Udemy - Warmup Test - Quick Assessment - 2

Warmup Test: Quick Assessment

10 questions | 30 minutes | 70% correct required to pass

This 10-question warmup test should give you a good idea of how prepared you really are for the full practice exam, and for the real one - without investing 3 hours in the process. We chose these questions to be representative of the domains covered by the real exam, and some of the more difficult topics you'll be expected to know on it. If you're surprised by the topics and level of detail you encounter, you know you have more preparation and studying to do.

The AWS Certified Machine Learning Specialty exam goes beyond AWS topics, and tests your knowledge in feature engineering, model tuning, and modeling as well as how deep neural networks work. You need to both have expert-level knowledge of AWS's machine learning services (especially SageMaker), and expert-level knowledge in machine learning and AI in general.

Question 1:

You wish to use a SageMaker notebook within a VPC. SageMaker notebook instances are Internet-enabled, creating a potential security hole in your VPC. How would you use SageMaker within a VPC without opening up Internet access?

- ○ Uncheck the option for Internet access when creating your notebook instance, and it will handle the rest automatically.
- ○ Use IAM to restrict Internet access from the notebook instance.
- ○ No action is required, the VPC will block the notebook instances from accessing the Internet.
- ◉ Disable direct Internet access when specifying the VPC for your notebook instance, and use VPC interface endpoints (PrivateLink) to allow the connections needed to train and host your model. Modify your instance's security group to allow outbound connections for training and hosting.

Question 2:

Your company wishes to monitor social media, and perform sentiment analysis on Tweets to classify them as positive or negative sentiment. You are able to obtain a data set of past Tweets about your company to use as training data for a machine learning system, but they are not classified as positive or negative. How would you build such a system?

- ○ Stream both old and new tweets into an Amazon Elasticsearch Service cluster, and use Elasticsearch machine learning to classify the tweets.
- ○ Use Amazon Machine Learning with a binary classifier to assign positive or negative sentiments to the past Tweets, and use those labels to train a neural network on an EMR cluster.
- ◉ Use SageMaker Ground Truth to label past Tweets as positive or negative, and use those labels to train a neural network on SageMaker.
- ○ Use RANDOM_CUT_FOREST to automatically identify negative tweets as outliers.

Question 3:

Your automatic hyperparameter tuning job in SageMaker is consuming more resources than you would like, and coming at a high cost. What are TWO techniques that might reduce this cost?

- ☑ Use logarithmic scales on your parameter ranges
- ☐ Use linear scales on your parameter ranges
- ☐ Use more concurrency while tuning
- ☑ Use less concurrency while tuning
- ☐ Use inference pipelines

Question 4:

You are training an XGBoost model on SageMaker with millions of rows of training data, and you wish to use Apache Spark to pre-process this data at scale. What is the simplest architecture that achieves this?

- (●) Use sagemaker_pyspark and XGBoostSageMakerEstimator to use Spark to pre-process, train, and host your model using Spark on SageMaker.
- ( ) Use Amazon EMR to pre-process your data using Spark, and then use AWS Data Pipelines to transfer the processed training data to SageMaker
- ( ) Use Amazon EMR to pre-process your data using Spark, and use the same EMR instances to host your SageMaker notebook.
- ( ) Use Sparkmagic to pre-process your data within a SageMaker notebook, transform the resulting Spark DataFrames into RecordIO format, and then use Spark's XGBoost algorithm to train the model.

A large news website needs to produce personalized recommendations for articles to its readers, by training a machine learning model on a daily basis using historical click data. The influx of this data is fairly constant, except during major elections when traffic to the site spikes considerably. Which system would provide the most cost-effective and reliable solution?

- (●) Publish click data into Amazon S3 using Kinesis Firehose, and process the data nightly using Apache Spark and MLLib using spot instances in an EMR cluster. Publish the model's results to DynamoDB for producing recommendations in real-time.
- ( ) Publish click data into Amazon S3 using Kinesis Firehose, and process the data nightly using Apache Spark and MLLib using reserved instances in an EMR cluster. Publish the model's results to DynamoDB for producing recommendations in real-time.
- ( ) Publish click data into Amazon S3 using Kinesis Streams, and process the data in real time using Splunk on an EMR cluster with spot instances added as needed. Publish the model's results to DynamoDB for producing recommendations in real-time.
- ( ) Publish click data into Amazon Elasticsearch using Kinesis Firehose, and query the Elasticsearch data to produce recommendations in real-time.

Question 6:

You are developing a computer vision system that can classify every pixel in an image based on its image type, such as people, buildings, roadways, signs, and vehicles. Which SageMaker algorithm would provide you with the best starting point for this problem?

- ○ Object Detection
- ○ Rekognition
- ◉ Semantic Segmentation
- ○ Object2Vec

Question 7:

You are developing an autonomous vehicle that must classify images of street signs with extremely low latency, processing thousands of images per second. What AWS-based architecture would best meet this need?

- ◉ Develop your classifier with TensorFlow, and compile it for an NVIDIA Jetson edge device using SageMaker Neo, and run it on the edge with IoT GreenGrass.
- ○ Use Amazon Rekognition in edge mode
- ○ Develop your classifier using SageMaker Object Detection, and use Elastic Inference to accelerate the model's endpoints called over the air from the vehicle.
- ○ Use Amazon Rekognition on AWS DeepLens to identify specific street signs in a self-contained manner.

Question 8:

A system designed to classify financial transactions into fraudulent and non-fraudulent transactions results in the confusion matrix below. What is the recall of this model?

	Actual Positives	Actual Negatives
Predicted Positives	90	45
Predicted Negatives	10	20

- ○ 50%
- ● 90%
- ○ 74%
- ○ 66.67%

Question 9:

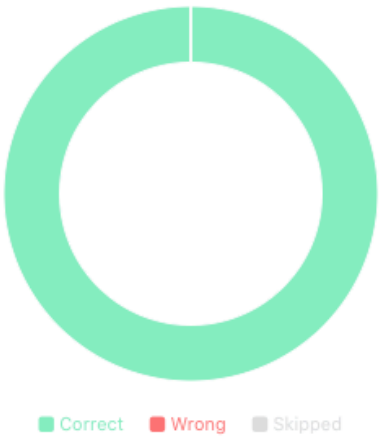
You are developing a machine learning model to predict house sale prices based on features of a house. 10% of the houses in your training data are missing the number of square feet in the home. Your training data set is not very large. Which technique would allow you to train your model while achieving the highest accuracy?

- ○ Drop all rows that contain missing data
- ● Impute the missing square footage values using kNN
- ○ Impute the missing values using the mean square footage of all homes
- ○ Impute the missing values using deep learning, based on other features such as number of bedrooms

Question 10:

After training a deep neural network over 100 epochs, it achieved high accuracy on your training data, but lower accuracy on your test data, suggesting the resulting model is overfitting. What are TWO techniques that may help resolve this problem?

- [ ] Employ gradient checking
- [x] Use early stopping
- [x] Use dropout regularization
- [ ] Use more features in the training data
- [ ] Use more layers in the network



Attempt 1: Passed! (70% required to pass)

100% correct (10/10)

4 minutes

October 25, 2020 10:12 AM

Review questions