

Special Offer | Flat 15% OFF on All Courses | Use Coupon - WHIZSITE15

[Home](#) > [My Courses](#) > [AWS Certified Machine Learning Specialty](#) > [Core ML Concepts](#) > [Report](#)

Search Courses



Core ML Concepts

Completed on 28-January-2021



Attempt

01



Marks Obtained

1 / 10



Your score

10%



Time Taken

00 H 00 M 11 S



Result

Failed

Domains wise Quiz Performance Report

Join us on [Slack community](#)

No	Domain	Total Question	Correct	Incorrect	Unattempted	Marked as Review
1	Exploratory Data Analysis	2	0	0	2	0
2	Modeling	8	1	0	7	0
Total	All Domain	10	1	0	9	0

Review the Answers

Sorting by

All

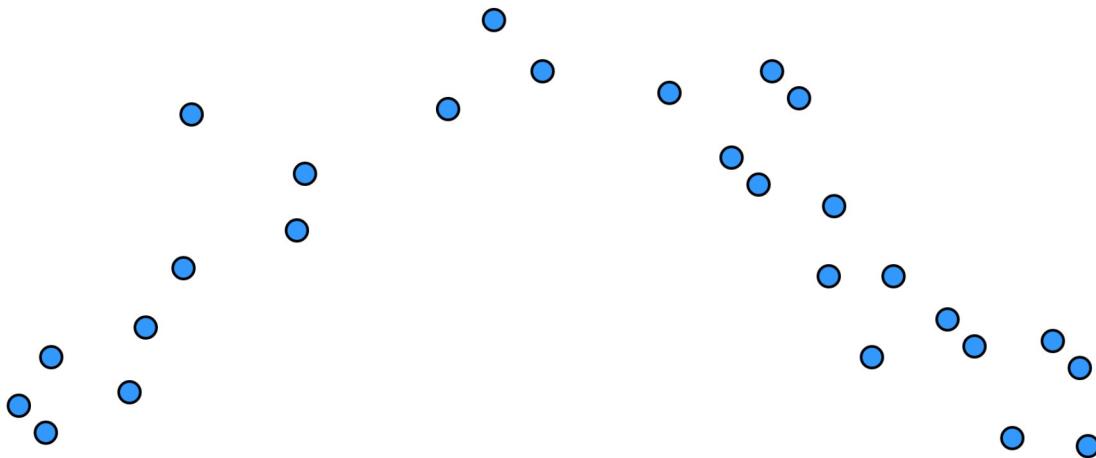
Question 1

Unattempted

Domain :Exploratory Data Analysis

You are a machine learning specialist working for an oil refinery company. Your team is working on a machine learning problem where you need to determine the relationship between oil well depth and oil well production. In order to select the appropriate machine learning algorithm to use to attempt to solve the oil well production problem, you need to gain a better understanding of your data. For example, what is the correlation between your oil well depth data and your oil well production data?

When you examine your data visually using the python matplotlib library, you find that your data has what looks like a non-Gaussian distribution of oil well depth and oil well production:



Which correlation coefficient would you use to best summarize the strength of the correlation between your oil well depth and oil well production?

- A. Covariance correlation coefficient
- B. Pearson's correlation coefficient
- C. Spearman's correlation coefficient
- D. Polychoric correlation coefficient

Explanation:

Answer: C

Option A is incorrect. Covariance is used when you have a Gaussian relationship between your variables.

Option B is incorrect. Pearson's correlation coefficient is also used when you have a Gaussian relationship between your variables.

Option C is correct. Spearman's correlation coefficient is used when you have a non-Gaussian relationship between your variables.

Option D is incorrect. The polychoric correlation coefficient is used to understand the

relationship of variables gathered via surveys such as personality tests and surveys that use rating scales.

Reference:

Please see the Machine Learning Mastery page titled **How to Calculate Correlation Between Variables in Python** (<https://machinelearningmastery.com/how-to-use-correlation-to-understand-the-relationship-between-variables/>), the Wikipedia page titled **Correlation coefficient** (https://en.wikipedia.org/wiki/Correlation_coefficient), the the Wikipedia page titled **Polychoric correlation** (https://en.wikipedia.org/wiki/Polychoric_correlation), and the Medium article titled **What are Covariance and Correlation coefficients and their significance?** (<https://towardsdatascience.com/covariance-and-correlation-321fdacab168>)

[Ask our Experts](#)[Rate this Question?](#)  [View Queries](#)[open ▾](#)**Question 2****Unattempted****Domain :Exploratory Data Analysis**

You are a machine learning specialist working for a clothing manufacturer. You have been tasked with building a machine learning model to determine the return on investment (ROI) for advertising a specific clothing line on social media based on the labeled data of past social media campaigns for similar clothing lines.

You decide to run a Pearson's correlation coefficient to better understand your data correlation. When you calculate your Pearson's correlation coefficient of social media advertising ROI you get a value of 0.35. What conclusions can you draw from this result?

- A. **There is a favorable relationship between your past social media advertising and corresponding campaign ROI.**
- B. **There is an unfavorable relationship between your past social media advertising and corresponding campaign ROI.**
- C. **There is no correlation between past social media advertising and the associated ROI.**
- D. **You cannot declare a notable correlation with confidence based on the resulting coefficient.** 

Explanation:**Answer: D**

Option A is incorrect. Your coefficient value is not high enough to indicate a positive

relationship. For a Pearson's correlation coefficient to indicate a notable correlation, the coefficient value should be above 0.5 for a positive correlation, or below -0.5 for a negative correlation. Your score is 0.35, which falls into the indeterminate range.

Option B is incorrect. Your coefficient value is not low enough to indicate a negative relationship. For a Pearson's correlation coefficient to indicate a notable correlation, the coefficient value should be above 0.5 for a positive correlation, or below -0.5 for a negative correlation. Your score is 0.35, which falls into the indeterminate range.

Option C is incorrect. A coefficient value of 0 or close to 0 indicates no correlation. Your value of 0.35 is not close enough to 0 to indicate no correlation.

Option D is correct. Your coefficient falls into the indeterminate range. For a Pearson's correlation coefficient to indicate a notable correlation, the coefficient value should be above 0.5 for a positive correlation, or below -0.5 for a negative correlation. Your score is 0.35, which falls into the indeterminate range.

Reference:

Please see the Machine Learning Mastery page titled **How to Calculate Correlation Between Variables in Python** (<https://machinelearningmastery.com/how-to-use-correlation-to-understand-the-relationship-between-variables/>), the Wikipedia page titled **Correlation coefficient** (https://en.wikipedia.org/wiki/Correlation_coefficient), and the Medium article titled **What are Covariance and Correlation coefficients and their significance?** (<https://towardsdatascience.com/covariance-and-correlation-321fdacab168>)

Ask our Experts

Rate this Question?  

View Queries

open ▾

Question 3

Unattempted

Domain :Modeling

You are a machine learning specialist working for the social media software development division of your company. The social media features of your web applications allow users to post text messages and pictures about their experiences with your company's products. You need to be able to quickly block posts that contain inappropriate words. You have defined a vocabulary of words deemed inappropriate for your site.

Which algorithm is best suited to your task?

- A. Multinomial Naive Bayes
- B. Bernoulli Naive Bayes 
- C. Gaussian Naive Bayes

D. Polychoric Naive Bayes

Explanation:

Answer: B

Option A is incorrect. The Multinomial Naive Bayes algorithm is best suited for document classification tasks where you wish to know the frequency of a given word from your vocabulary in your observed text. You need to know whether a word from your vocabulary appears in the given post text or not.

Option B is correct. The Bernoulli Naive Bayes algorithm is used in document classification tasks where you wish to know whether a word from your vocabulary appears in your observed text or not. This is exactly what you are trying to accomplish, you need to know whether a word from your vocabulary of inappropriate words appears in the given post text or not.

Option C is incorrect. The Gaussian Naive Bayes algorithm works continuous values in your observations, not discrete values. Your classification problem uses discrete data, the occurrence of a word or not.

Option D is incorrect. There is no Polychoric Naive Bayes algorithm.

Reference:

Please see the DatumBox page titled **Machine Learning Blog & Software Development News** (<http://blog.datumbox.com/machine-learning-tutorial-the-naive-bayes-text-classifier/>), the SebastianRaschka page titled **Naive Bayes and Text Classification – Introduction and Theory** (http://sebastianraschka.com/Articles/2014_naive_bayes_1.html#3_3_multivariate), the Medium page titled **Naive Bayes Classifier** (<https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c>), the Packt page titled **Machine Learning Algorithms: Implementing Naive Bayes with Spark MLLib** (<https://hub.packtpub.com/machine-learning-algorithms-naive-bayes-with-spark-mllib/>), the Wikipedia article page titled **Naive Bayes classifier** (https://en.wikipedia.org/wiki/Naive_Bayes_classifier)

Ask our Experts

Rate this Question?  

View Queries

open ▾

Question 4

Unattempted

Domain :Modeling

You are a machine learning specialist working for a government agency that uses a series of web application forms to gather citizen data for census purposes. You have been tasked with

finding novel user entries as they are entered by your citizens. Where a novel user entry is defined as an outlier compared to the established set of citizen entries in your datastore. You have cleaned your citizen datastore to remove any existing outliers. You now need to build a model to determine whether new entries on your web application are novel. Which algorithm best fits these requirements?

- A. Multinomial Naive Bayes
- B. Bernoulli Naive Bayes
- C. Principal Component Analysis
- D. Support Vector Machine 

Explanation:

Answer: D

Option A is incorrect. The Multinomial Naive Bayes algorithm is best suited for classification tasks where you wish to know the frequency of a given observation. You are trying to determine whether you have a novel observation.

Option B is incorrect. The Bernoulli Naive Bayes algorithm is used in classification tasks where you wish to know whether a known class appears in your observation. You are trying to determine whether you have a novel observation.

Option C is incorrect. The Principal Component Analysis algorithm is used to reduce feature dimensionality. You are trying to determine whether you have a novel observation.

Option D is correct. The Support Vector Machine algorithm can be used when your training data has no outliers and you want to detect whether a new observation is a novel entry.

Reference:

Please see the SciKit Learn page titled **1.4. Support Vector Machines** (<https://scikit-learn.org/stable/modules/svm.html>), the SciKit Learn page titled **2.7. Novelty and Outlier Detection** (https://scikit-learn.org/stable/modules/outlier_detection.html#outlier-detection), and the Amazon SageMaker developer guide titled **Principal Component Analysis (PCA) Algorithm** (<https://docs.aws.amazon.com/sagemaker/latest/dg/pca.html>)

Ask our Experts

Rate this Question?  

View Queries

open ▾

Question 5

Unattempted

Domain :Modeling

You are a machine learning specialist working for a translation service company. Your company offers several mobile applications used for translation on smartphones and tablets. As a new feature of one of your translation apps, your company is offering a feature to generate handwritten notes from spoken text.

Which algorithm is the best choice for your new feature?

- A. Long Short-Term Memory (LSTM) 
- B. Convolutional Neural Network
- C. Multilayer Perceptron
- D. Support Vector Machine

Explanation:

Answer: A

Option A is correct. The Long Short-Term Memory (LSTM) can work with sequences of spoken language and can be used to generate sequenced output such as handwritten text.

Option B is incorrect. Convolutional Neural Networks are primarily used to work with image data. You are working with sound data, spoken text.

Option C is incorrect. The Multilayer Perceptron algorithm is used primarily for classification predictions and regression predictions. Your problem to solve is to convert spoken text to handwritten text.

Option D is incorrect. The Support Vector Machine algorithm is primarily used for classification, regression, and anomaly detection. Your problem to solve is to convert spoken text to handwritten text.

Reference:

Please see the Machine Learning Mastery article titled **When to Use MLP, CNN, and RNN Neural Networks** (<https://machinelearningmastery.com/when-to-use-mlp-cnn-and-rnn-neural-networks/>), the SciKit Learn page titled **1.4. Support Vector Machines** (<https://scikit-learn.org/stable/modules/svm.html>), and the Wikipedia page titled **Long short-term memory** ([https://en.wikipedia.org/wiki/Long_short-term_memory#:~:text=Long%20short%2Dterm%20memory%20\(LSTM\)%20is%20an%20artificial%20text=LSTM%20networks%20are%20well%2Ds suited,events%20in%20a%20time%20series.](https://en.wikipedia.org/wiki/Long_short-term_memory#:~:text=Long%20short%2Dterm%20memory%20(LSTM)%20is%20an%20artificial%20text=LSTM%20networks%20are%20well%2Ds suited,events%20in%20a%20time%20series.))

Ask our Experts

Rate this Question?  

[View Queries](#)[open ▾](#)**Question 6****Unattempted**

Domain :Modeling

You are a machine learning specialist working for a healthcare company where you are building a cancer detection model using a linear regression algorithm. You have gathered your data of hundreds of thousands of patients with over 100 features. However, when you train your model you notice that it appears to be over-fitting your data.

Which technique can you use to simultaneously correct the over-fitting and reduce your model complexity by removing less relevant features?

- A. Use Ridge Regression
 - B. Use Lasso Regression
 - C. Use Stochastic Gradient Descent
 - D. Use a Gaussian Process
-

Explanation:

Answer: B

Option A is incorrect. The Ridge Regression approach would reduce the coefficients in your model but not all the way to 0. Therefore, it reduces complexity, but does not entirely eliminate any of the over 100 features in your data.

Option B is correct. The Lasso Regression approach would reduce some of the coefficients in your model to zero, effectively eliminating some of the over 100 features in your data. This will effectively reduce the complexity of your model.

Option C is incorrect. The Stochastic Gradient Descent approach can use a regularization parameter, but it cannot be used to eliminate features from your dataset.

Option D is incorrect. The Gaussian Process approach is used for regression problems, but it does not work well with high dimensional datasets, i.e. over a few dozen features. Your dataset has over 100 features. Also, it cannot be used to eliminate features from your dataset.

Reference:

Please see the Medium article titled **Ridge and Lasso Regression: L1 and L2 Regularization** (<https://towardsdatascience.com/ridge-and-lasso-regression-a-complete-guide-with-python-scikit-learn-e20e34bcbf0b>), and the SciKit Learn page titled **1. Supervised learning** (https://scikit-learn.org/stable/supervised_learning.html#supervised-learning)

[Ask our Experts](#)[Rate this Question?](#)  [View Queries](#)[open](#) ▾**Question 7****Unattempted****Domain :Modeling**

You are a machine learning specialist working for a sports gambling company where you are responsible for building a machine learning model to predict the point spread and over/under of NCAA and NFL games. You have built your custom deep learning model using TensorFlow in SageMaker. You have attempted to train your model on a single GPU but you have noticed that the amount of game data you need to train with exceeds the single GPU capacity. How can you change your machine learning code to get it to use multiple GPUs with the least amount of effort on your part?

- A. Rewrite your model to use the Factorization Machines algorithm

- B. Rewrite your code in PySpark and use spark to run your code across multiple GPUs
- C. Add Horovod to your code and use its distributed deep learning training framework for TensorFlow 
- D. Rewrite your model to use the DeepAR Forecasting algorithm

Explanation:

Answer: C

Option A is incorrect. The Factorization Machines algorithm is used for classification and regression problems, not deep learning predictions.

Option B is incorrect. Rewriting your model in PySpark would require more work compared to using the Horovod framework.

Option C is correct. Using the Horovod distributed deep learning training framework for TensorFlow allows you to easily distribute your training across many GPUs in parallel.

Option D is incorrect. Rewriting your code to use the DeepAR algorithm would require more work compared to using the Horovod framework.

Reference:

Please see the GitHub page titled **Horovod** (<https://github.com/horovod/horovod>), the Amazon SageMaker developer guide titled **Use Amazon SageMaker built-in algorithms** (<https://docs.aws.amazon.com/sagemaker/latest/dg/algos.html>), and the Medium article titled **3 Methods for Parallelization in Spark** (<https://towardsdatascience.com/3-methods-for-parallelization-in-spark-6a1a4333b473>)

[Ask our Experts](#)[Rate this Question?](#)  

[View Queries](#)[open ▾](#)**Question 8****Unattempted****Domain :Modeling**

You are a machine learning specialist working for a credit card company where you are building a fraud detection model. You have your model built using the XGBoost algorithm and you are now attempting to find the best version of your model by performing automatic model tuning. You are creating your hyperparameter tuning job and you need to select the appropriate technique the job will use to find your best hyperparameters. You wish to run the least number of hyperparameter tuning training jobs as possible.

Which hyperparameter tuning technique is best suited to your requirements?

- A. Bayesian optimization 
- B. Bayesian classification
- C. Random search
- D. Logistic regression

Explanation:

Answer: A

Option A is correct. The Bayesian optimization approach to hyperparameter tuning results in less tuning job runs than the random search method.

Option B is incorrect. The Bayesian technique used by the hyperparameter tuning job in SageMaker is Bayesian optimization, which solves the problem using regression, not classification.

Option C is incorrect. The random search method is a valid option when using SageMaker hyperparameter tuning, but the random search approach usually requires running many more training jobs to get the best hyperparameters when compared to the Bayesian optimization approach. You have a requirement to run the least number of hyperparameter tuning training jobs as possible.

Option D is incorrect. Logistic regression is not a valid option when using SageMaker hyperparameter tuning.

Reference:

Please see the Amazon SageMaker developer guide titled **How Hyperparameter Tuning Works** (<https://docs.aws.amazon.com/sagemaker/latest/dg/automatic-model-tuning-how-it-works.html>), the AWS Machine Learning blog titled **Amazon SageMaker automatic model tuning now supports random search and hyperparameter scaling** (<https://aws.amazon.com/blogs/machine-learning/amazon-sagemaker-automatic-model-tuning-now-supports-random-search-and-hyperparameter-scaling/>), and the AWS Machine Learning blog titled **Simplify machine learning with XGBoost and Amazon SageMaker** (<https://aws.amazon.com/blogs/machine-learning/simplify-machine-learning-with-xgboost-and-amazon-sagemaker/>)

Ask our Experts

Rate this Question?  

View Queries

open 

Question 9**Unattempted****Domain :Modeling**

You are a machine learning specialist working for a social media company where your team is responsible for building a machine learning model to classify the images that your users submit to your service. You have built a neural network to classify the images. You are now performing mini-batch training of the neural network and you see that your resulting training accuracy is oscillating. What is a likely reason for this issue?

- A. The epochs hyperparameter is set too low
- B. The momentum hyperparameter is set to 0.9
- C. The dropout hyperparameter is set to 0
- D. The learning_rate hyperparameter is set too high 

Explanation:

Answer: D

Option A is incorrect. The epochs hyperparameter controls how many training epochs. A low epoch value will not cause oscillating accuracy results.

Option B is incorrect. The momentum hyperparameter is used to control the speed of the optimization process. It can be used to prevent oscillations, but it would not cause oscillation.

Option C is incorrect. The dropout hyperparameter is used to prevent overfitting. A low value, such as 0, would not cause oscillating accuracy results.

Option D is correct. The learning_rate hyperparameter, when set to a very high value, can cause oscillation of accuracy results.

Reference:

Please see the AWS Machine Learning blog titled **Amazon SageMaker automatic model tuning produces better models, faster** (<https://aws.amazon.com/blogs/machine-learning/amazon-sagemaker-automatic-model-tuning-produces-better-models-faster/>), the AWS Machine Learning blog titled **The importance of hyperparameter tuning for scaling deep learning training to multiple GPUs** (<https://aws.amazon.com/blogs/machine-learning/the-importance-of-hyperparameter-tuning-for-scaling-deep-learning-training-to-multiple-gpus/>), the Amazon SageMaker developer guide titled **Image Classification Hyperparameters** (<https://docs.aws.amazon.com/sagemaker/latest/dg/IC-Hyperparameter.html>), the Nanonets article titled **How To Make Deep Learning Models That Don't Suck** (<https://nanonets.com/blog/hyperparameter-optimization/>), and the Hackernoon article titled **Hyperparameter Tuning Platforms are Becoming a New Market in the Deep Learning Space**

(<https://medium.com/hackernoon/hyperparameter-tuning-platforms-are-becoming-a-new-market-in-the-deep-learning-space-7106foac1689>)

[Ask our Experts](#)Rate this Question?  [View Queries](#)

open ▾

Question 10**Correct**

Domain :Modeling

You are a machine learning specialist working for a retail clothing conglomerate. Your company sells many lines of clothing, such as budget casual wear, office casual wear, office formal wear, etc. For each of your existing products in these categories you have been using autoregressive integrated moving average (ARIMA) models to forecast demand. You now wish to forecast demand for a new product based on the collective historical time series data from your existing products. Which approach should you take to forecast demand for your new product?

- A. Forecast demand for your new product using the XGBoost algorithm
- ✓ B. Forecast demand for your new product using the DeepAR algorithm 
- C. Forecast demand for your new product using an ARIMA model
- D. Forecast demand for your new product using k-means clustering

Explanation:**Answer: B**

Option A is incorrect. The XGBoost algorithm is used to solve regression, classification, and ranking problems. It would not be a good choice for a forecasting problem.

Option B is correct. The DeepAR Forecasting algorithm works great when you are trying to forecast using many similar time series across a set of cross-sectional units. Which is exactly what you are trying to do: you have several similar time series from across your other product lines that you would now like to use to train your single model using all of these time series. The collective time series will help you predict sales for your new product.

Option C is incorrect. The ARIMA forecasting method uses one time series to train the model. You need to train using several time series, that's where the DeepAR algorithm capabilities differentiate; DeepAR forecasts using many similar time series across a set of cross-sectional units.

Option D is incorrect. K-means clustering is used to find discrete groupings in your data. K-means does not fit your problem definition as well as the DeepAR algorithm.

Reference:

Please see the Amazon SageMaker developer guide titled **K-Means Algorithm** (<https://docs.aws.amazon.com/sagemaker/latest/dg/k-means.html>), the Amazon SageMaker developer guide titled **XGBoost Algorithm** (<https://docs.aws.amazon.com/sagemaker/latest/dg/xgboost.html>), the Amazon SageMaker developer guide titled **DeepAR Forecasting Algorithm** (<https://docs.aws.amazon.com/sagemaker/latest/dg/deepar.html>), and the Medium article titled **Understanding Auto Regressive Moving Average Model — ARIMA** (<https://medium.com/fintechexplained/understanding-auto-regressive-model-arima-4bd463b7a1bb>)

Ask our Experts

Rate this Question?  

View Queries

open ▾

Finish Review

Certification

Cloud Certification

Java Certification

PM Certification

Big Data Certification

Company

Become Our Instructor

Support

Discussions

Blog

Business

Support

Contact Us

Help Topics



Join us on Slack!

Join our open **Slack community** and get your queries answered instantly! Our experts are online to answer your questions!

Follow us

