Additional Notes for ML Exam

Handling imbalanced datasets:
https://elitedatascience.com/imbalanced-classes

1. Up-sampling Minority Class / Synthetic Minority Oversampling with SMOTE => https://docs.aws.amazon.com/machine-learning/latest/dg/step-1-download-edit-and-upload-data.html
2. Down-sampling Majority class
3. Change performance metric - typically use AUROX: Area Under ROC Curve
4. Penalize algorithm -> Penalized-SVM

Dealing with missing data
Multiple **imputation** for missing data makes it possible for the researcher to obtain approximately unbiased estimates of all the parameters from the random error. The researcher cannot achieve this result from deterministic imputation, which the multiple imputation for missing data can do.

Amazon Presto
https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-presto.html
fast SQL query engine designed for interactive analytic queries over large datasets from multiple sources -> included in Amazon EMR
https://prestodb.io/

Top 10 Performance Tuning Tips for amazon Athena
https://aws.amazon.com/blogs/big-data/top-10-performance-tuning-tips-for-amazon-athena/
- partition the data
- bucket the data
- use compression / recommend Apache Parquet or Apache ORC which compress the data by default and are splittable
- optimize file size. Too small files (<128Mb) adds an overhead. One remedy: use S3DictCP utility on amazon EMR to combine smaller files into larger objects

RCF
Amazon SageMaker Random Cut Forest (RCF) is an **unsupervised** algorithm for detecting **anomalous data points** within a data set. When using RCF the optional test channel is used to compute accuracy, precision, recall, and F1-score metrics

on labeled data.
Train and test data content types can be either application/x-recordio-protobuf or text/csv and AWS recommends using ml.m4, ml.c4, and ml.c5 instance families for training. Random Cut Forest (RCF) Algorithm - Amazon SageMaker

Security - All data to be encrypted at rest
You can encrypt your Amazon SageMaker storage volumes used for Training and Hosting with AWS Key Management Service (KMS). AWS KMS gives you centralized control over the encryption keys used to protect your data. You can create, import, rotate, disable, delete, define usage policies for, and audit the use of encryption keys used to encrypt your data.
=> You **specify a KMS Key ID when you create Amazon SageMaker notebook instances, training jobs or endpoints**. The attached ML storage volumes are encrypted with the specified key.

Each data scientist with their own notebook
We can attach an IAM policy to each Data Scientist user role allowing them access to their respective notebook instances. Step 7: Create an IAM Role for Amazon SageMaker Notebooks - AWS Glue

Normalization Vs Standardization
You are apply **standardization** techniques to a feature in your dataset. The column has the following values {5, 20, 15}. The standard deviation is 6.23 and the mean of the feature 13.33
Let's take the value 5. To calculate the standardization value we use the following formula z = (x - u) / s where 'z' is the standardized value, where 'x' is our observed value, where 'u' is the mean value of the feature, and 's' is the standard deviation. For 5, -1.33 = (5 - 13.33) / 6.23. For 15, 0.26 = (15 - 13.33) / 6.23. For 20, 1.06 = (20 - 13.33) / 6.23.
Since 5 is the only value that produces a negative value, {-1.33, 1.06, 0.26} is the only acceptable answer.
The result of **standardization** (or **Z-score normalization**) is that the features will be rescaled to ensure the mean and the standard deviation to be 0 and 1, respectively. The equation is shown below:

$$x_{\text{stand}} = \frac{x - \text{mean}(x)}{\text{standard deviation }(x)}$$

You are apply **normalization** techniques to a column in your dataset. The column has the following values {1, 5, 7}. When we apply normalization what will the respective output results be?Applying normalization translates each feature individually such that it is in the given range on the training set **between 0 and 1**. In this case {1, 5, 7} maps to {0.00, 0.66, 1.00} respectively.

- **Normalization** is good to use when you know that the distribution of your data does not follow a Gaussian distribution. This can be useful in algorithms that do not assume any distribution of the data like K-Nearest Neighbors and Neural Networks.
- **Standardization**, on the other hand, can be helpful in cases where the data follows a Gaussian distribution. However, this does not have to be necessarily true. Also, unlike normalization, standardization does not have a bounding range. So, even if you have outliers in your data, they will not be affected by standardization.

Incremental Training

https://docs.aws.amazon.com/sagemaker/latest/dg/incremental-training.html
Over time, you might find that a model generates inference that are not as good as they were in the past. With incremental training, you can use the artifacts from an existing model and use an expanded dataset to train a new model. Incremental training saves both time and resources.
Use incremental training to:

- Train a new model using an expanded dataset that contains an underlying pattern that was not accounted for in the previous training and which resulted in poor model performance.

- Use the model artifacts or a portion of the model artifacts from a popular publicly available model in a training job. You don't need to train a new model from scratch.

- Resume a training job that was stopped.

- Train several variants of a model, either with different hyperparameter settings or using different datasets.

Training High Error Rate / Testing Low

When training error is high and testing error is low, this is highly unusual as it infers that the model is somehow predicting better than the data which was used to train the model. This is usually an indicator of a data issue or some systemic problem in the algorithm. Overfitting - Wikipedia

Training High Error Rate / Testing High
When both training and testing errors are high, it indicates that our model is underfitting the data. We can try to add more details to the dataset, gather more data for training and/or run the training session longer. We might also need to identify a better algorithm.

Training High Error Rate / Testing Low
When training accuracy is high and testing accuracy is low, it usually indicates overfitting or insufficient randomization across the training and testing datasets. You can randomize the data and try training again or maybe use a cross-validation method like k-fold. For overfitting, one suggestion is to reduce the number of features in the model.

Multi-class Classifier
Both **XGBoost** and **Linear Learner** are perfect choices for multi classification problems.

- When we are trying to solve a multi classification problem using **XGBoost** we set the objective hyperparameter to **multi:softmax**
- When using the **Linear Learner** algorithm, we set the predictor hyperparameter to **multiclass_classifier**. XGBoost Algorithm - Amazon SageMaker

The **Macro F1 Score** is an unweighted average of the F1 score across all classes and is typically used to evaluate the accuracy of multi-class models. A number **closer to 1** indicates higher accuracy

Linear Regression
Linear Learner and **XGBoost** are two algorithms most closely identified with the use case of a linear regression problem.

For regression tasks, the industry standard **Root Mean Square Error (RMSE)** metric. It is a distance measure between the predicted numeric target and the actual numeric answer (ground truth). The **smaller** the value of the RMSE, the better is the predictive accuracy of the model. A model with perfectly correct predictions would have an RMSE of 0.

Binary Classifier
XGBoost

For binary classification problems, the **AUC** or Area Under the Curve is an industry-standard metric to evaluate the quality of a binary classification machine learning model. AUC measures the ability of the model to predict a higher score for positive examples, those that are "correct," than for negative examples, those that are "incorrect." The AUC metric returns a decimal value from 0 to 1. AUC values near 1 indicate an ML model that is highly accurate.

Topic extraction
Latent Dirichlet Allocation (**LDA**) and Neural Topic Model (**NTM**) algorithms both can perform topic extraction from bodies of text but each use a slightly different method. Neural Topic Model (NTM) Algorithm - Amazon SageMaker Latent Dirichlet Allocation (LDA) Algorithm - Amazon SageMaker

Hyper parameter tuning
Amazon SageMaker uses **Bayesian** optimization for automatic hyperparameter tunin

Amazon Rekognition
Rekognition Video has the ability to detect faces and facial expressions such as smiling in a video stream. This service could be used to replace or augment the work of the students. Amazon Rekognition – Video - AWS

Copying data from S3 to Redshift
You can add data to your Amazon Redshift tables either by using an INSERT command or by using a COPY command. At the scale and speed of an Amazon Redshift data warehouse, the COPY command is many times faster and more efficient than INSERT commands. You can load data from an Amazon DynamoDB table, or from files on Amazon S3, Amazon EMR, or any remote host through a Secure Shell (SSH) connection. When loading data from S3, you can load table data from a single file, or you can split the data for each table into multiple files. The COPY command can load data from multiple files in parallel. Using a COPY Command to Load Data - Amazon Redshift

Using Python custom libraries with AWS Glue
When you are creating a new Job on the console, you can specify one or more library .zip files by choosing Script Libraries and job parameters (optional) and entering the full Amazon S3 library path(s) in the same way you would when creating a development endpoint: Using Python Libraries with AWS Glue - AWS Glue
You can use Python extension modules and libraries with your AWS Glue ETL

scripts as long as they are written in pure Python. C libraries such as pandas are not supported at the present time, nor are extensions written in other languages. Unless a library is contained in a single .py file, it should be packaged in a .zip archive. The package directory should be at the root of the archive, and must contain an __init__.py file for the package. Python will then be able to import the package in the normal way.

One Hot encoding Vs Ordinal

Since these classification values are ordinal (order does matter) we cannot use one-hot encoding techniques. We either need to map these values to a scale, or we train our model with different encodings and seeing which encoding works best.

EC2 F1 Instances

F1 instance types provide the most performance via use of FPGAs to enable delivery of custom hardware accelerations. Good target applications for F1 are ones that have a modest number of distinct operations that account for significant portions of application run-time. Examples of such applications include big data analytics, genomics, electronic design automation (EDA), image and video processing, compression, security, and search/analytics. The down-side of FPGAs is that they usually require very specialized knowledge and programming to deploy models on them.

Other EC2 Instances

General purpose

T2 instances are Burstable Performance Instances that provide a baseline level of CPU performance with the ability to burst above the baseline.

M5 instances are the latest generation of General Purpose Instances powered by Intel Xeon® Platinum 8175M processors. This family provides a balance of compute, memory, and network resources, and is a good choice for many applications.

Amazon EC2 A1 instances deliver significant cost savings and are ideally suited for scale-out and Arm-based workloads that are supported by the extensive Arm ecosystem. A1 instances are the first EC2 instances powered by AWS Graviton Processors that feature 64-bit Arm Neoverse cores and custom silicon designed by AWS.

Compute optimized

C5 instances are optimized for compute-intensive workloads and deliver cost-effective high performance at a low price per compute ratio.

Memory optimized
R5 instances deliver 5% additional memory per vCPU than R4 and the largest size provides 768 GiB of memory. In addition, R5 instances deliver a 10% price per GiB improvement and a ~20% increased CPU performance over R4.
X1 instances are optimized for large-scale, enterprise-class and in-memory applications, and offer one of the lowest price per GiB of RAM among Amazon EC2 instance types.

Accelerated computing
P3 instances are the latest generation of general purpose GPU instances.
F1 instances offer customizable hardware acceleration with field programmable gate arrays (FPGAs).
G4 instances are designed to help accelerate machine learning inference and graphics-intensive workloads.

Amazon Polly

Using SSML-enhanced input text gives you additional control over how Amazon Polly generates speech from the text you provide. Using these tags allows you to substitute a different word (or pronunciation) for selected text such as an acronym or abbreviation. You can also create a dictionary lexicon to apply to any future tasks instead of apply SSML to each individual document

Using TensorFlow with Amazon Pagemaker

https://docs.aws.amazon.com/sagemaker/latest/dg/tf.html
You can use Amazon SageMaker to train and deploy a model using custom TensorFlow code. The SageMaker Python SDK TensorFlow estimators and models and the SageMaker open-source TensorFlow containers make writing a TensorFlow script and running it in SageMaker easier.

Amazon Semantic Segmention

Amazon SageMaker semantic segmentation expects the training dataset to be on Amazon S3. The dataset in Amazon S3 is expected to be presented in two channels, one for training and one for validation using four directories, two for images and two for annotations. Annotations are expected to be uncompressed PNG images. The dataset might also have a label map that describes how the annotation mappings are established. If not, the algorithm uses a default

Amazon Kinesis Data Firehose is the easiest way to load streaming data into

AWS. Kinesis Data Firehose delivery stream can automatically convert the JSON data into Apache Parquet or Apache ORC format before delivering it to your S3 bucket. Kinesis Data Firehose references table definitions stored in AWS Glue. Choose an AWS Glue table to specify a schema for your source records

S3 VPC Endpoint
Using a VPC Endpoint will redirect the S3 traffic through the AWS private network rather than egressing to the public internet. Both of these attributes will reduce egress costs and increase security
https://docs.aws.amazon.com/vpc/latest/userguide/vpc-endpoints.html
A VPC endpoint enables private connections between your VPC and supported AWS services and VPC endpoint services powered by AWS PrivateLink without requiring an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC do not require public IP addresses to communicate with resources in the service. Traffic between your VPC and the other service does not leave the Amazon network.

VPC Interface Endpoints allow traffic to flow between a VPC and select AWS services without having to exit to the public internet. This would be a good way to keep the sensitive data off the internet.

Restrict access to notebooks by dev team
Of the given options, the one that makes the most sense is using a **ResourceTag** condition attached to the respective Dev team groups. Then you would add a Project tag to the notebook instance indicating it's project designation. These two items together would allow you to restrict notebook instances to only those in the respective DEV team group.
https://docs.aws.amazon.com/sagemaker/latest/dg/security_iam_id-based-policy-examples.html