after the first iteration

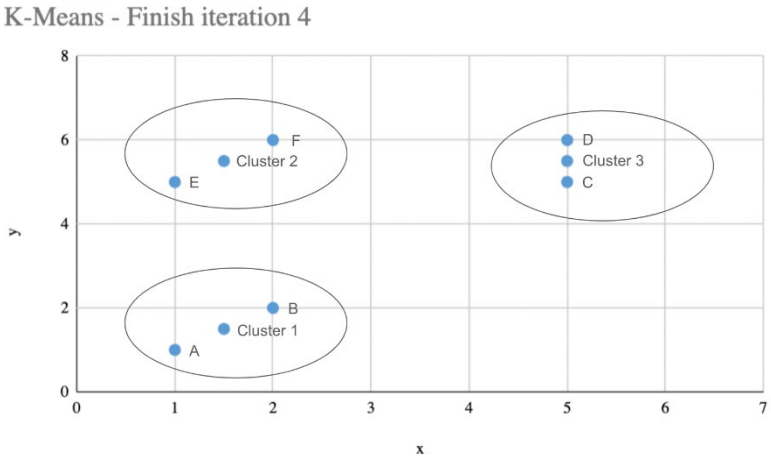


Figure 6.16 – Plotting the K-Means results after the fourth iteration

$$SSE = \sum_{i=1}^{n} (x_i - x_p)^2$$

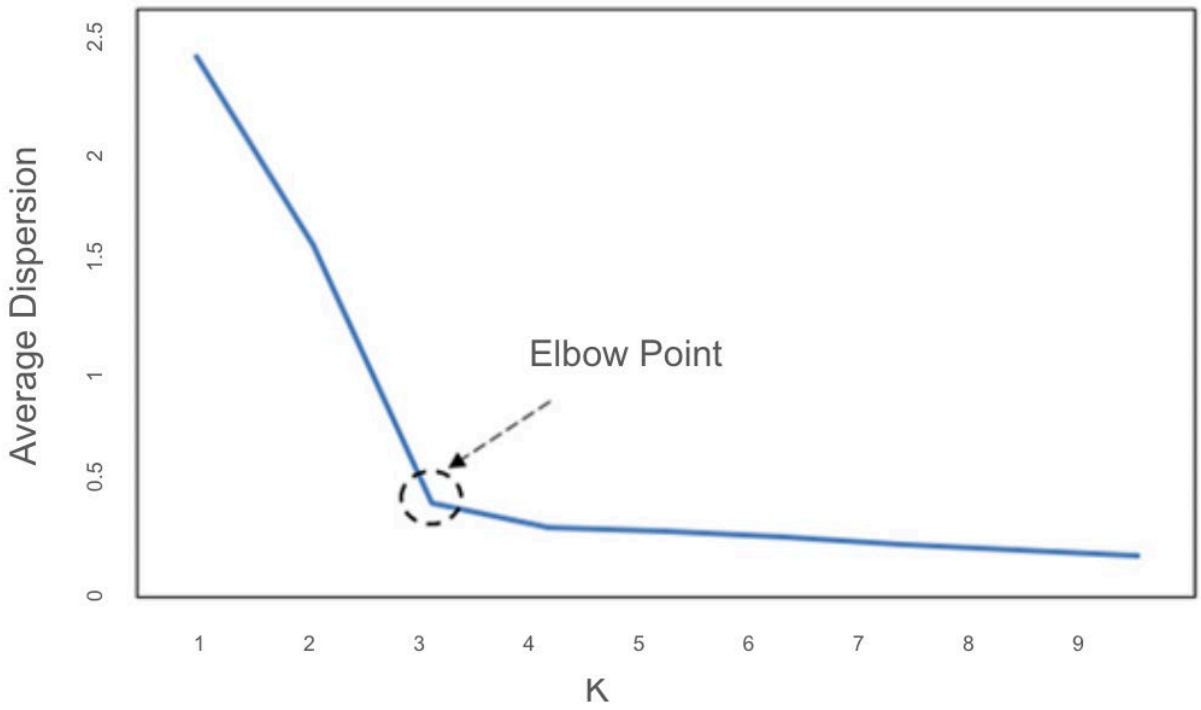


Figure 6.17 – The elbow method

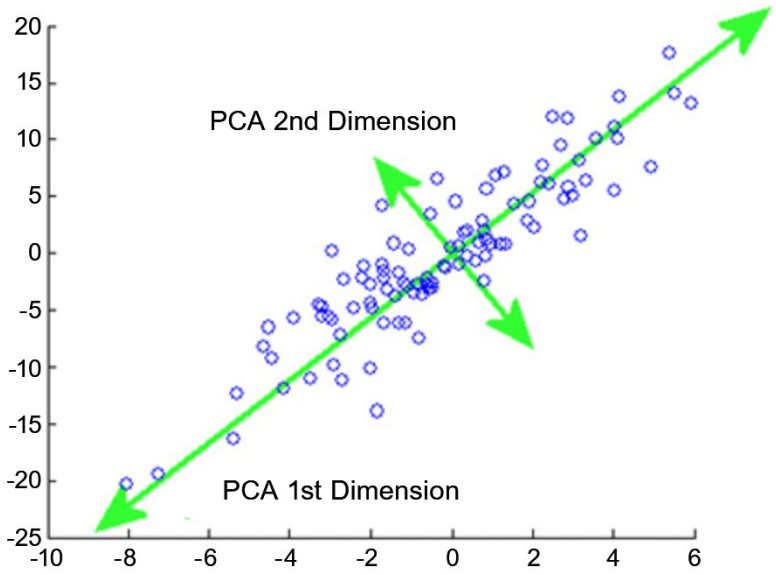


Figure 6.18 – Finding PCs in PCA

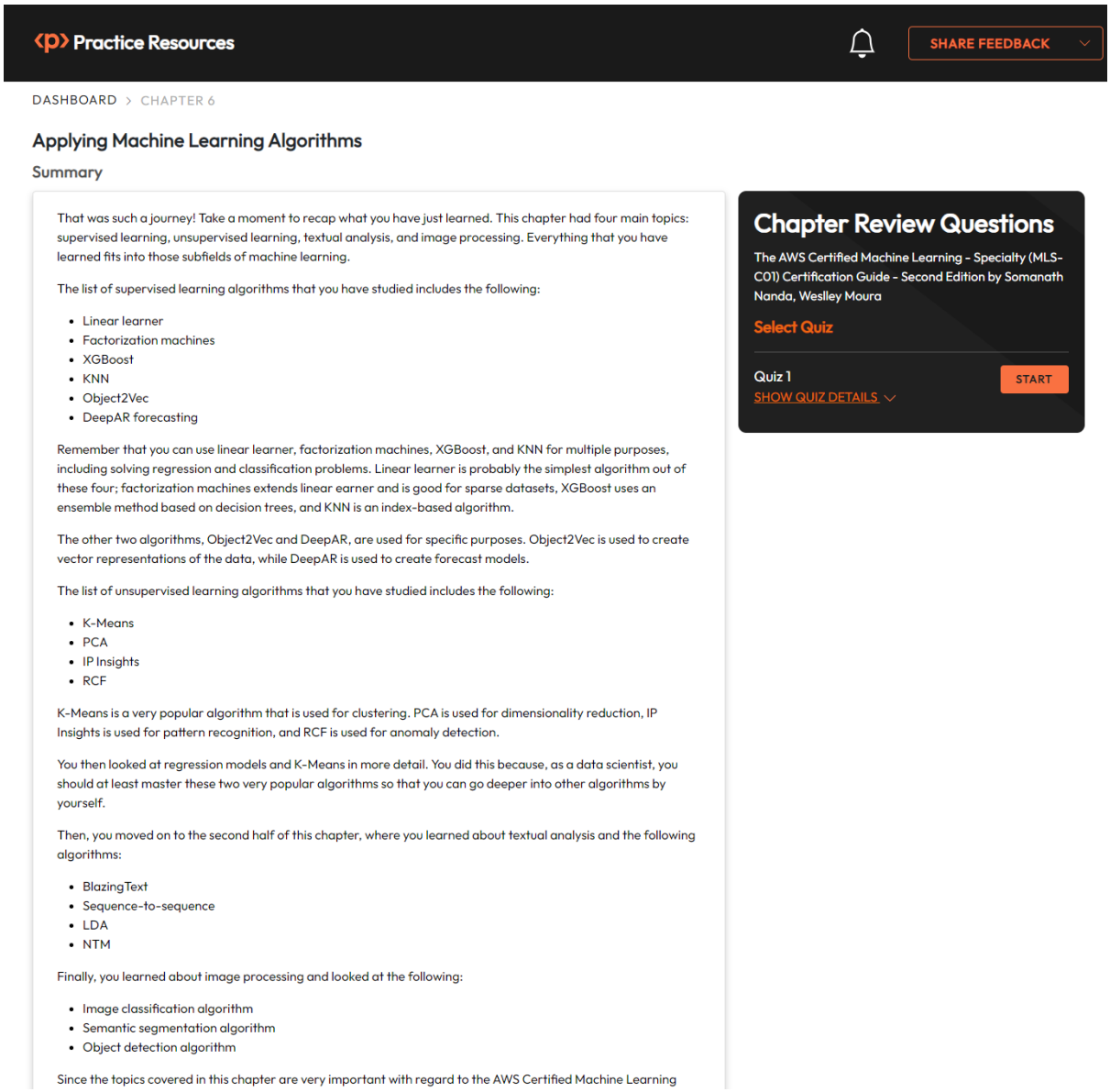


Figure 6.20 – Chapter Review Questions for Chapter 6

Chapter 7: Evaluating and Optimizing Models

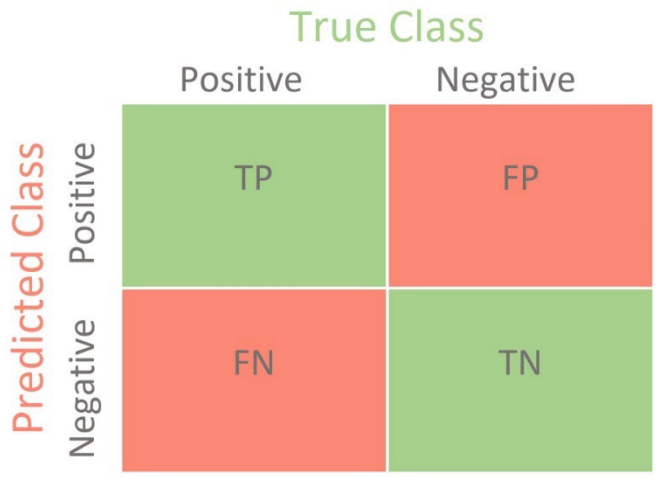


Figure 7.1 – A confusion matrix

$$Accuracy = \frac{TrueNegatives + TruePositive}{TruePositive + FalsePositive + TrueNegative + FalseNegative}$$

Figure 7.2 – Formula for accuracy

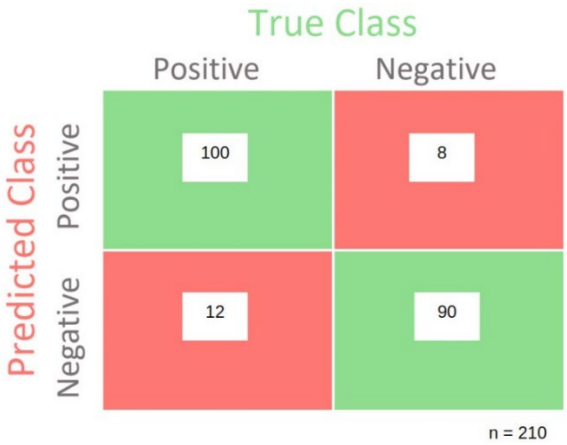


Figure 7.3 – A confusion matrix filled with some examples

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Figure 7.4 – Formula for recall

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Figure 7.5 – Formula for precision

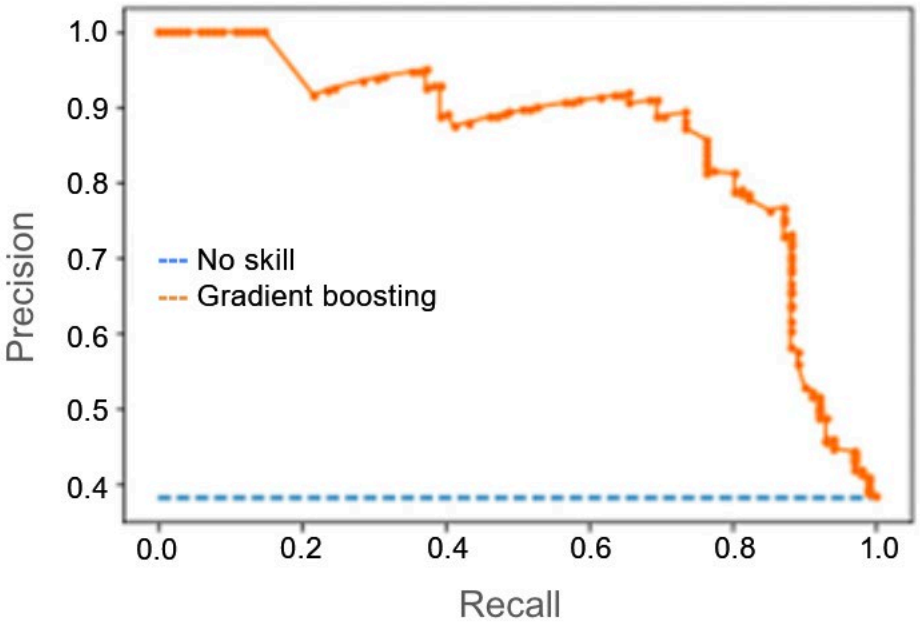


Figure 7.6 – A precision-recall curve

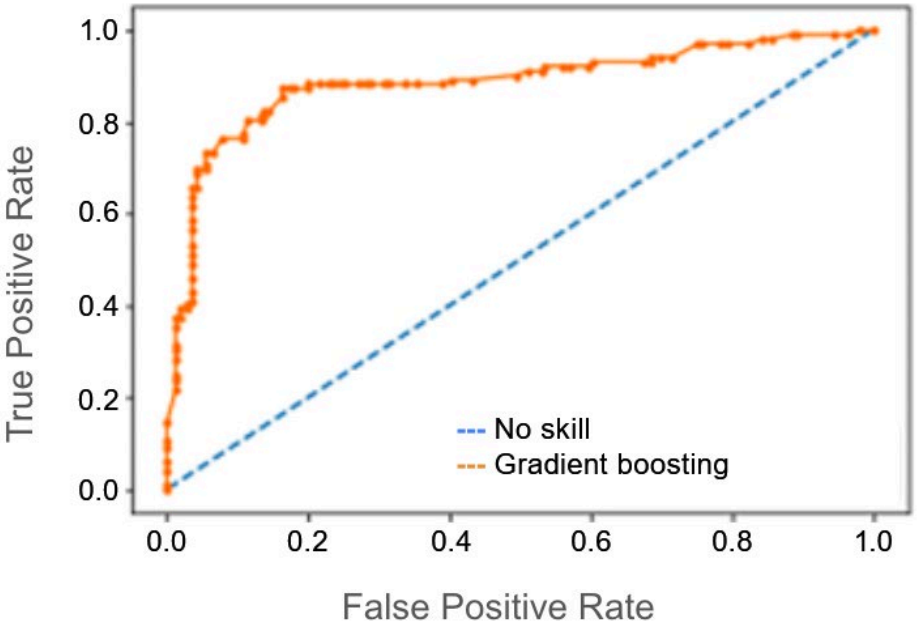


Figure 7.7 – ROC curve

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

Figure 7.8 – Formula for error of each prediction

$$\text{MSE} = \underbrace{\frac{1}{n}\sum_{i=1}}_{\text{test set}}^{n} \underbrace{(y_i}_{\text{predicted vaue}} - \underbrace{\hat{y}_i)^2}_{\text{actual value}}$$

Figure 7.9 – Formula for MSE

$$RMSE = \sqrt{\sum_{i=1}^{n}\frac{(\hat{y}_i - y_i)^2}{n}}$$

Figure 7.10 – Formula for RMSE

$$MAPE = \overbrace{\frac{100\%}{n}}^{\text{Multiplying the value by 100\% converts it into a percentage}} \sum \left| \frac{\overbrace{y - \hat{y}}^{\text{The residual}}}{\underbrace{y}_{\text{Each residual is scaled against the actual value}}} \right|$$

Figure 7.11 – Formula for MAPE

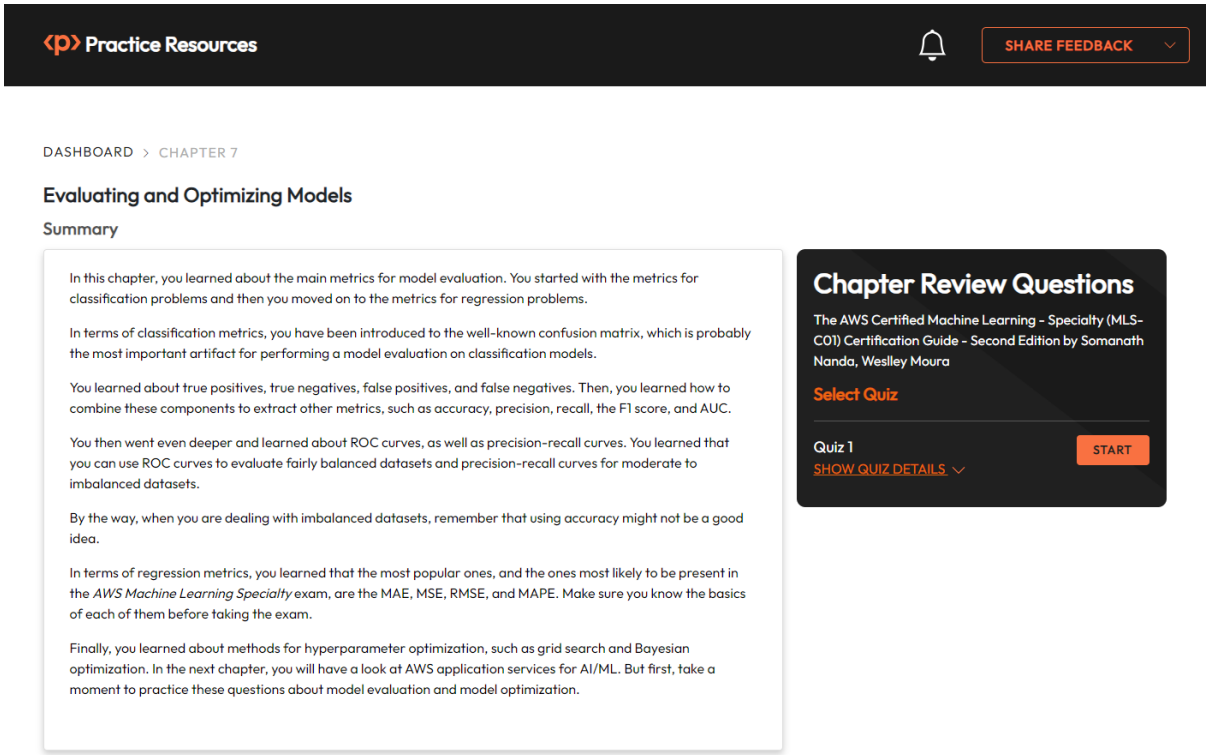


Figure 7.13 – Chapter Review Questions for Chapter 7

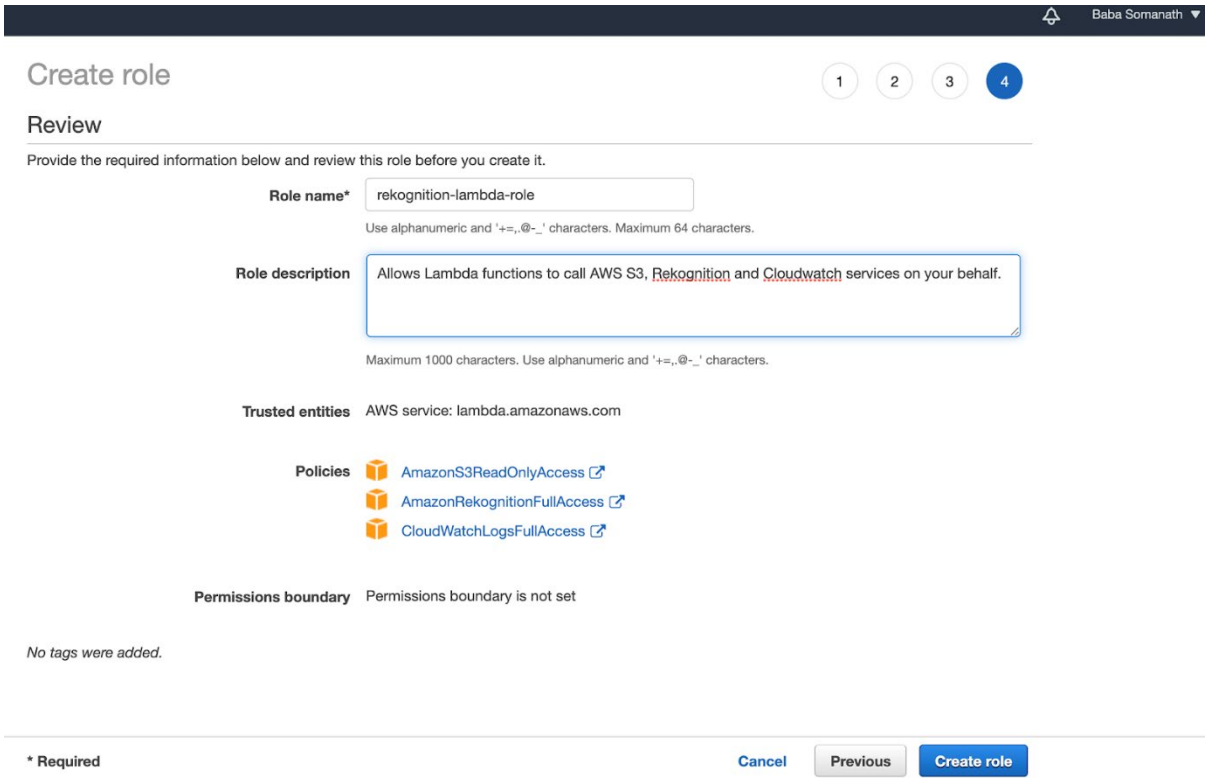


Figure 8.1 – The Create role dialog

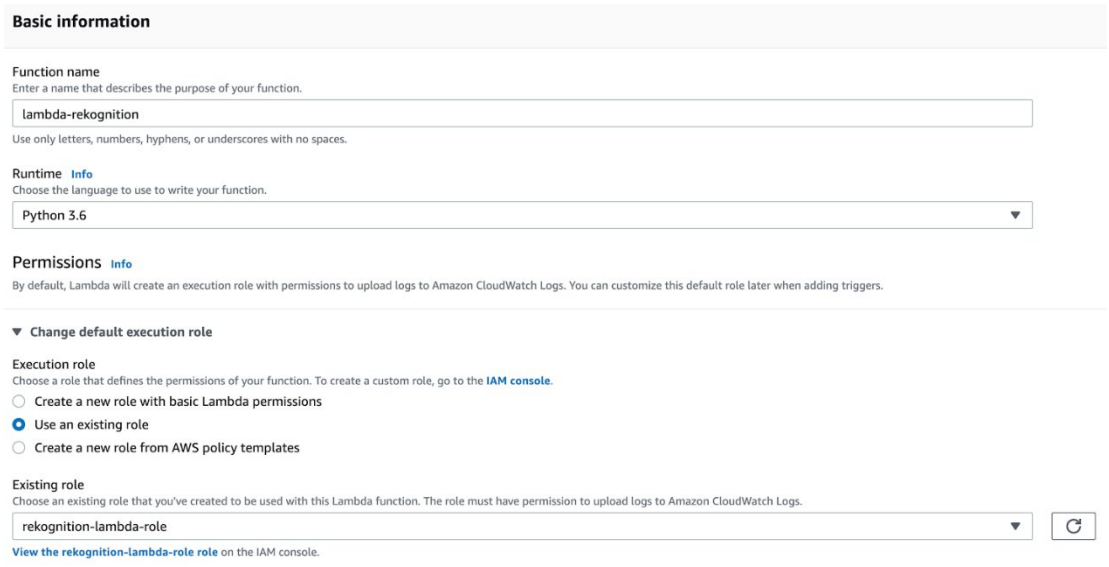


Figure 8.2 – Creating the Lambda function

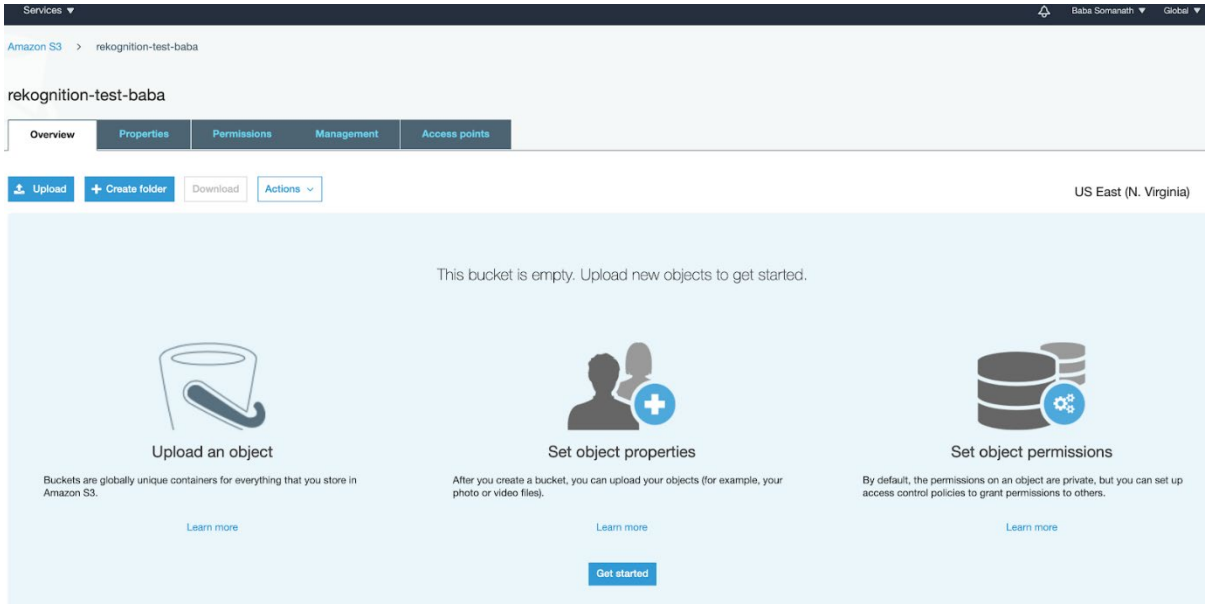


Figure 8.3 – AWS S3 console page

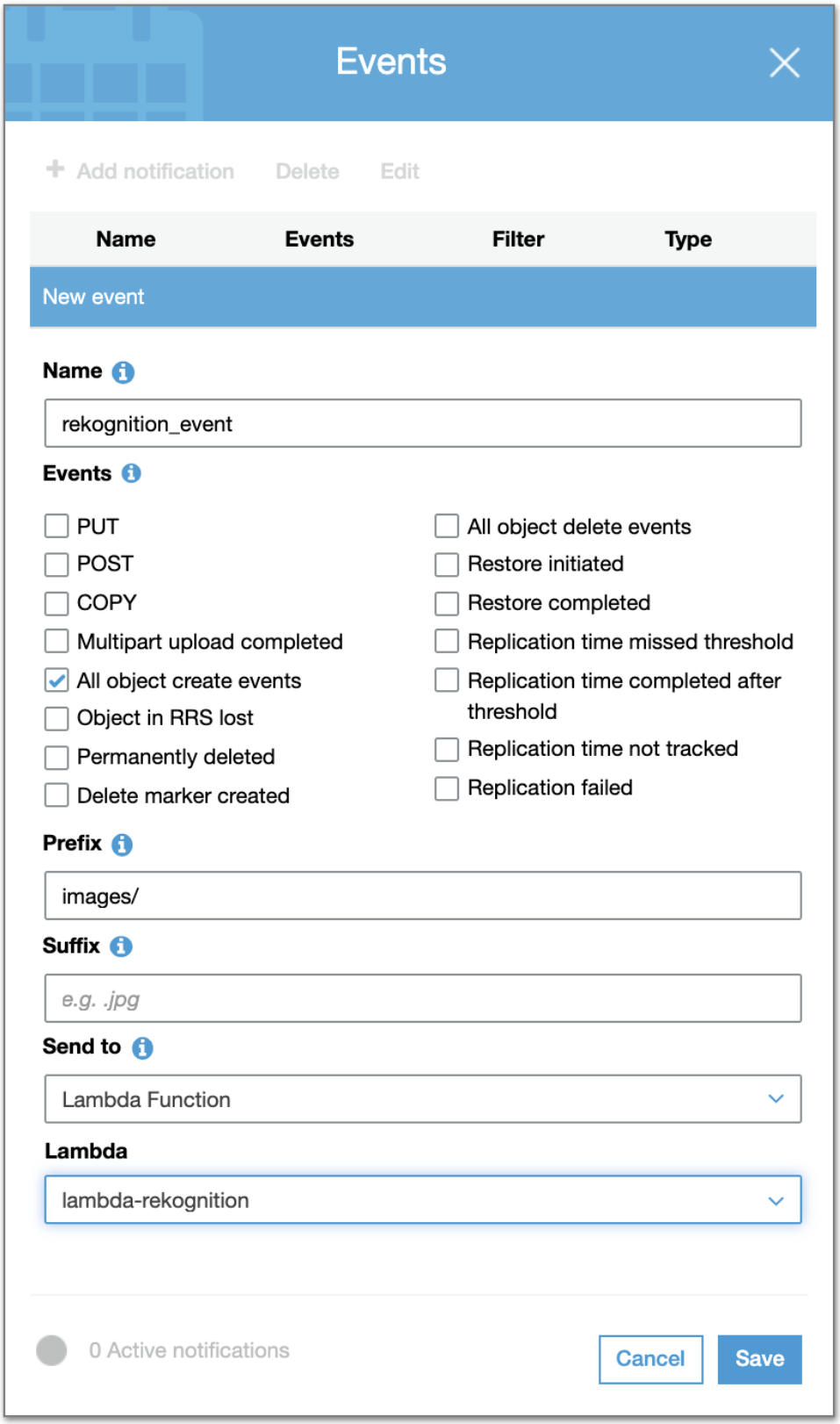


Figure 8.4 – S3 bucket Events window

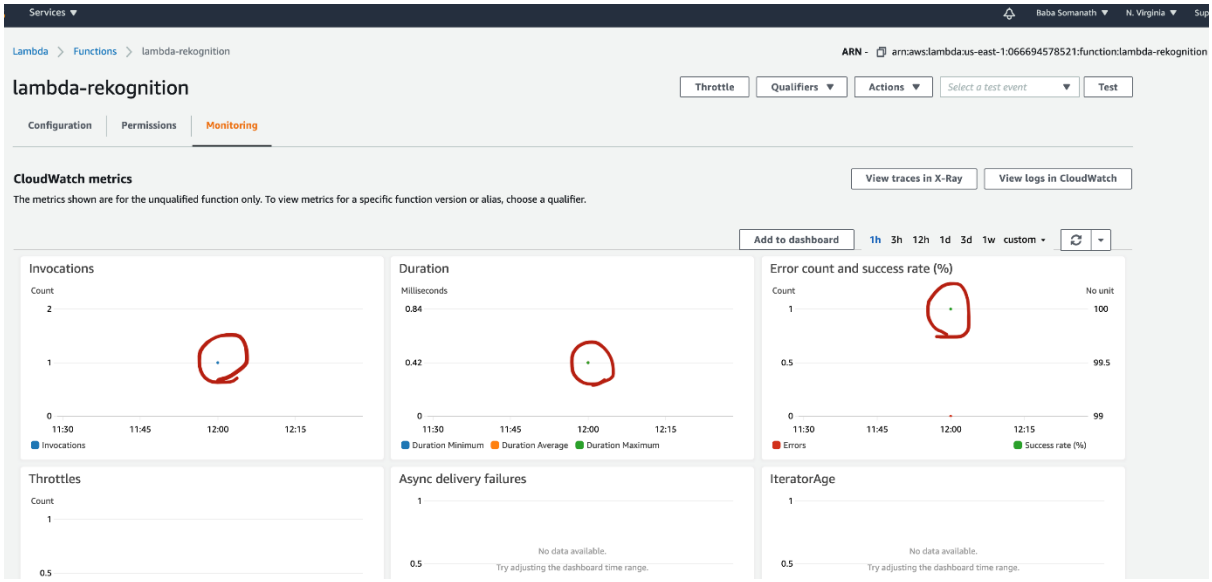


Figure 8.5 – CloudWatch monitoring the event in the Lambda console

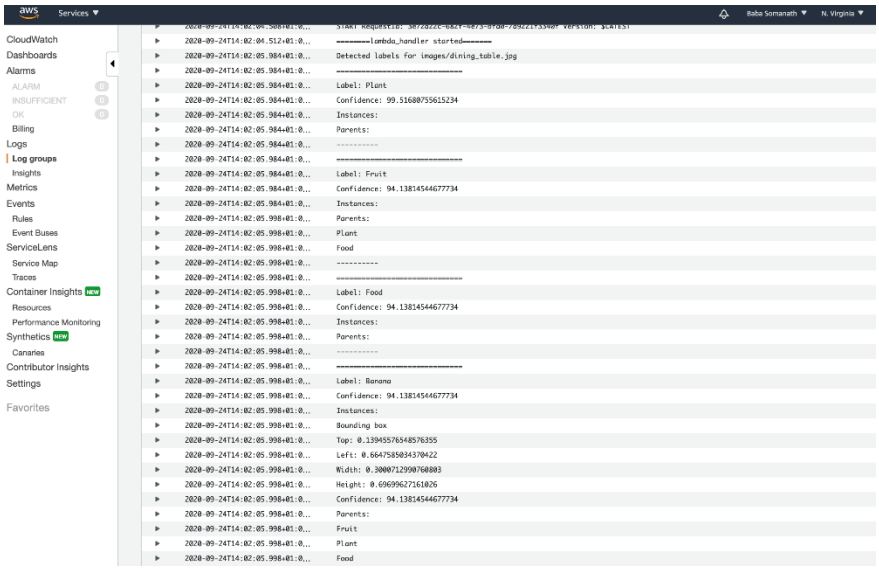


Figure 8.6 – CloudWatch Logs

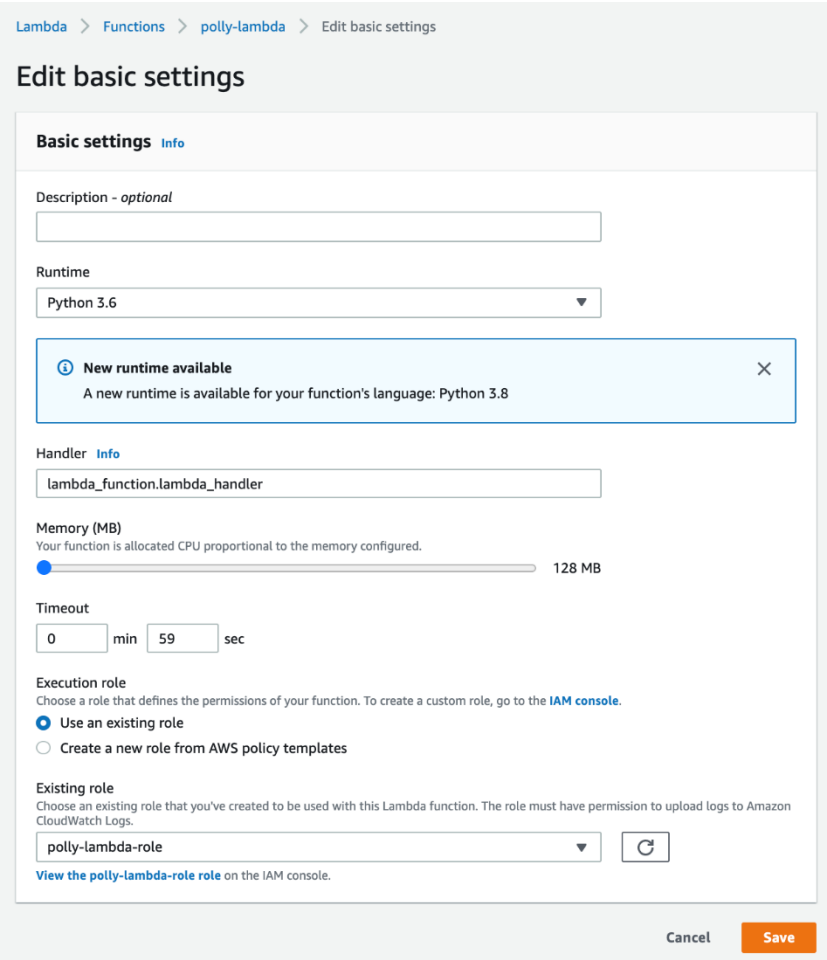


Figure 8.7 – Edit basic settings window

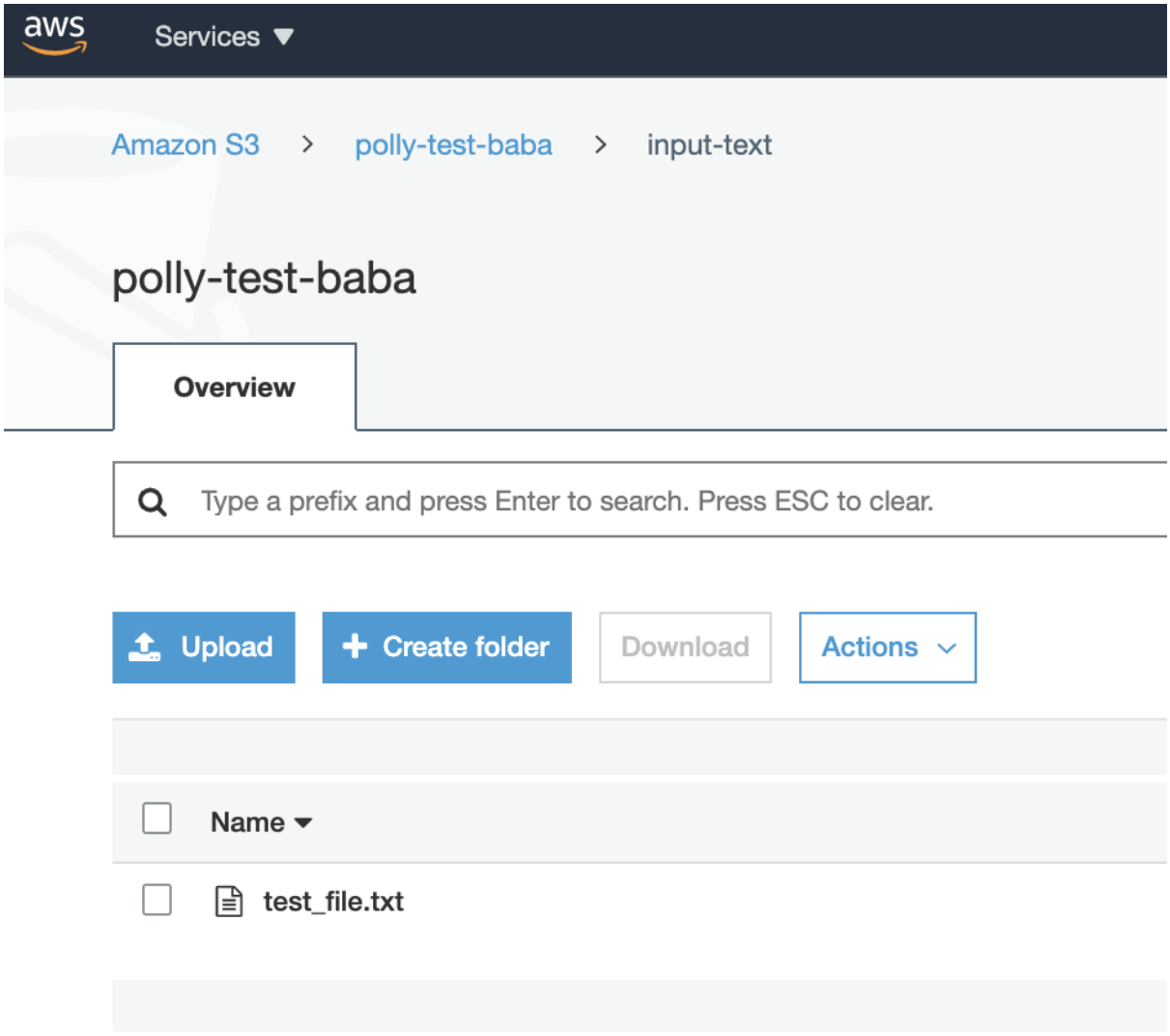


Figure 8.8 – The S3 bucket after uploading a text file for further processing

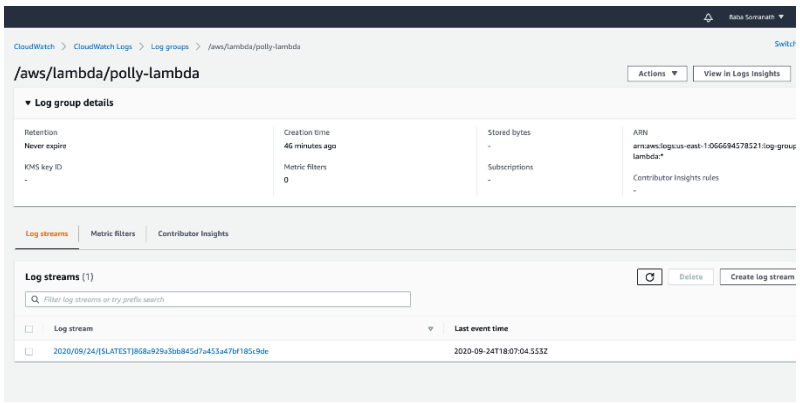


Figure 8.9 – Log groups in the CloudWatch console

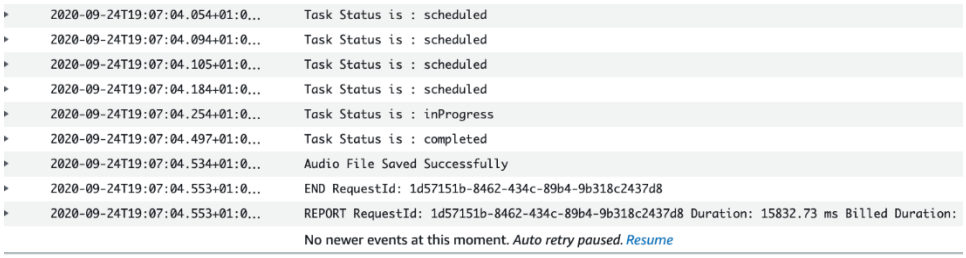


Figure 8.10 – The logs in the CloudWatch console

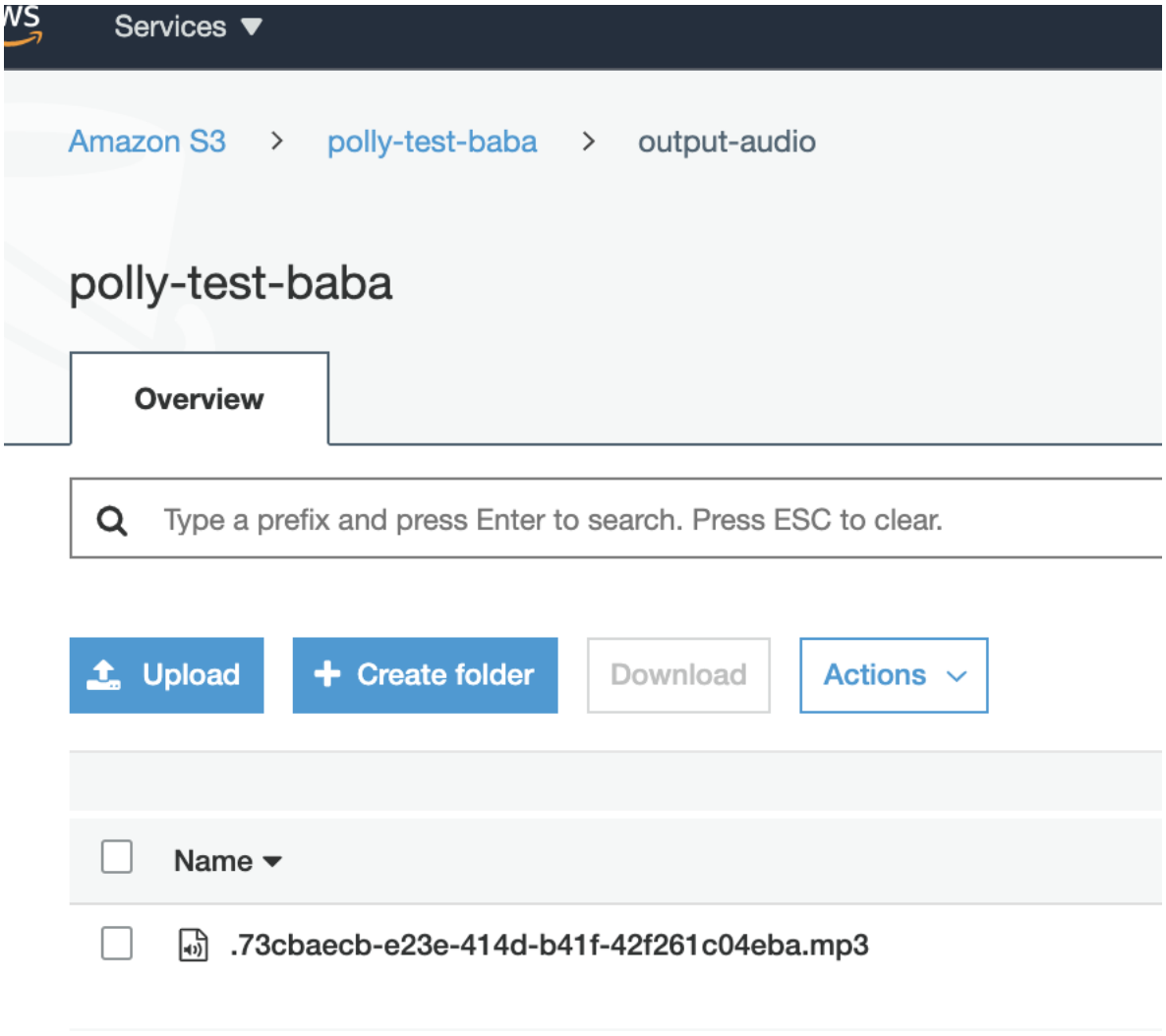


Figure 8.11 – The output file that was created in the S3 bucket

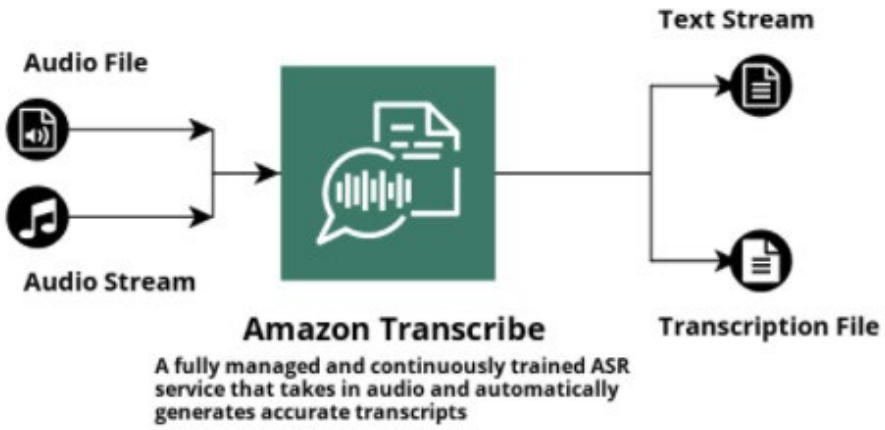


Figure 8.12 – Block diagram of Amazon Transcribe's input and output

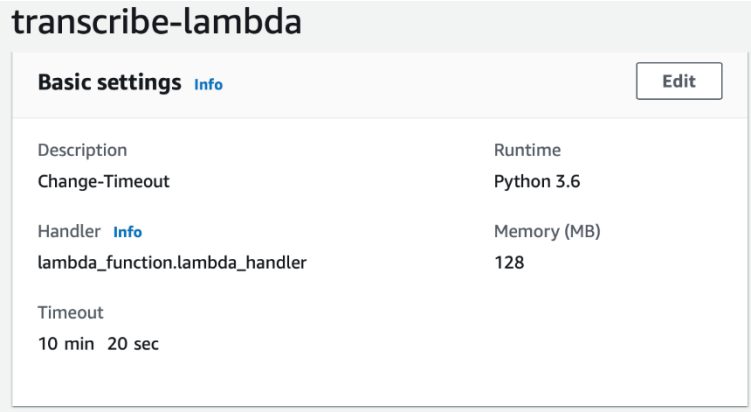


Figure 8.13 – The Basic settings section of our created lambda function

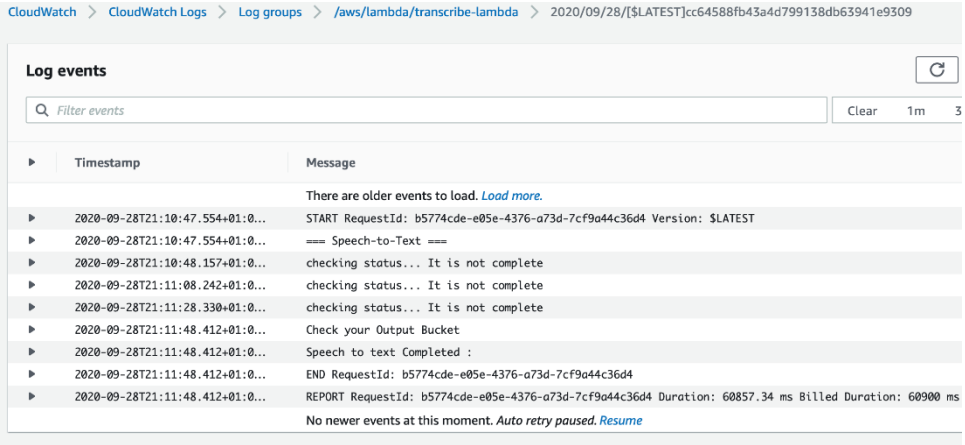


Figure 8.14 – The logs in a Log Stream for the specified log groups in the CloudWatch console

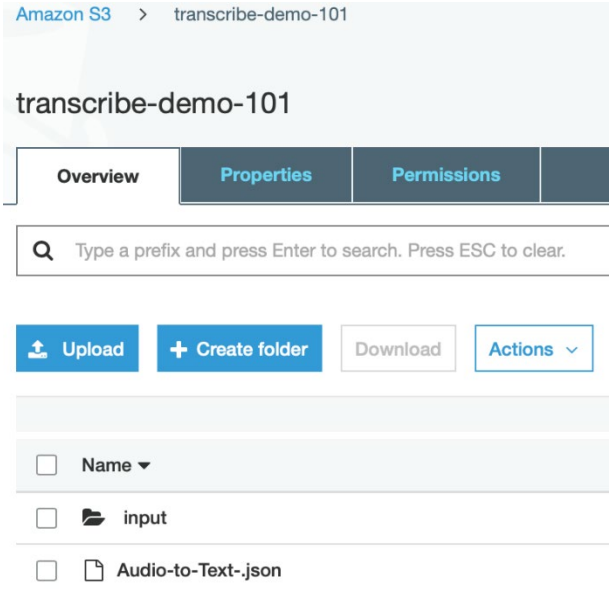


Figure 8.15 – The output JSON file in an S3 bucket

Figure 8.16 – A block diagram showing Amazon Comprehend's capabilities

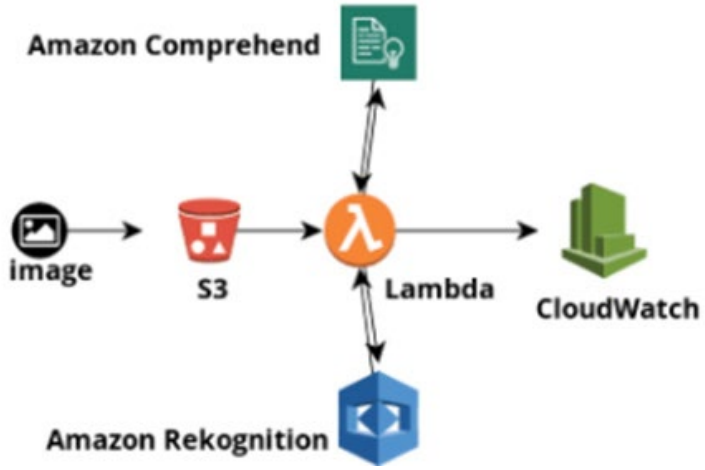


Figure 8.17 – Architecture diagram of the required use case

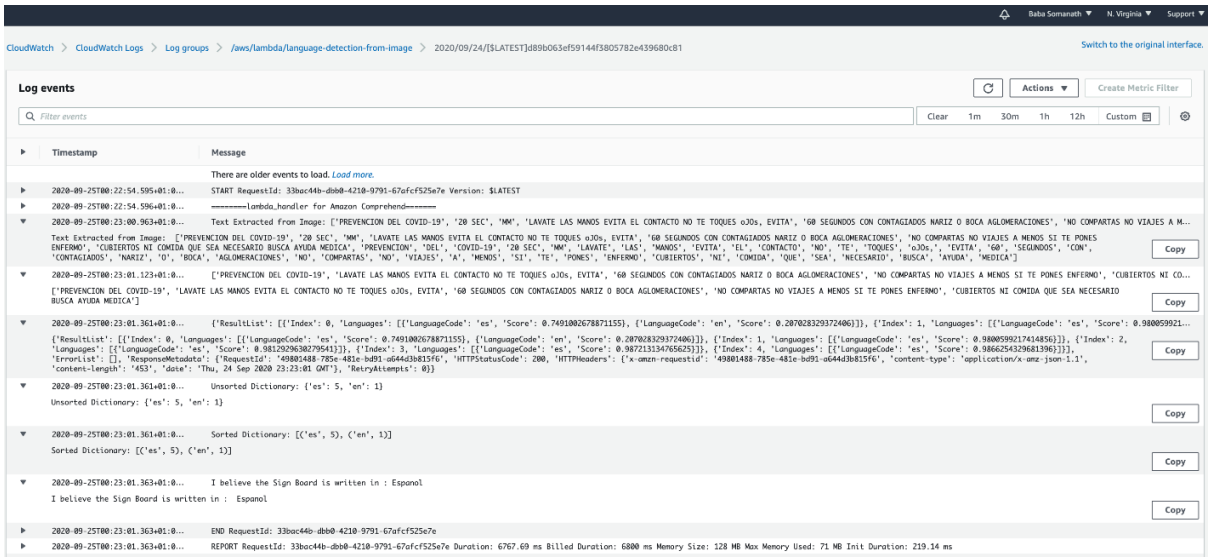


Figure 8.18 – The logs in CloudWatch for verifying the output

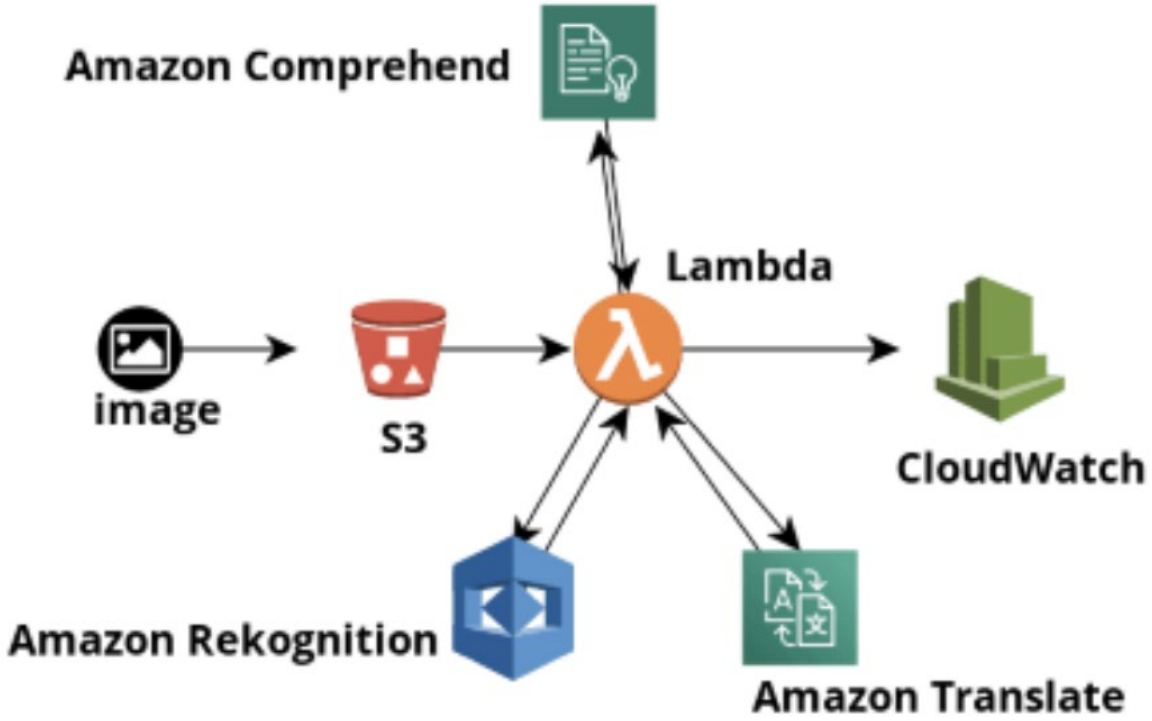


Figure 8.19 – Architecture diagram of the required use case

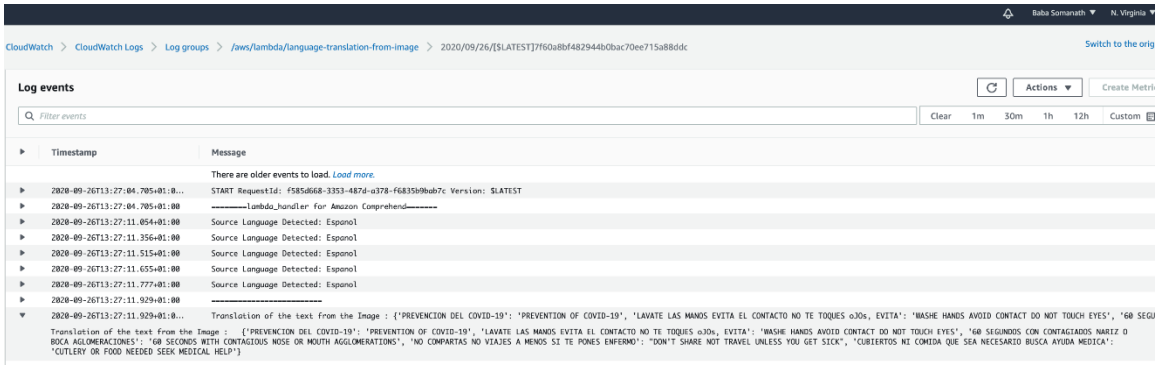


Figure 8.20 – The logs in CloudWatch for verifying the output

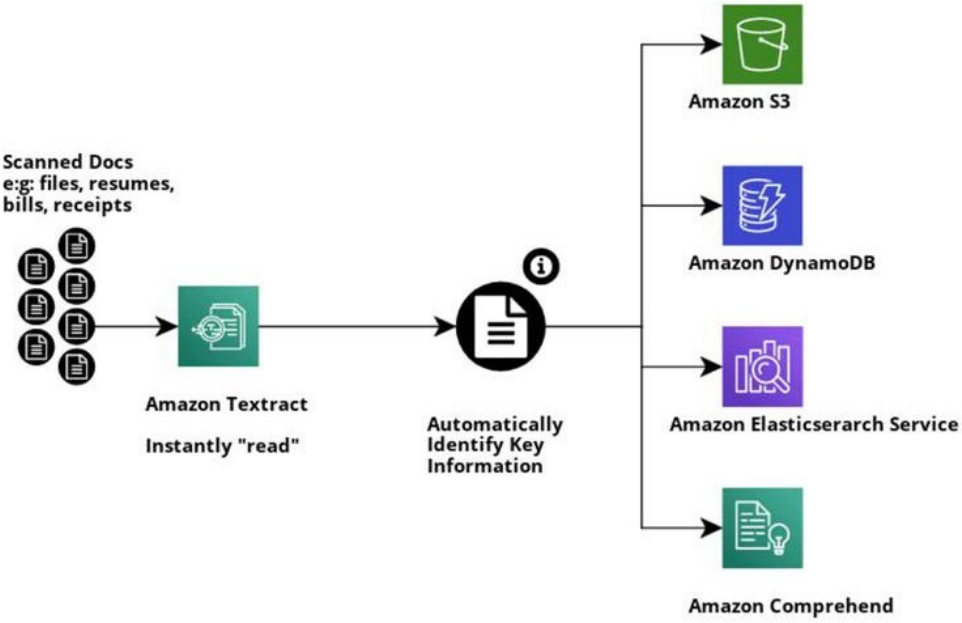


Figure 8.21 – Block diagram representation of Amazon Textract and how it stores its output

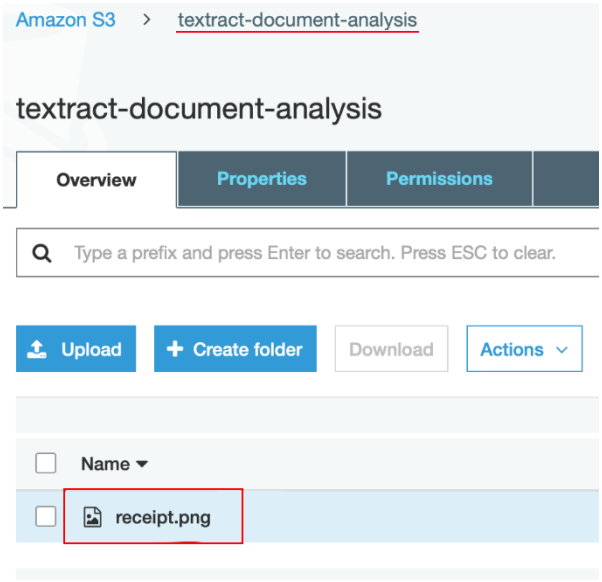


Figure 8.22 – An S3 bucket with an image (.png) file uploaded to the input folder

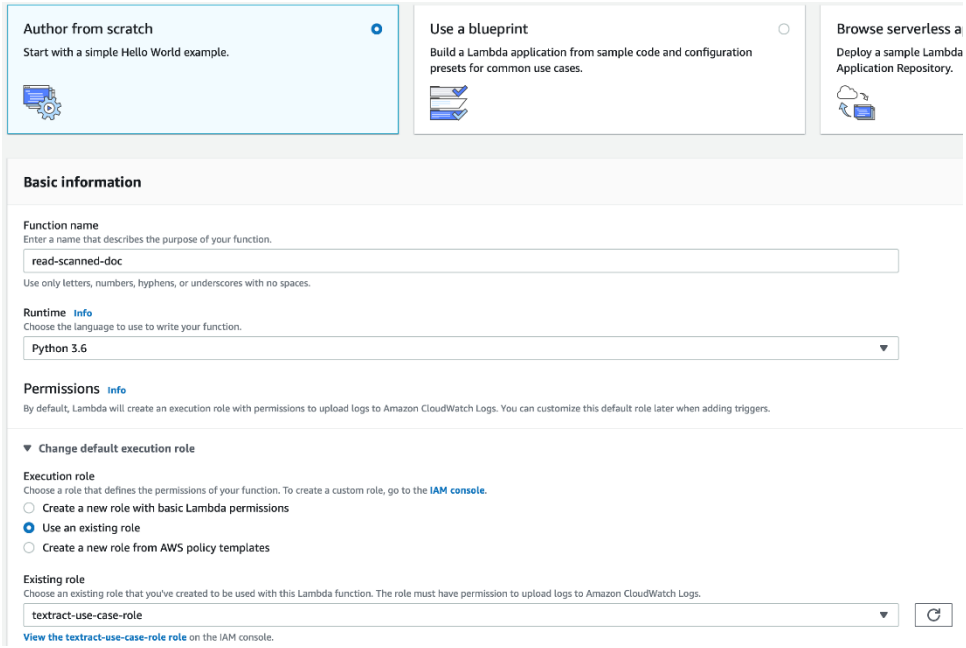


Figure 8.23 – The AWS Lambda Create function dialog

Figure 8.24 – The Event template for testing the Lambda function

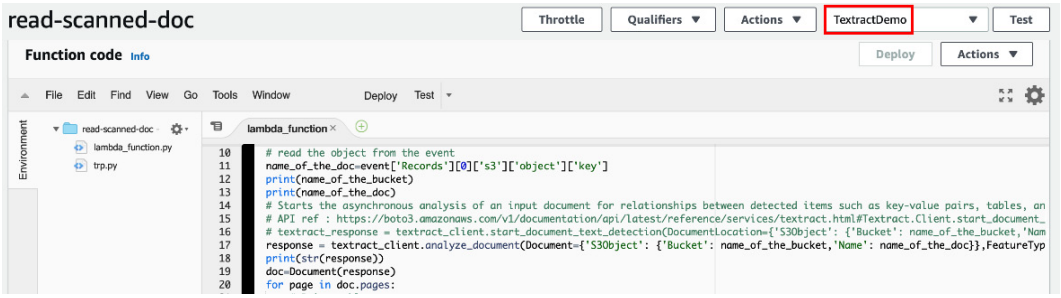


Figure 8.25 – Selecting the test configuration before running your test

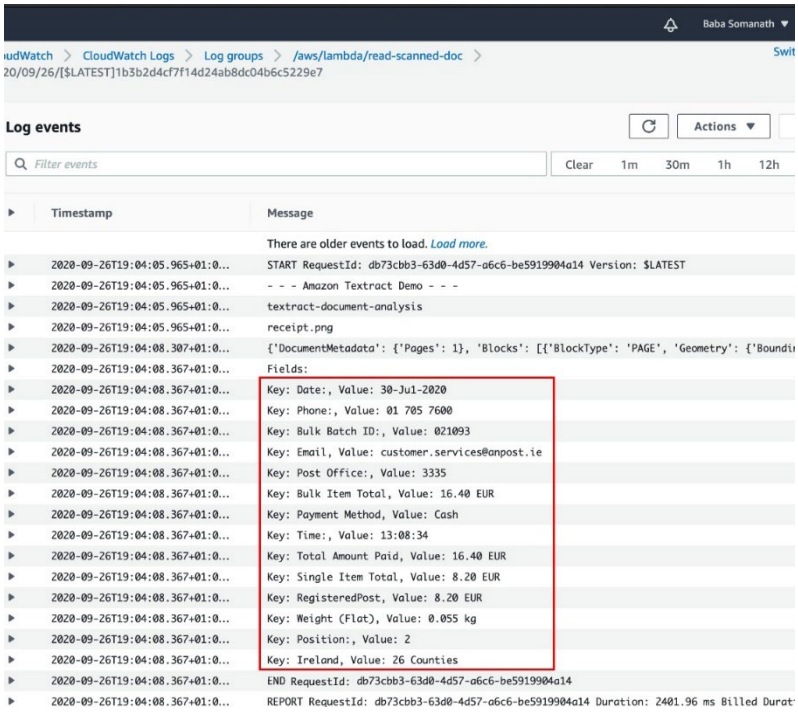


Figure 8.26 – The logs in CloudWatch for verifying the output

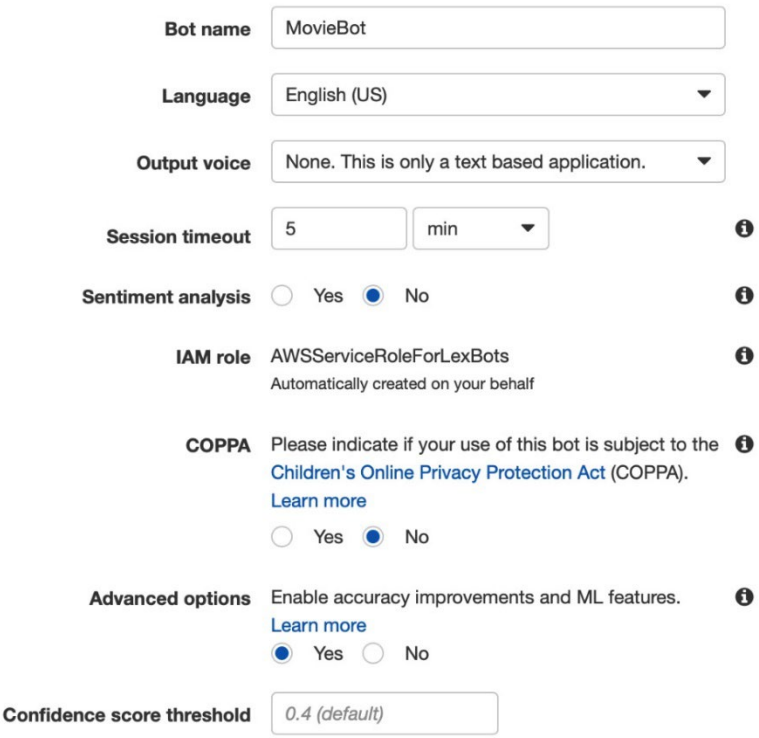


Figure 8.27 – The Create dialog of Amazon Lex

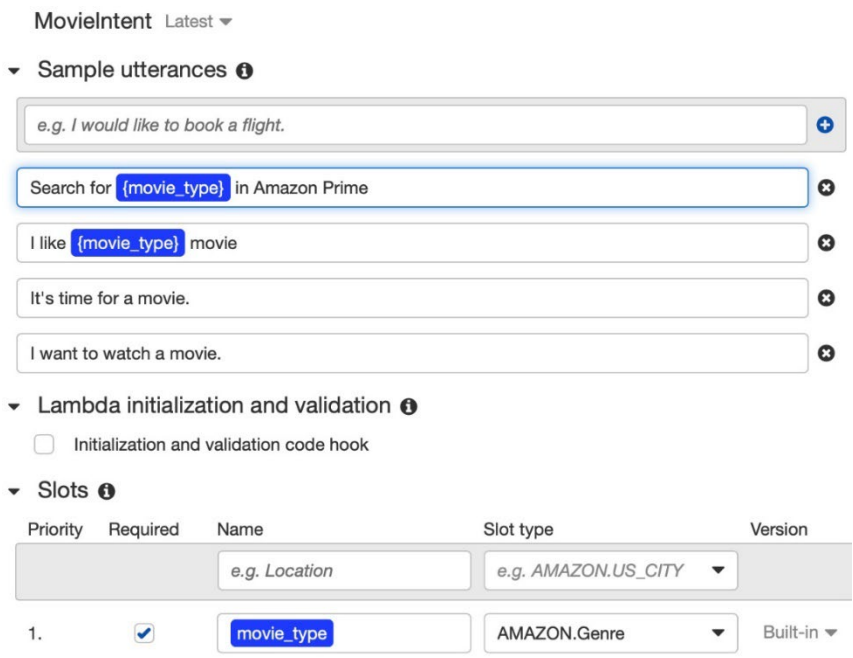


Figure 8.28 – The Sample utterances section

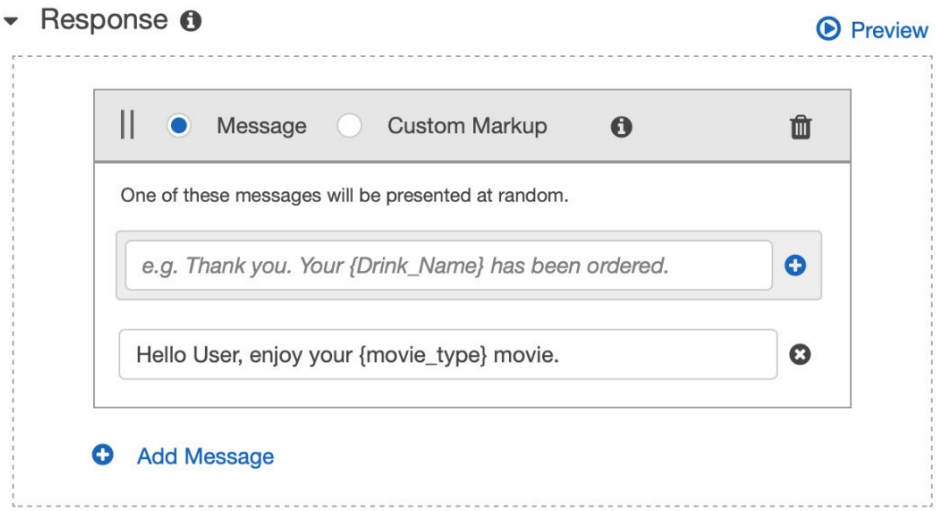


Figure 8.29 – The Response section of Amazon Lex

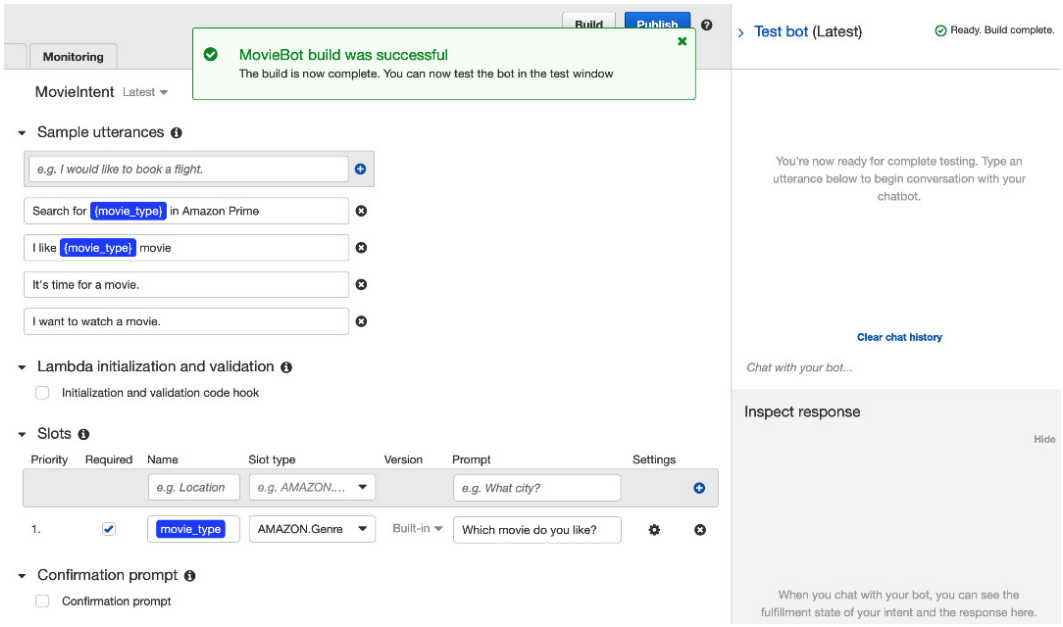


Figure 8.30 – The Response section of Amazon Lex

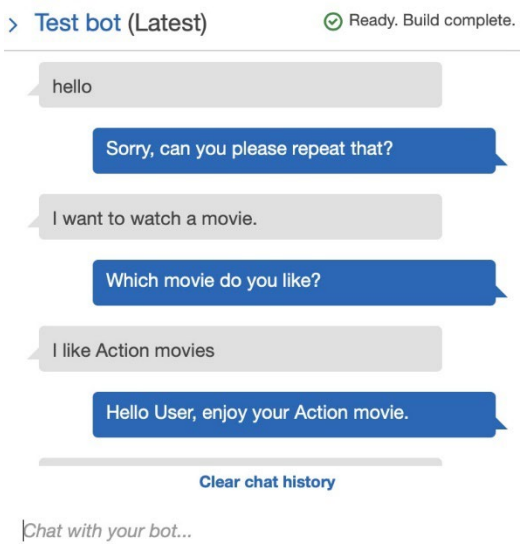


Figure 8.31 – The Test bot dialog

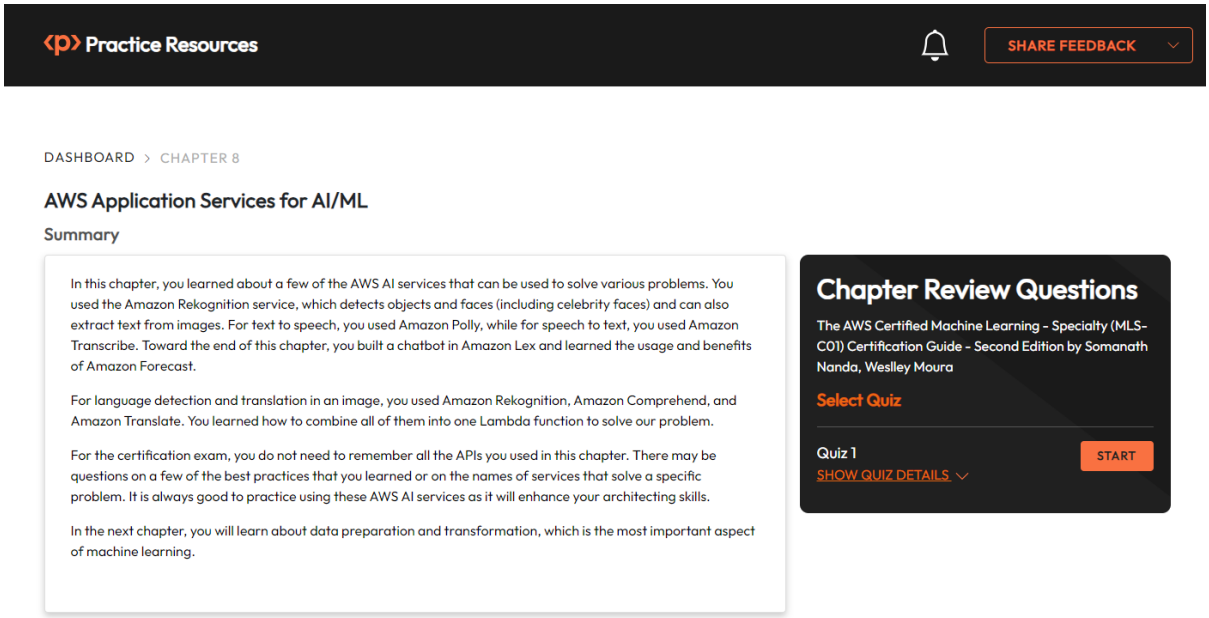


Figure 8.33 – Chapter Review Questions for Chapter 8

Chapter 9: Amazon SageMaker Modeling

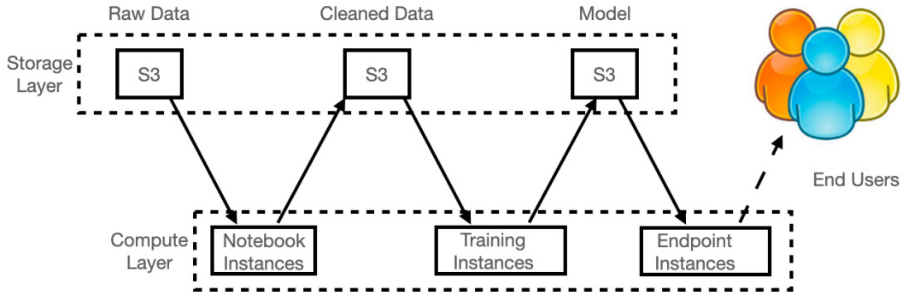


Figure 9.1 – A pictorial representation of the different layers of the Amazon SageMaker instances

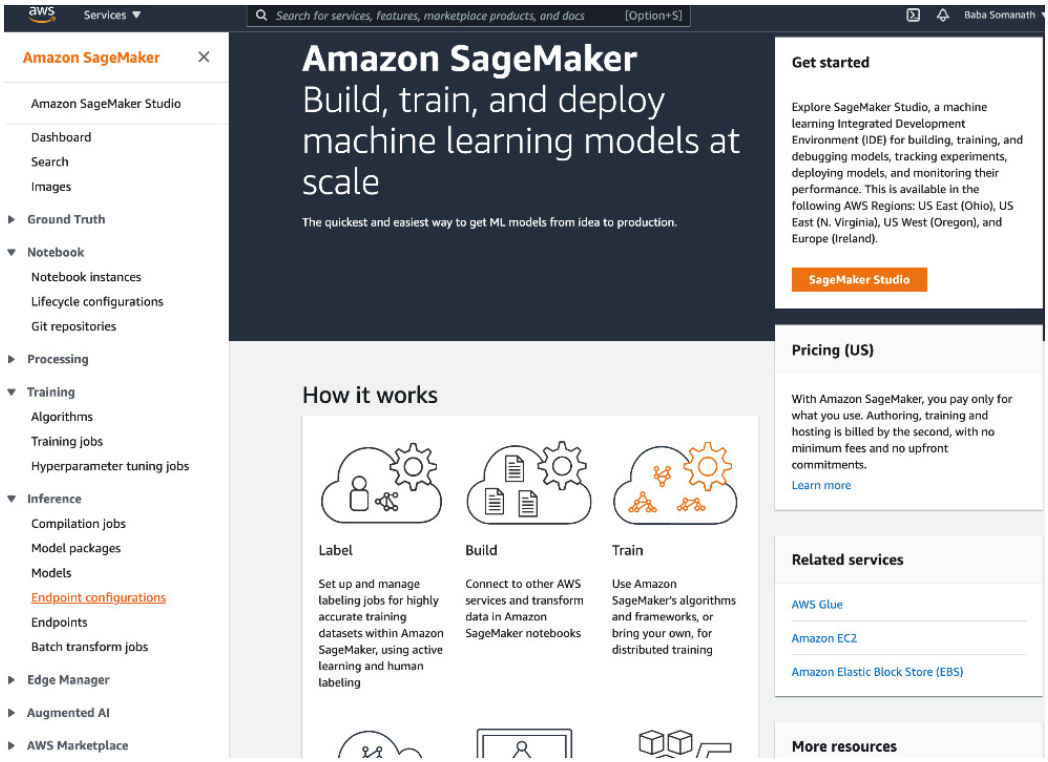


Figure 9.2 – A quick look at the SageMaker console

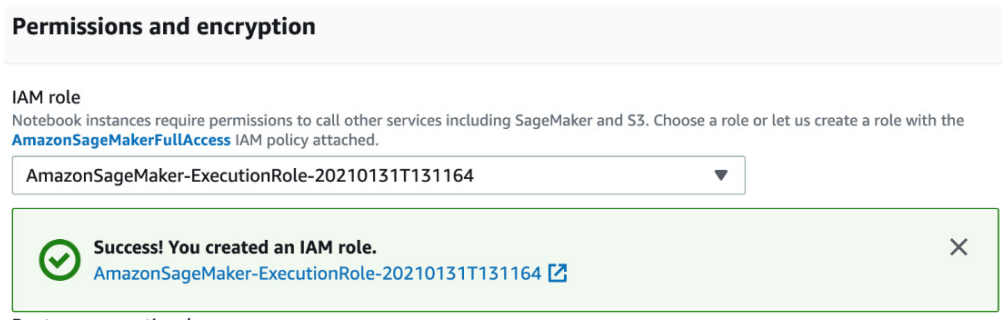


Figure 9.3 – Amazon SageMaker role creation

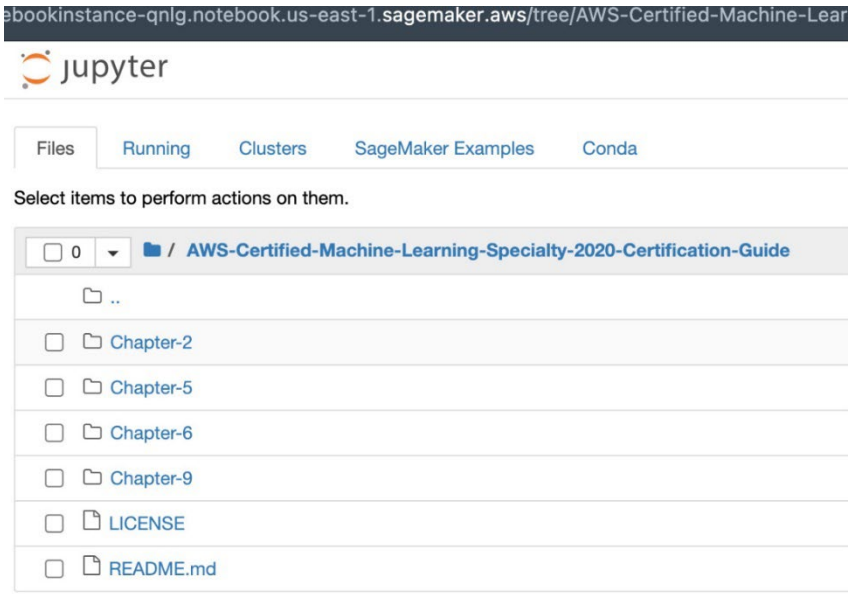


Figure 9.4 – Jupyter Notebook

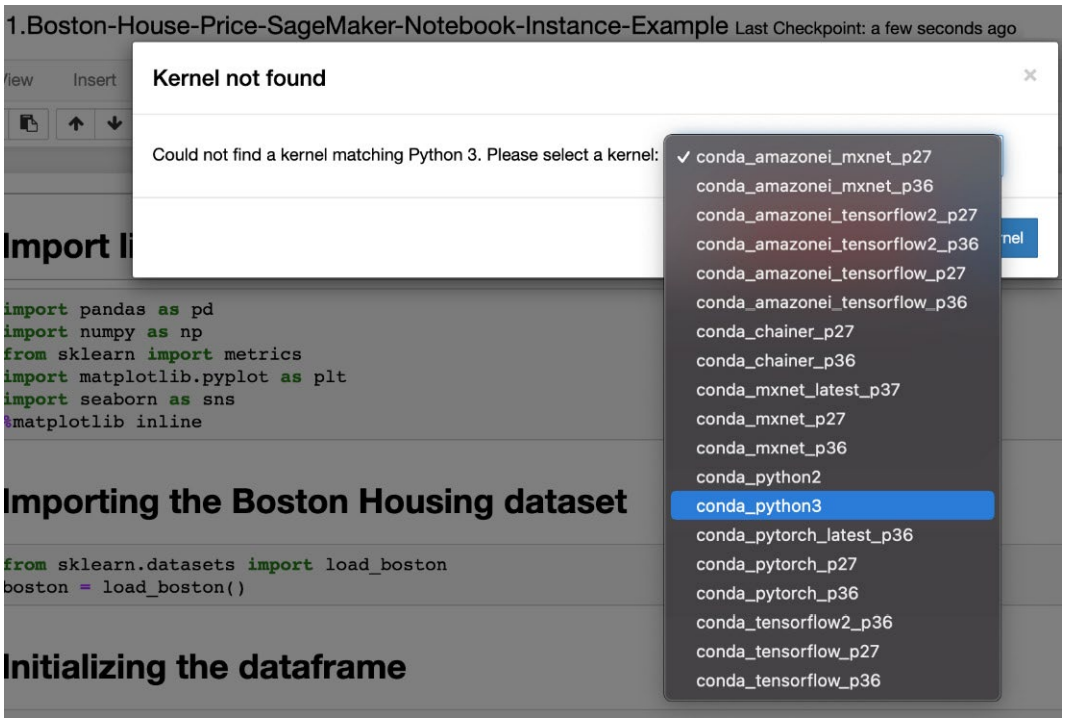


Figure 9.5 – Jupyter Notebook kernel selection

```
models = pd.DataFrame({
    'Model': ['Linear Regression', 'Random Forest', 'XGBoost', 'Support Vector Machines'],
    'R-squared Score': [acc_linreg*100, acc_rf*100, acc_xgb*100, acc_svm*100],
    'Adjusted R-squared Score': [adj_R2_linreg*100, adj_R2_rf*100, adj_R2_xgb*100, adj_R2_svm*100],
    'MAE': [mae_linreg, mae_rf, mae_xgb, mae_svm],
    'MSE': [mse_linreg, mse_rf, mse_xgb, mse_svm],
    'RMSE': [rmse_linreg, rmse_rf, rmse_xgb, rmse_svm]})
models.sort_values(by='R-squared Score', ascending=False)
```

	Model	R-squared Score	Adjusted R-squared Score	MAE	MSE	RMSE
2	XGBoost	85.799520	84.461793	2.530958	14.828152	3.850734
1	Random Forest	82.181317	80.502745	2.500316	18.606282	4.313500
0	Linear Regression	71.218184	68.506853	3.859006	30.053993	5.482152
3	Support Vector Machines	59.001585	55.139415	3.756145	42.810575	6.542979

Figure 9.6 – Comparing the models

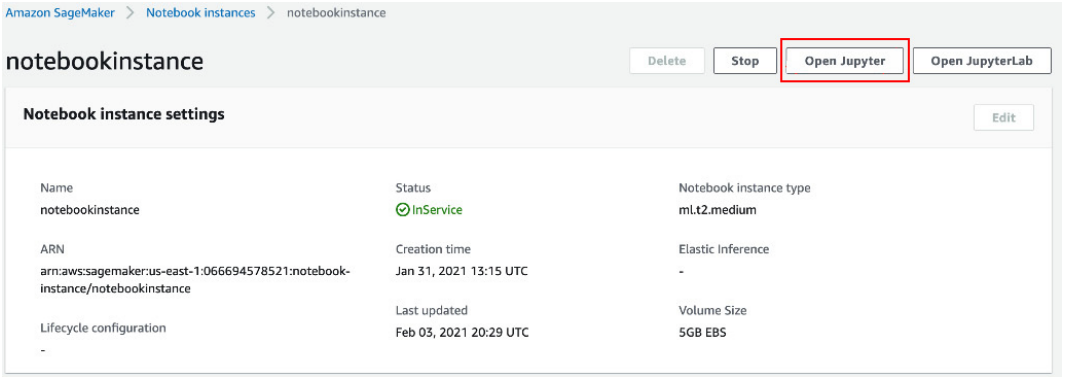


Figure 9.7 – The InService instance

```
hyperparams = {"feature_dim": 54, "k": 10, "sample_size": 200000, "predictor_type": "classifier"}
output_path = f"s3://{bucket}/{prefix}/default_example/output"
knn_estimator = trained_estimator_from_hyperparams(
    s3_train_data, hyperparams, output_path, s3_test_data=s3_test_data
)
```

```
The method get_image_uri has been renamed in sagemaker>=2.
See: https://sagemaker.readthedocs.io/en/stable/v2.html for details.
Defaulting to the only supported framework/algorithm version: 1. Ignoring framework/algorithm version: 1.

2021-02-03 22:00:05 Starting - Starting the training job...
2021-02-03 22:00:30 Starting - Launching requested ML instancesProfilerReport-1612389605: InProgress
.........
2021-02-03 22:01:51 Starting - Preparing the instances for training...
2021-02-03 22:02:32 Downloading - Downloading input data...
2021-02-03 22:02:52 Training - Downloading the training image...
2021-02-03 22:03:33 Training - Training image download completed. Training in progress..Docker entrypoint called with
argument(s): train
Running default environment configuration script
[02/03/2021 22:03:35 INFO 140457495148352] Reading default configuration from /opt/amazon/lib/python2.7/site-package
s/algorithm/resources/default-conf.json: {u'index_metric': u'L2', u'_tuning_objective_metric': u'', u'_num_gpus': u'a
uto', u'_log_level': u'info', u'feature_dim': u'auto', u'faiss_index_ivf_nlists': u'auto', u'epochs': u'1', u'index_t
ype': u'faiss.Flat', u'_faiss_index_nprobe': u'5', u'_kvstore': u'dist_async', u'_num_kv_servers': u'1', u'mini_batch
_size': u'5000'}
[02/03/2021 22:03:35 INFO 140457495148352] Merging with provided configuration from /opt/ml/input/config/hyperparamet
```

Figure 9.8 – The SageMaker fit API call

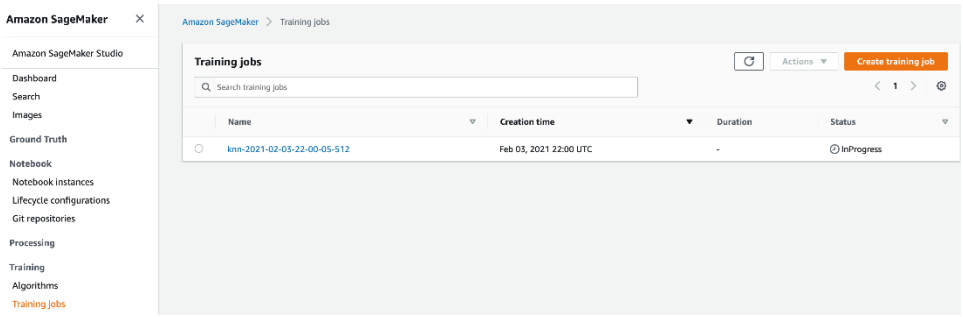


Figure 9.9 – Training jobs

```
import time

instance_type = "ml.t2.medium"
model_name = "knn_%s" % instance_type
endpoint_name = "knn-baba-test-%s" % (str(time.time()).replace(".", "-"))
print("setting up the endpoint..")
predictor = predictor_from_estimator(
    knn_estimator, model_name, instance_type, endpoint_name=endpoint_name
)
```

```
setting up the endpoint..
-
```

Figure 9.10 – Creating the predictor object with endpoint details

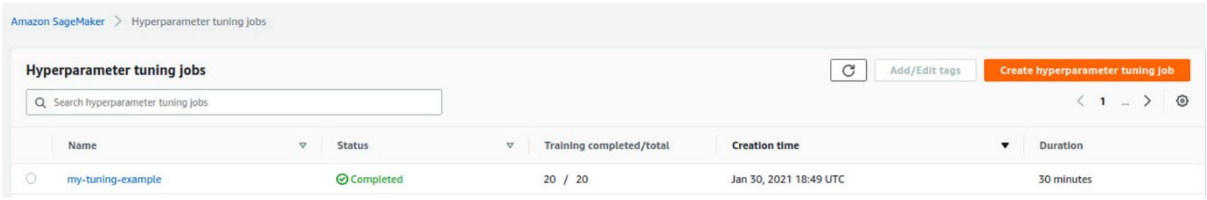


Figure 9.11 – Finding your tuning job

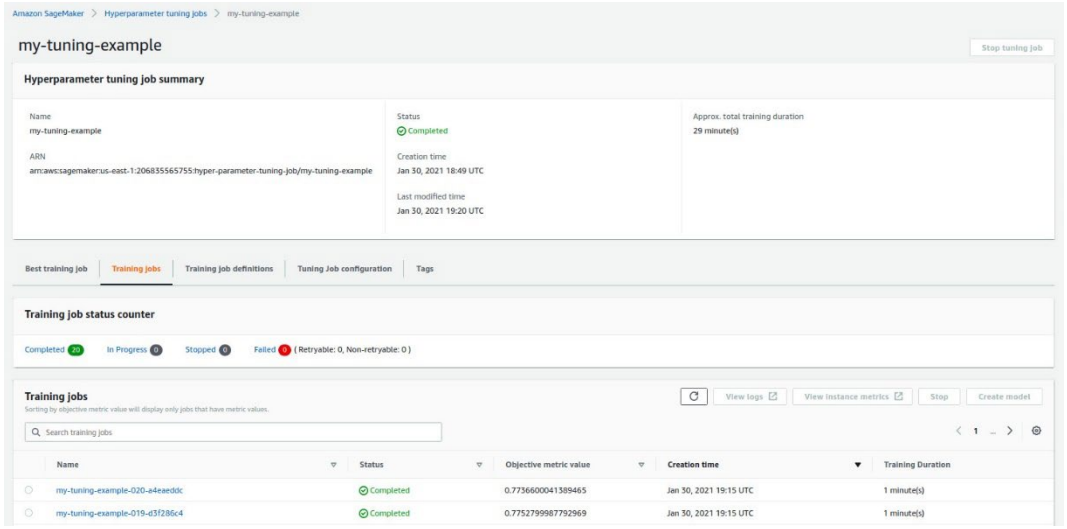


Figure 9.12 – Summary of the training jobs in the tuning process

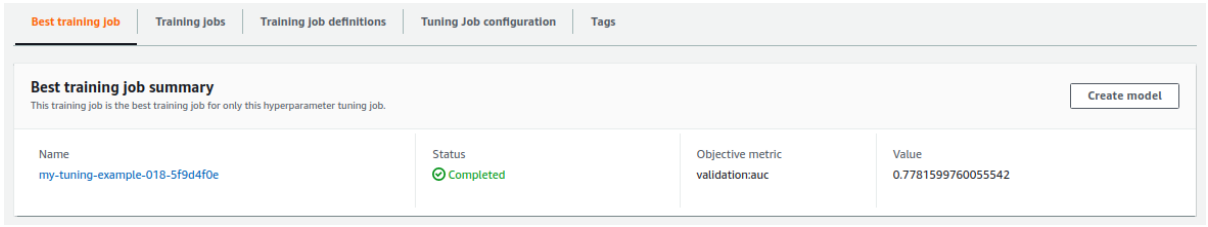


Figure 9.13 – Finding the best set of hyperparameters

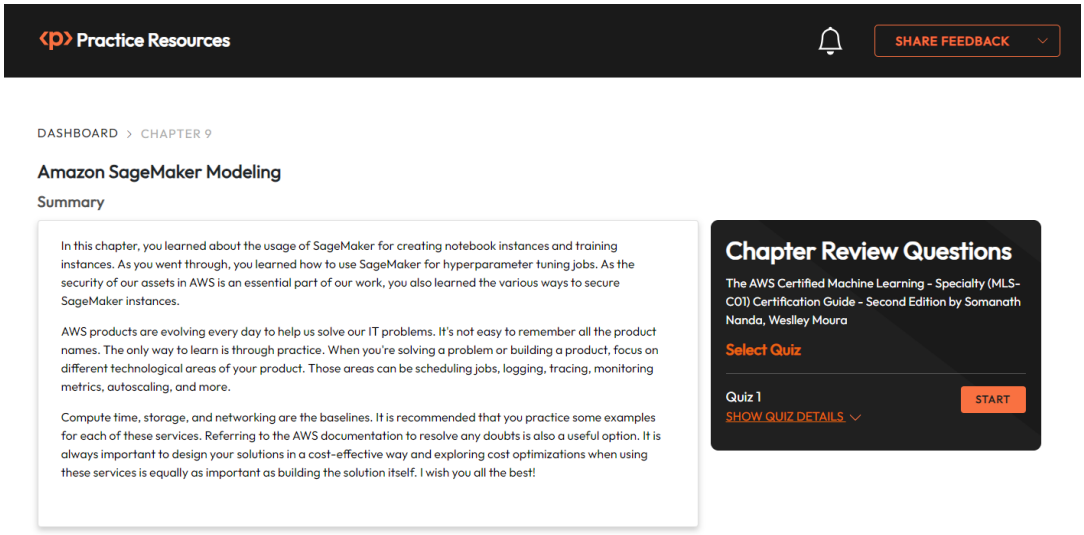


Figure 9.15 – Chapter Review Questions for Chapter 9

Chapter 10: Model Deployment

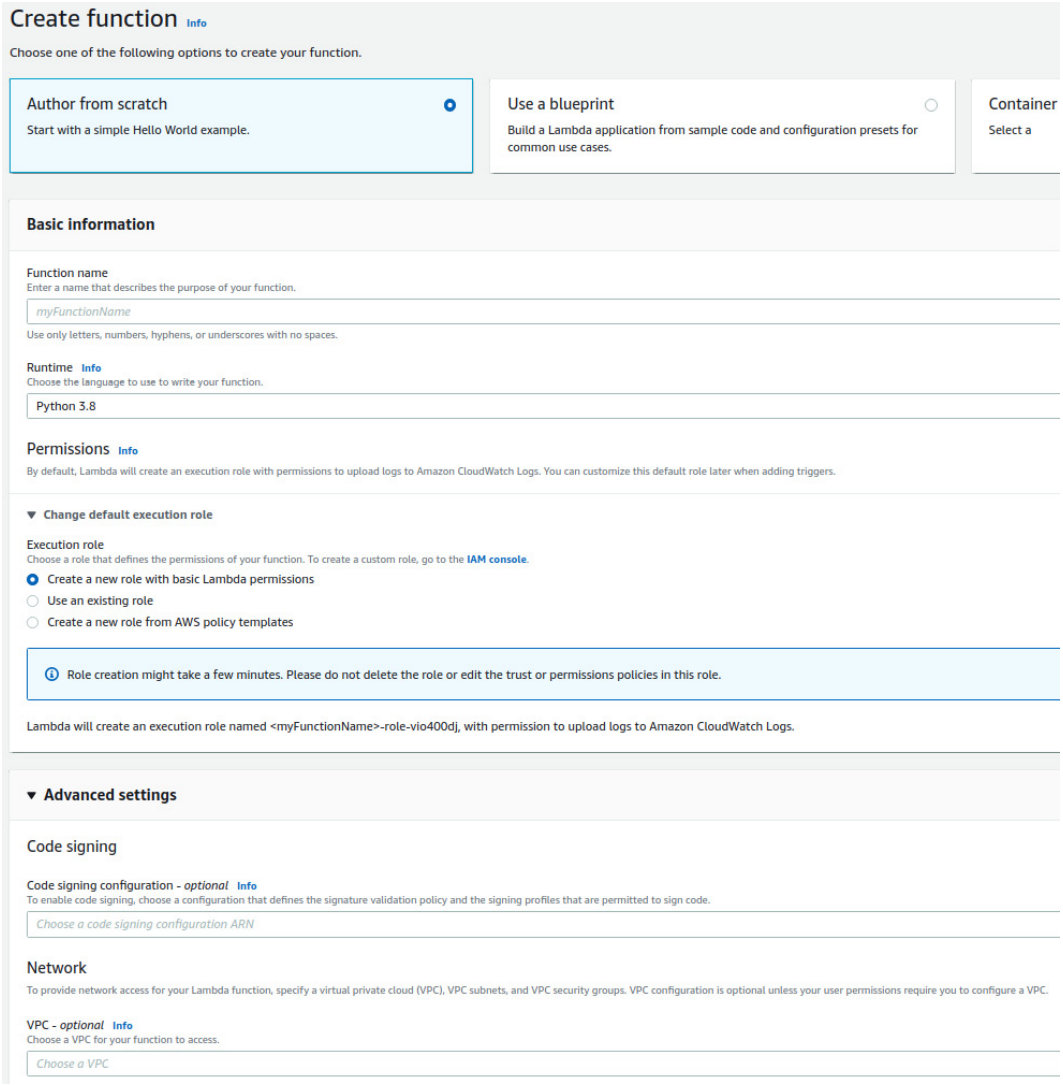


Figure 10.1 – Creating a new Lambda function from the AWS Lambda console

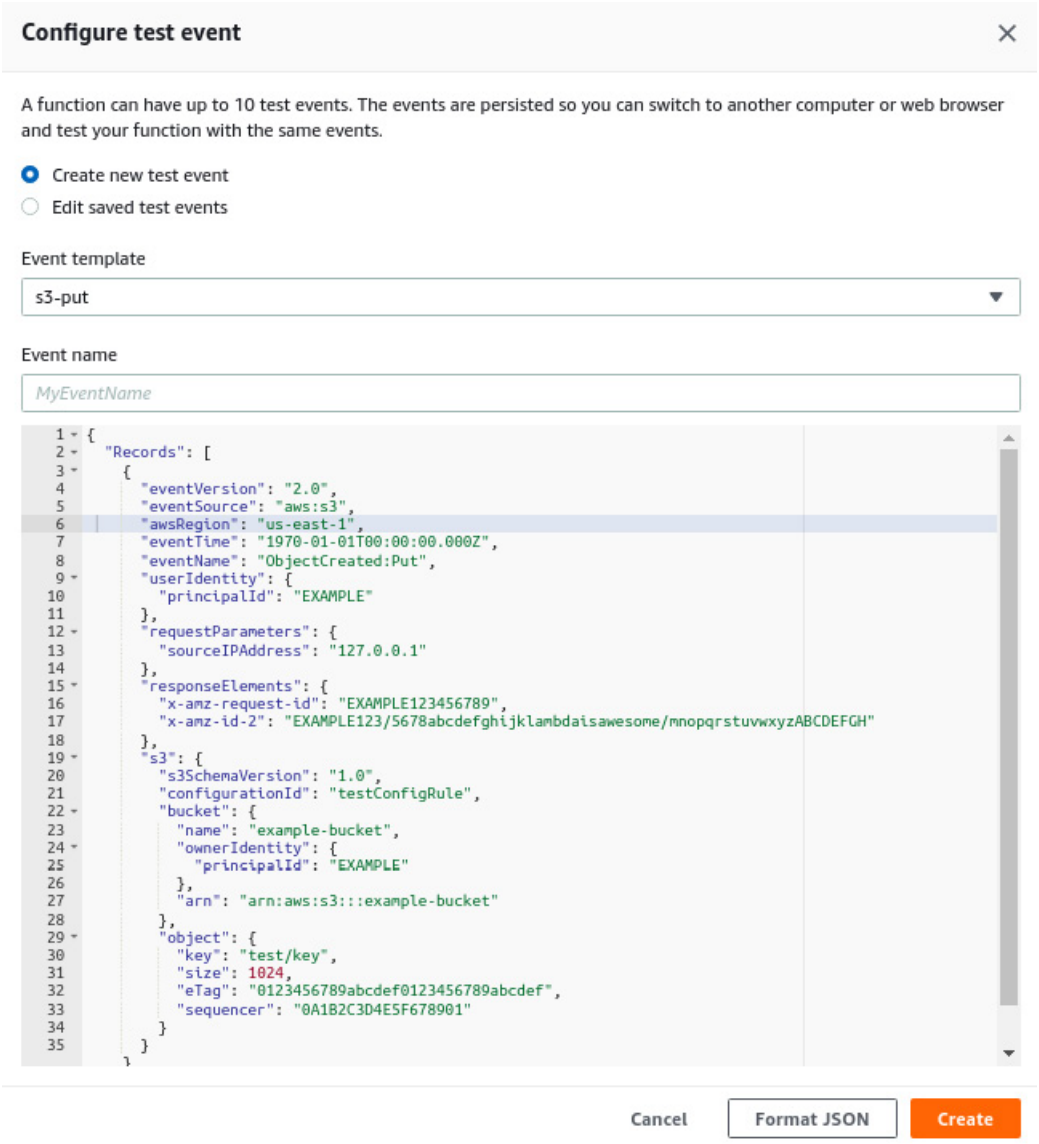


Figure 10.2 – Creating a test event from the Lambda console

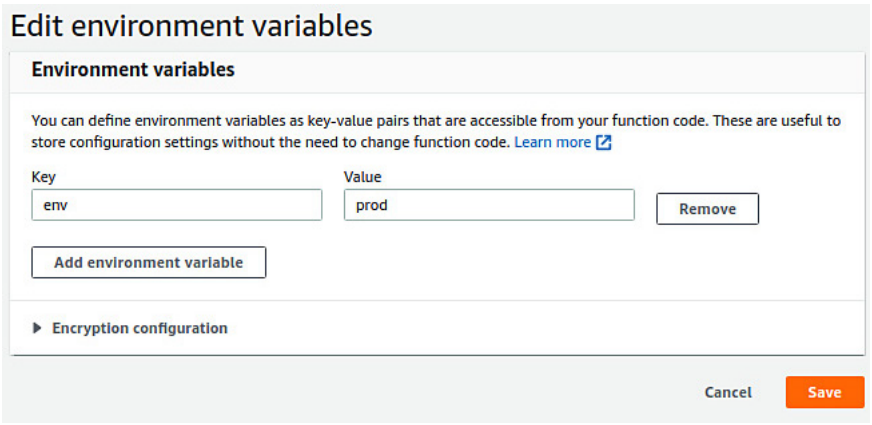


Figure 10.3 – Adding environment variables to a Lambda function

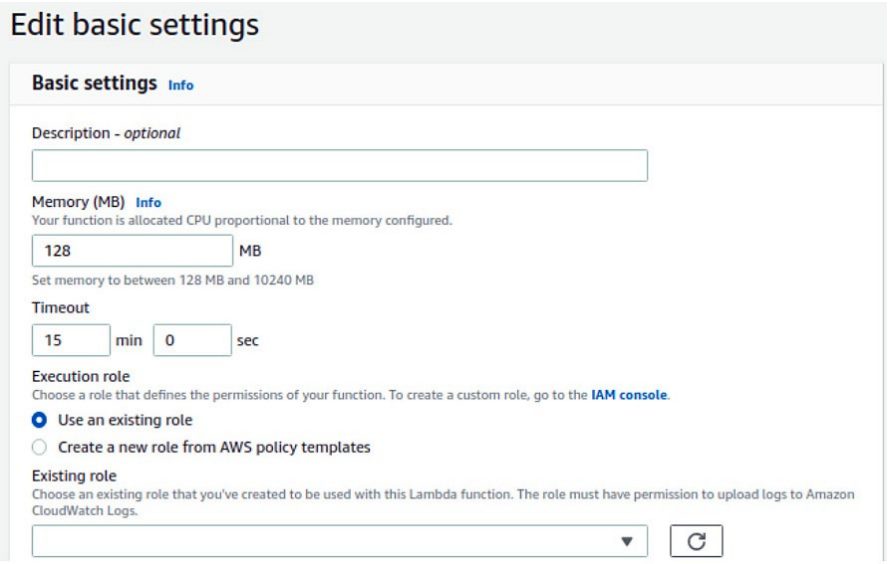


Figure 10.4 – Changing the basic settings of a Lambda function

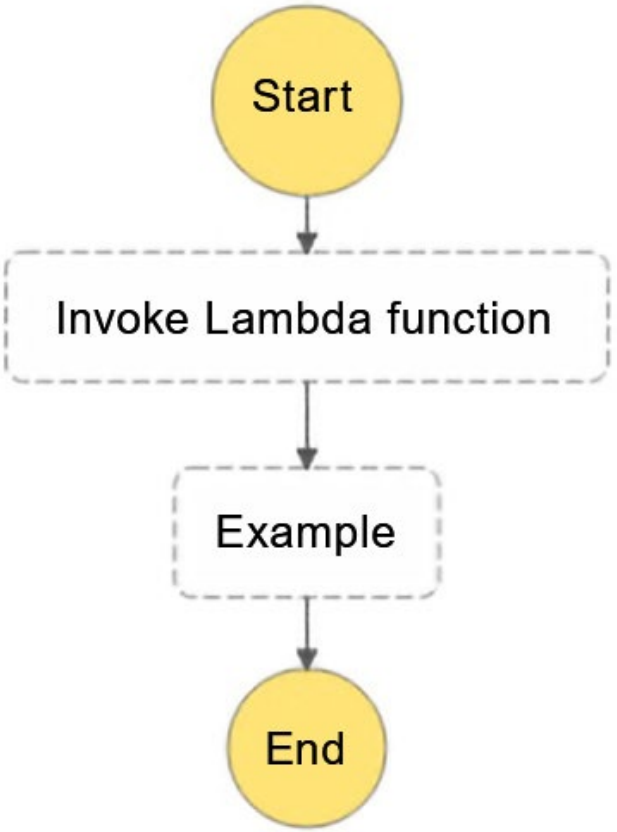


Figure 10.5 – The state machine

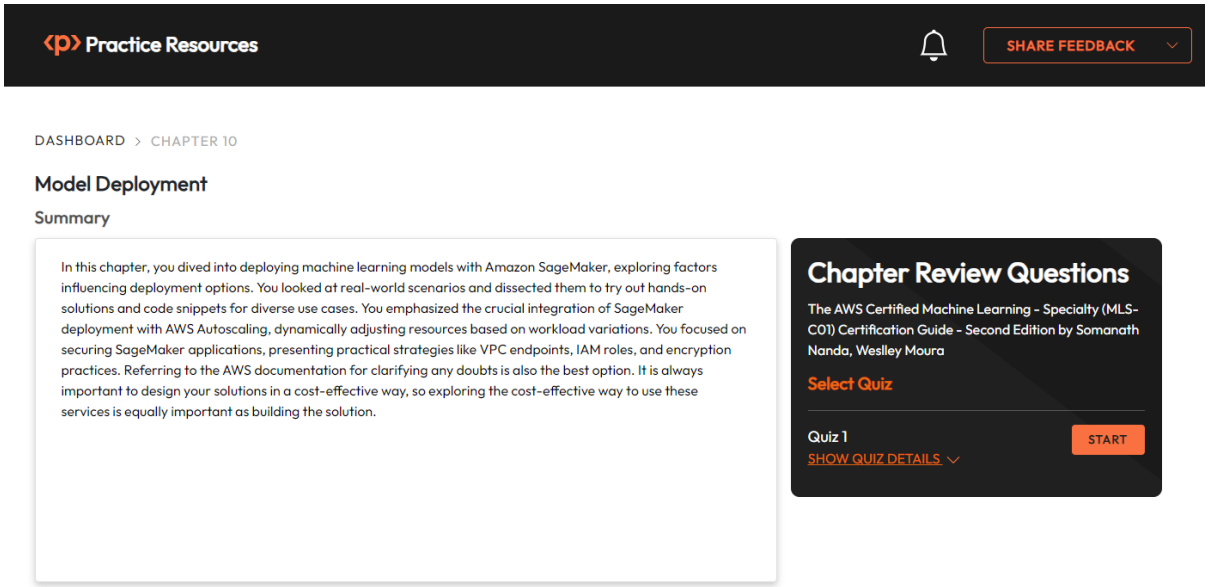


Figure 10.7 – Chapter Review Questions for Chapter 10

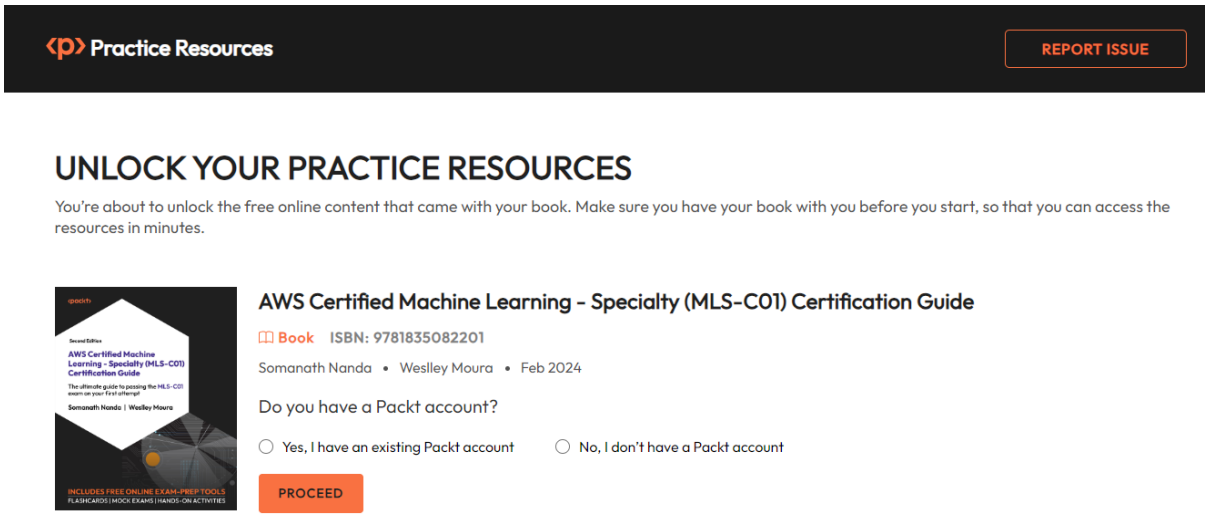


Figure 11.2 – Unlock page for the online practice resources

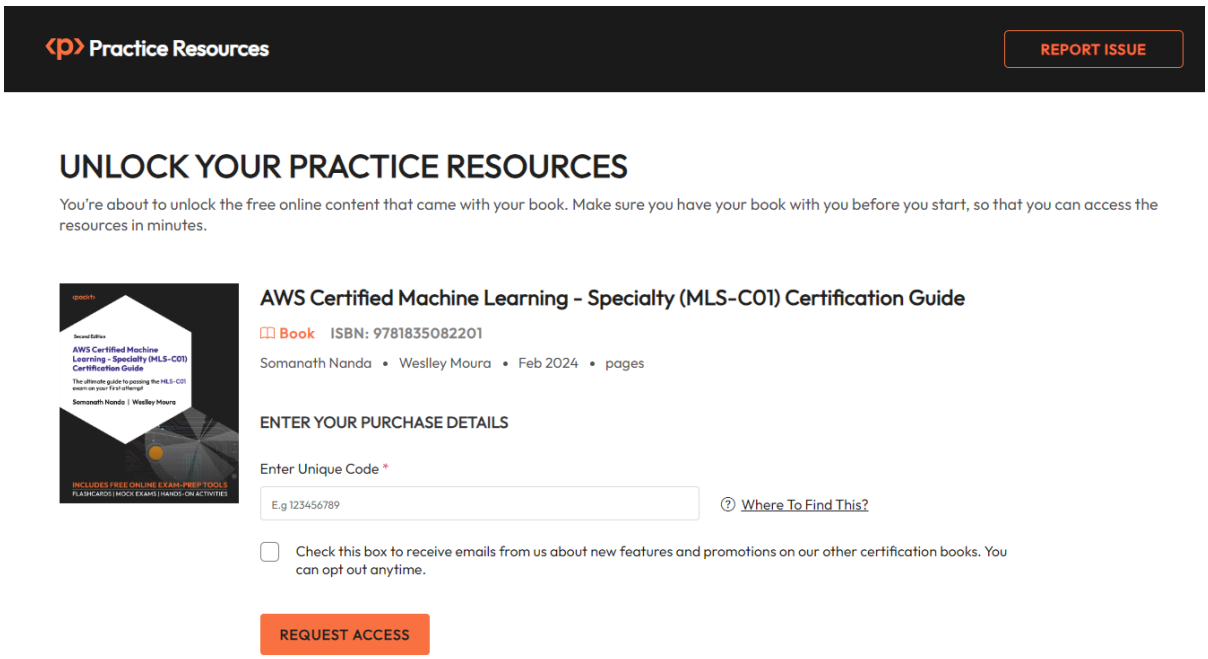


Figure 11.3 – Enter your unique sign-up code to unlock the resources

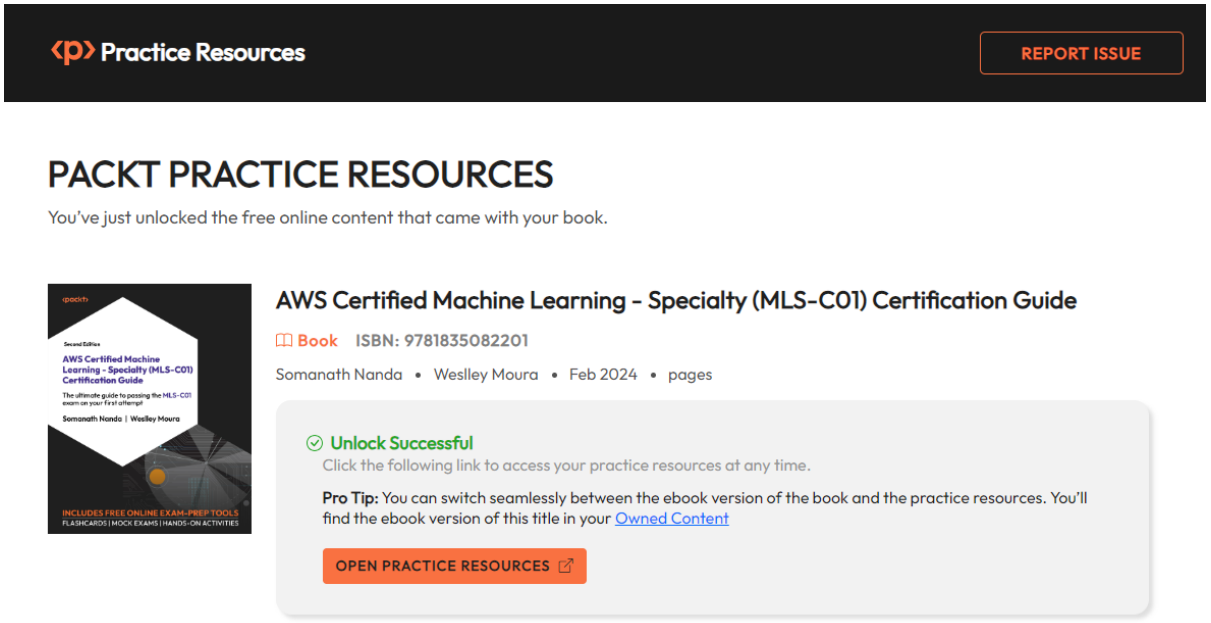


Figure 11.4 – Page that shows up after a successful unlock

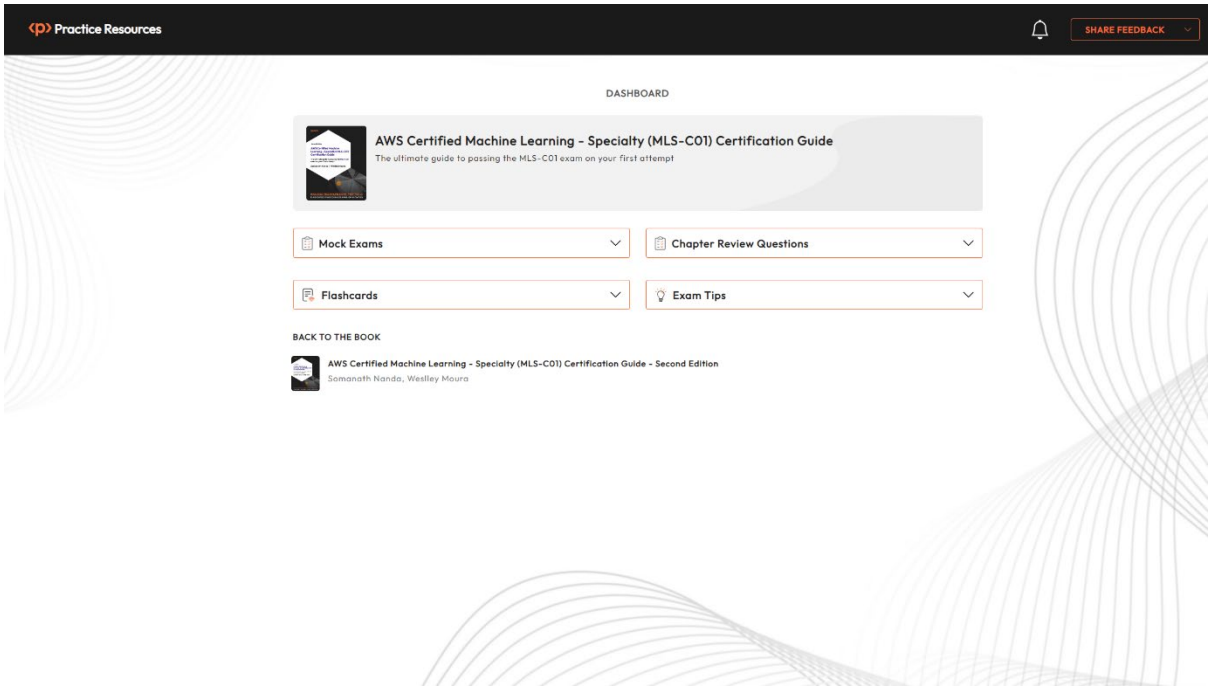


Figure 11.5 – Dashboard page for AZ-900 practice resources

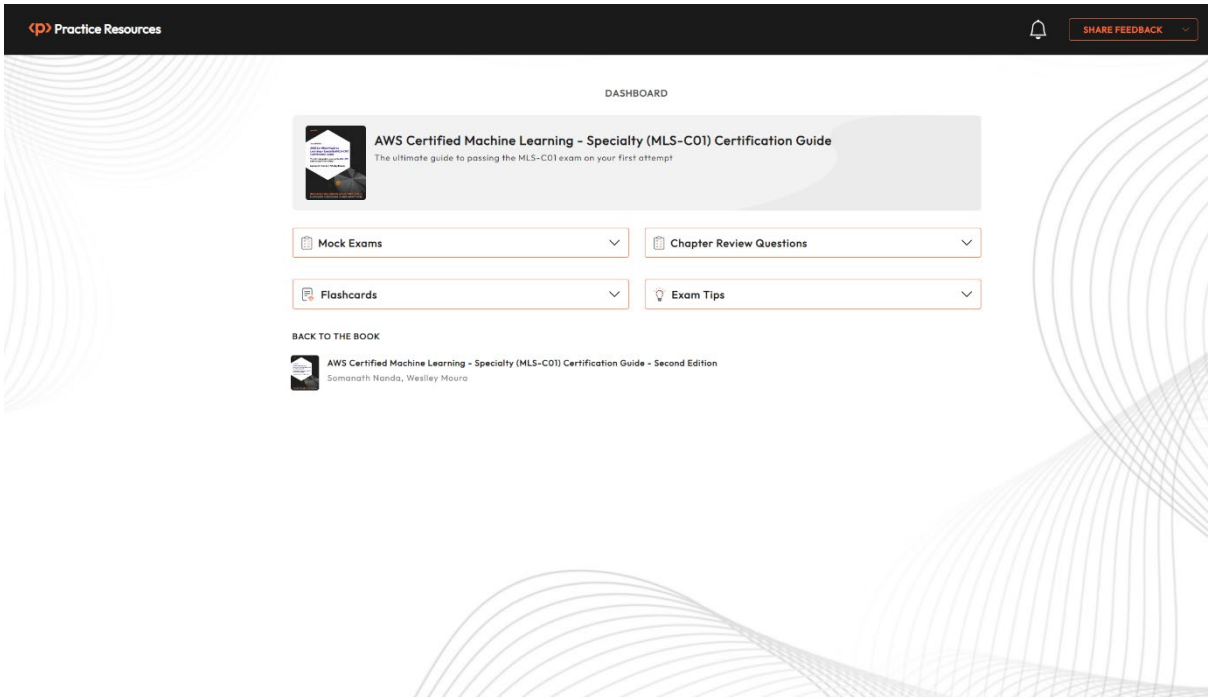


Figure 11.7 – Dashboard interface on a desktop device

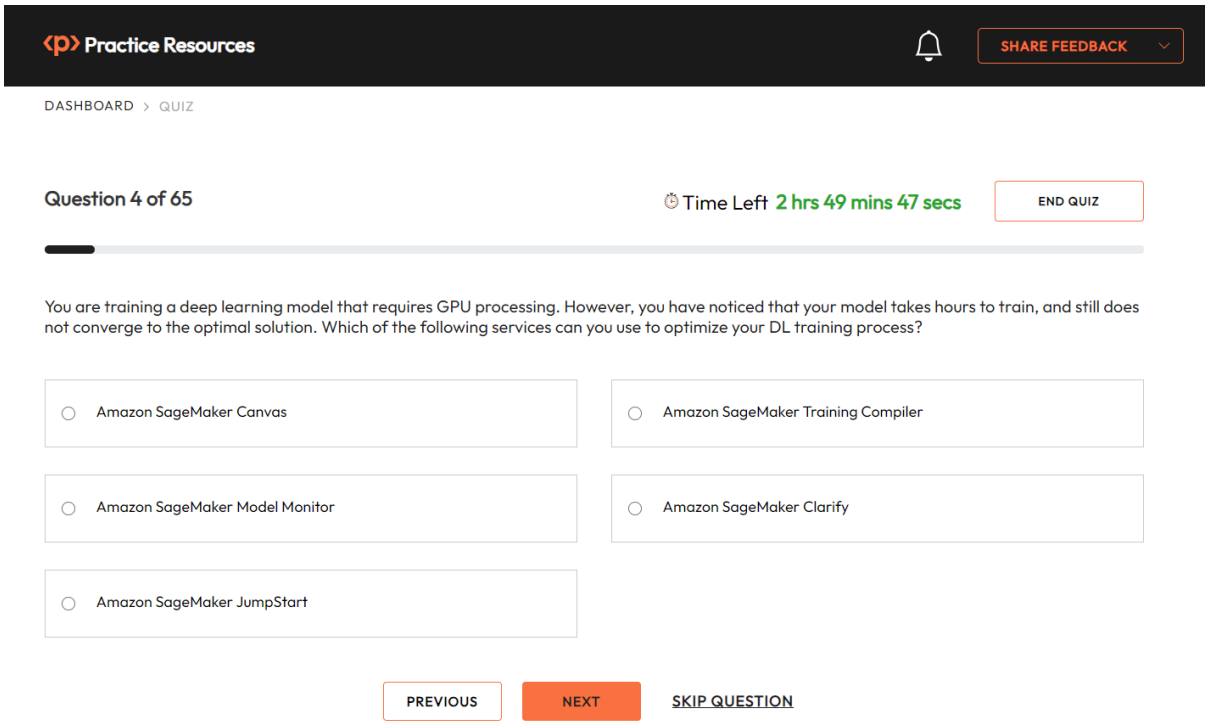


Figure 11.8 – Practice Questions Interface on a desktop device

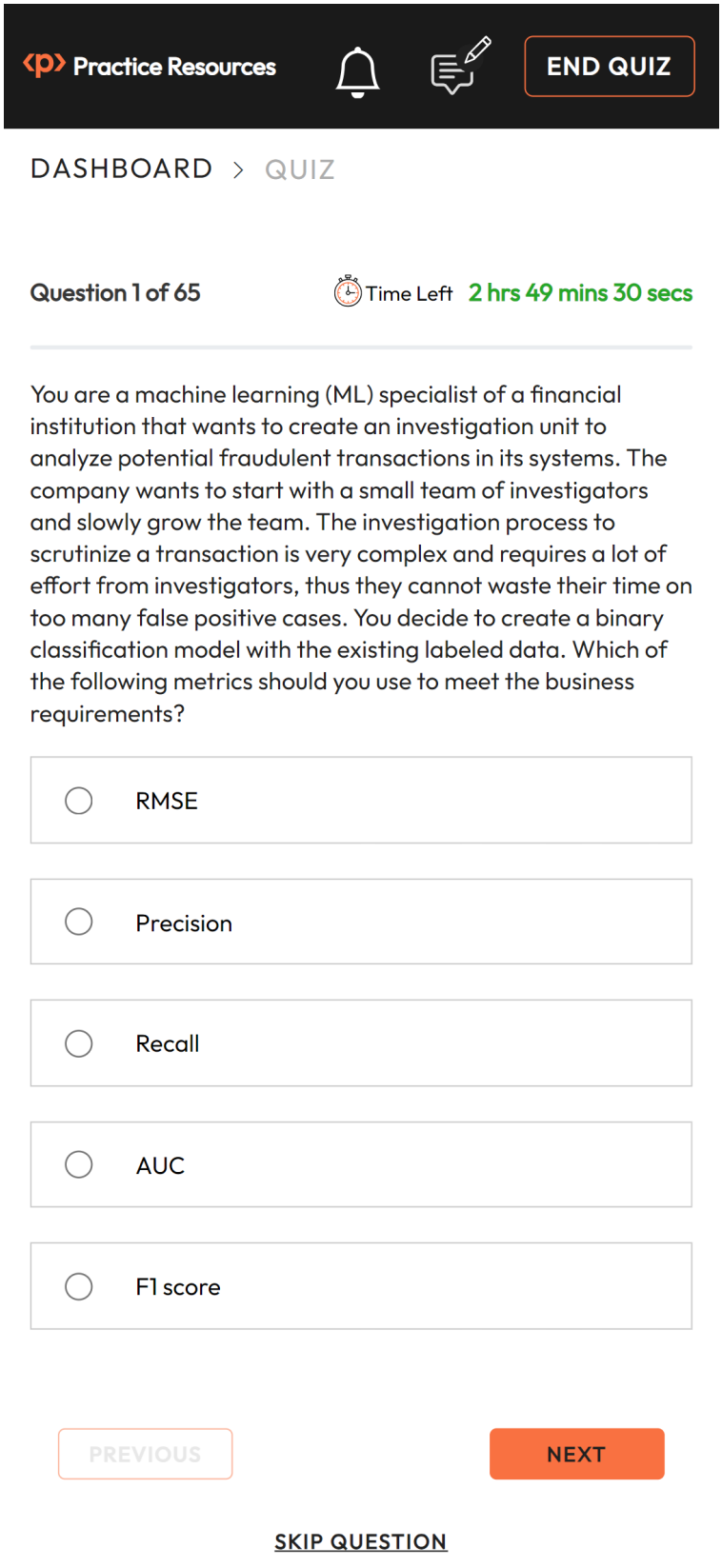


Figure 11.9 – Quiz interface on a mobile device

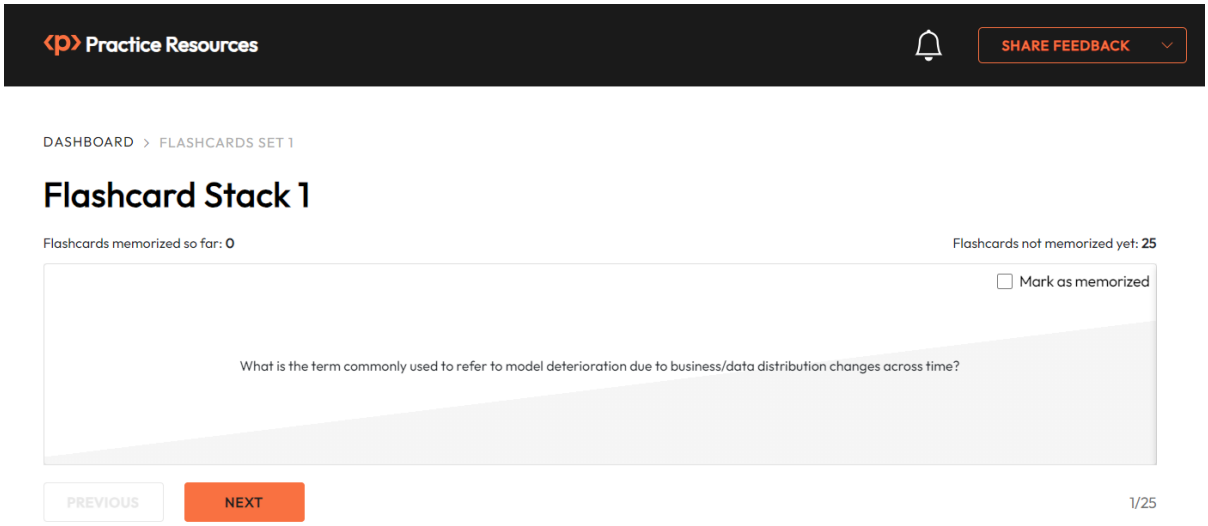


Figure 11.10 – Flashcards interface

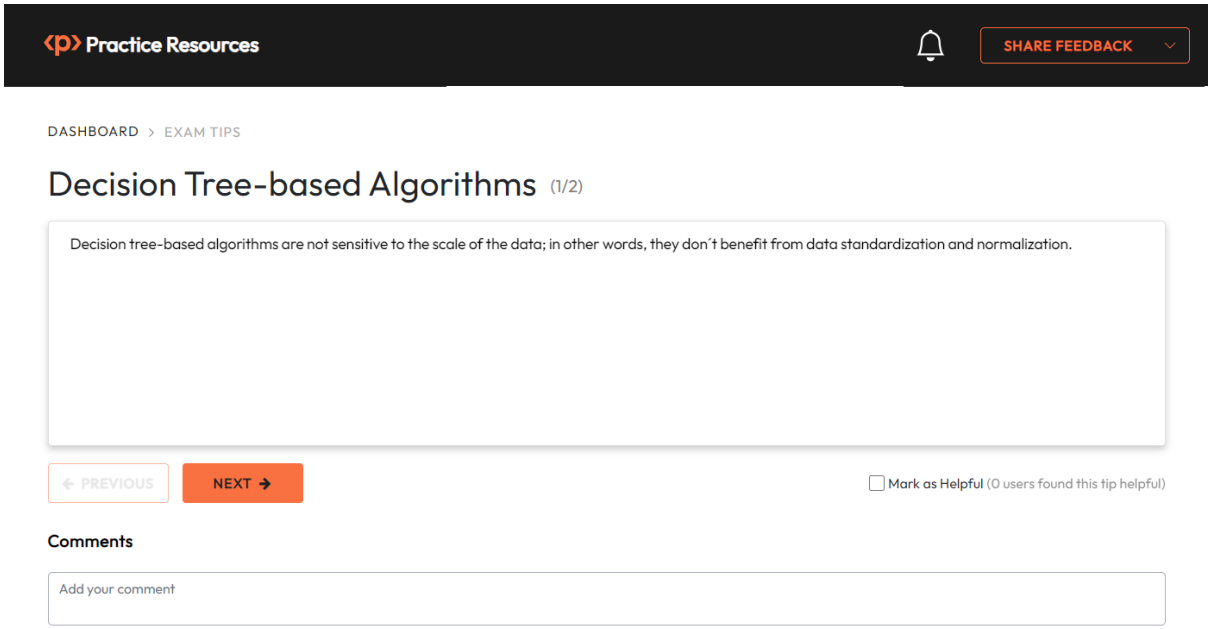


Figure 11.11 – Exam Tips Interface

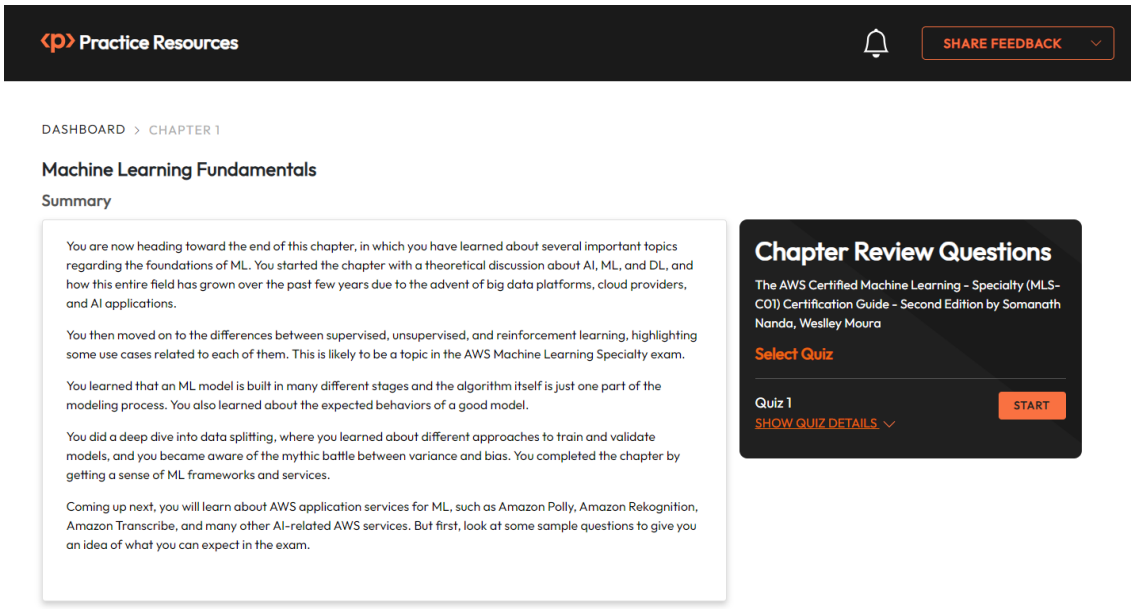


Figure 11.12 – Chapter Review Questions Page

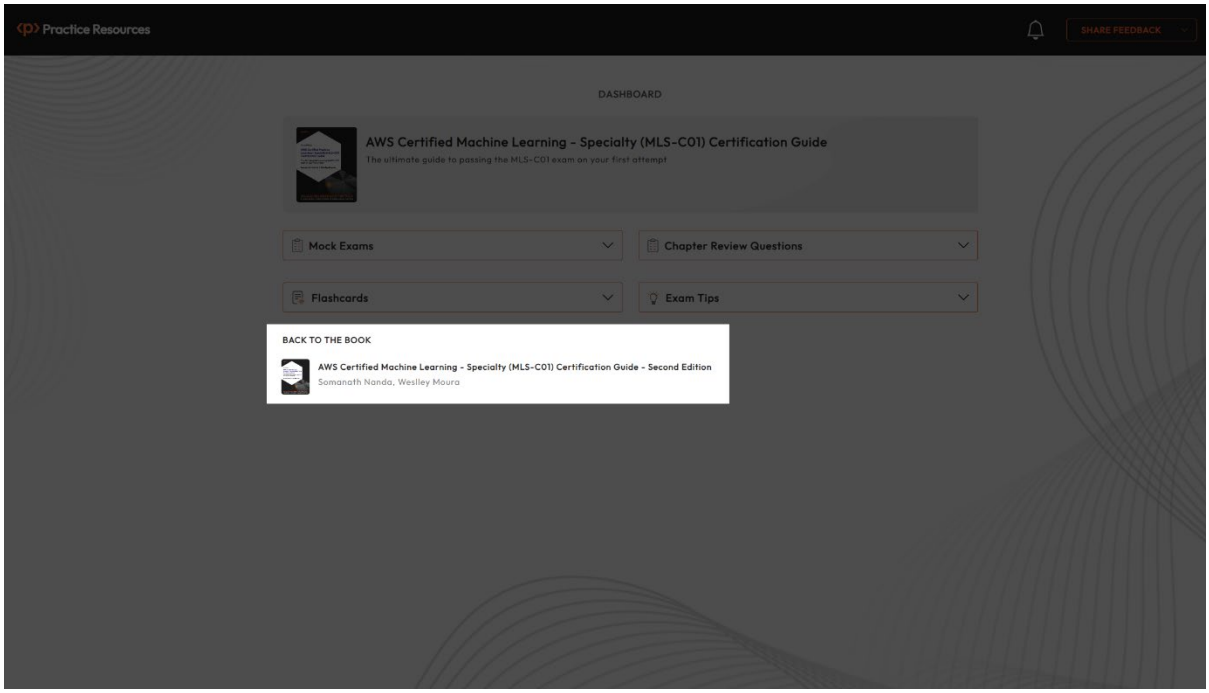


Figure 11.13 – Jump back to the book from the dashboard