Notas sobre Machine Learning com AWS

April 16, 2022

1 Engenharia de Dados

1.1 Amazon S3

Visão Geral

- O amazon S3 permite que as pessoas armazenem objetos (arquivos) em buckets (diretórios);
- Os buckets devem ter um nome único global, isso porque o endereço de acesso de todos os usuários do S3 deve ser único;
- Os objetos (arquivos) tem uma chave. A chave é o caminho completo:

  <my_bucket>/my_file.txt

  <my_bucket>/my_folder1/another_folder/my_file.txt
- Isso é útil e interessante quando olharmos partições: as partições permitem uma consulta mais otimizada;
- O tamanho máximo de um objeto é de 5TB;
- Tags de objeto (chave/valor, até 10), úteis para segurança e ciclo de vida;

AWS S3 para Machine Learning

- Backbone para muitos seviços de ML do AWS (ex. SageMaker)
- Cria um DataLake:
  - Tamanho infinito, sem necessidade de provisionar o espaço que será utilizado;
- Durabilidade 99,99999999%
  - A durabilidade é uma medida da probabilidade de seus dados serem não serem corrompidos, quanto maior a durabilidade, menos a chance de serem corrompidos.
- Armazenamento (S3) desacoplado do processamento (EC2, Amazon, Athena, Amazon RedShift Spectrum, Amazon Rekognition e AWS Glue);
- Arquitetura centralizada: isso é importante para que se tenha todos os seus arquivos em único local de forma centralizada;
- Armazenamento de objetos: suporta qualquer tipo de arquivo;
- Formatos comuns para ML: CSV, JSON, Parquet, ORC, Avro, Protobuf;

AWS S3: Particionamento

- Padrões para acelerar consultas em intervalos (ex: AWS Athena)
- Por data: S3://bucket/my-data-set/year/month/day/hour/data_00.csv
- Por produto: S3://bucket/my-data-set/product-id/data_32.csv
- Você pode definir qual estratégia de particionamento você quer
- O particionamento de dados será feito por algumas ferramentas que vamos usar. (AWS Glue)

1.2 S3 Storage - Camadas de Armazenamento e Políticas de Ciclo de Vida

- Amazon S3 Standard - Uso geral;
- Amazon S3 - Acesso Infrequente (IA);
- Amazon S3 One Zone - Acesso Infrequente: Armazenada em um local único;
- Amazon S3 - Camada Inteligente: O serviço escolhe os melhores locais de armazenamento visando melhores preços;
- Amazon Glacier: Utilizado para arquivar dados que não serão mais utilizados;

Regras de Ciclo de vida

- Conjunto de regras para mover os dados entre diferentes camadas, para reduzir custo de armazenamento;
- Exemplo: Primeiramente o arquivo é alocado para uso geral, depois de um tempo, este arquivo é alocado para uso infrequente, depois de mais algum tempo, esse arquivo então é arquivado, sendo alocado para o Amazon Glacier;
- Ações de transição: Objetos fazem a transição para outra classe de armazenamento;
  - Mover objetos da classe Padrão para o Uso Infrequente depois de 60 dias de criação;
  - Mover os objetos do Uso Infrequente para o Glacier após 6 meses;
- Ações de expiração: O S3 exclui objetos experiados para nós ao nosso critério.

1.3 Segurança

Criptografia S3 para Objetos

- Criptografia S3: A criptografia pode ser realizada no servidor AWS ou pelo próprio cliente;
- Exitem 4 métodos de criptografia para objetos no S3;
  - SSE-S3: Criptografa objetos S3 usando chaves criadas e gerenciadas pelo AWS;
  - SSE-KMS: Usa o serviço de gestão de chaves do AWS para gerenciar chaves criptográficas;
    - Segurança adicional (usuário deve ter acesso à chave KMS);
    - Trilha de auditoria da chave KMS;
  - SSE-C: Quando se quer gerencias as suas próprias chaves;
  - Criptografia no cliente: Ocorre fora do ambiente AWS;
- De uma perspectiva de ML, SSE-S3 e SSE-KMS são os mais usados;

Gestão de Segurança

- Gestão de segurança baseada no usuário
  - Políticas IAM (Identity Access Management) - Quais chamadas de API devem ser permitidas para usuários específicos;
- Baseada em recursos
  - Políticas de buckets:
    - Regras abrangentes de buckets;
    - Permitem contas cruzadas
  - Lista de controle de acesso a objetos (ACL):
    - Controle mais detalhado;
  - Lista de controle de acesso ao bucket (ACL):
    - Menos comum;
- Políticas de Buckets S3:
  - Políticas baseadas em JSON:
    - Gerenciamento de Recursos: buckets e objetos
    - Ações: vai permitir ou negar o consumo de uma API;
    - Efeito da API: Permitir ou negar;
    - Principal: A conta ou usuário que aplica a política;
  - Use a política de Bucket S3 para:
    - Dar acesso ao público a um bucket;
    - Forçar objetos a serem criptografados no upload;
    - Dar acesso a outra conta (conta cruzada);
- Criptografia Padrão S3 vs Políticas de Buckets
  - A forma antiga de habilitar criptografia por padrão era usar uma política de Bucket e recusar qualquer comando HTTP sem o cabeçalho apropriado;
  - A nova forma é usar a opção de "criptografia padrão" no S3;

Segurança S3 - Outros

- Rede - VPC (Virtual Private Cloud) Endpoint Gateway:
  - Permite que o tráfico fique dentro da sua VPC, ao invés de ir para a internet pública;
  - Garante que seus serviços privados (AWS SageMaker) possam acessar o S3;
  - Muito importante para o exame AWS ML;
- Log e Auditoria
  - Logs de acesso podem ser armazenados em outro bucket S3;
  - Chamadas de API podem ser logadas na "AWS Cloud Trail";
- Baseados em Tags: Combina política IAM e políticas de Bucket:
  - Exemplo: Adicionar a tag classification=PHI a seus objetos;

1.4 AWS Kinesis Data Stream Data Firehose

AWS Kineses: Visão Geral

- Kinesis é uma alternativa gerenciada do Apache Kafka;
- Ótimo para logs de aplicações, IOT, clickstreams;
- Ótimo para processar dados em tempo real;
- Ótimo para frameworks de processamento em streaming (Spark, NiFi, etc.)
- Dados são automaticamente replicados de forma síncrona para $3AZ$
- O AWS Kinesis é formado por 4 aplicações principais:
  - Kinesis Streams: Ingestão de streaming com baixa latência e em grande escala;
  - Kinesis Analytics: Executa análise em tempo real em streams usando SQL;
  - Kinesis Firehose: Carrega streams no S3, RedShift, ElasticSearch e Splunk;
  - Kinesis Video Streams: Usado para stream de vídeo em tempo real;
- Visão Geral do Kinesis Streams:
  - Streams são divididos em Shards ordenados/partições;
  - Os shards devem ser provisionados com antecedência (Planejamento de capacidade);
  - Retenção de dados é de 24 horas por padrão, pode chegar a 7 dias;
  - Capaz de reprocessar dados;
  - Várias aplicações podem consumir o mesmo stream;
  - Uma vez que os dados são inseridos no Kinesis, não pode ser excluído (Imutabilidade);
  - Registros podem ter até 1MB de tamanho;
- Limites de dados do Kinesis Data
  - Produtores
    - 1MB por segundo ou 1000 mensagens na escrita por SHARD;
    - Ou então "ProvisionedThroughputException" será levantado, caso o limite seja ultrapassado;
  - Consumidor clássico:
    - 2MB/s de leitura por SHARD entre todos os consumidores;
    - 5 camadas de API por segundo por SHARD entre todos os consumidores
  - Retenção de dados:
    - 24 horas por padrão
    - Pode ser estendido até por 7 dias;
- Kinesis Data Firehose:
  - Serviço totalmente gerenciado, sem administração;
  - Próximo ao tempo real (mínimo de 60 segundos de latencia para batches não completos);
  - Ingestão de dados no RedShift/Amazon S3/ElasticSearch/Splunk
  - Escala automaticamente;
  - Suporte a muitos formatos de dados;

  - Conversão de dados CSV/JSON para Parquet/ORC (apenas para o S3);
  - Transformação de dados através do AWS Lambda (ex: CSV=>JSON);
  - Suporta compressão quando usa o Amazon S3 (GZIP, ZIP e SNAPPY);
  - Paga-se pela quantidade de dados que entra no Firehose;

- Kinesis Data Analytics
  - Casos de uso:
    * Streaming ETL: Seleciona colunas, faz transformações simples em dados em streaming;
    * Gerador contínuo de métricas: classificação ao vivo para um jogo mobile;
    * Análise responsiva: verifica certo critério e cria alertas (filtrando);
  - Características:
    * Paga apenas por recursos consumidos (mas não é barato);
    * Serverless: escala automaticamente;
    * Usa permissão IAM para acessar fontes de streaming e destinos;
    * SQL ou Flink para escrever o processamento;
    * Descoberta de Schema;
    * Lambda pode ser usado para pré-processamento;
  - Machine Learning no Kinesis Data Analytics:
    * RANDOM_CUT_FOREST:
      · Função SQL usada para detecção de anomalias em colunas numéricas de um stream;
      · Exemplo: detectar passageiros anormais no metrô durante a maratona de Nova Iorque;
      · Usa o histórico recente para processar o modelo;
    * HOTSPOTS:
      · Localiza e informa sobre regiões relativamente densas nos seus dados;
      · Exemplo: uma coleção de servidores superaquecidos em um datacenter;

- Resumo do Kinesis - Machine Learning:
  - Kinesis Data Stream: Cria aplicações de machine learning em tempo real;
  - Kinesis Data Analytics: algoritmos ETL/ML em tempo real em stream;
  - Kinesis Video Stream: Stream de vídeo em tempo real para criar aplicações em ML;

1.5 Glue Data Catalog

- Repositório de metadados para todas as suas tabelas;
  - Automatiza inferência de schema: lembrando que schema é a definição dos seus dados ou a estrutura que os dados tem.
  - O schema é versionado: a medida que a estrutura é mudada, ela é versionada;
- Integrado com o Athena ou RedShift Spectrum (schema e data discovery)
- Glue Crawlers pode lhe ajudar a construir um Glue Data Catalog;

Glue Data Catalog - Crawlers

- Crawlers exploram seus dados para inferir schemas e partições;
- Funciona com JSON, Parquet, CSV, relacional;
- Crawlers funciona com: S3, Amazon RedShift, Amazon RDS;
- O Crawler pode ser agendado ou rodado sob demanda;
- Para rodar o crawler precisa de uma credencial ou IAM Role para acessar as fontes de dados;

Glue e Partições S3

- Glue Crawler vai extrair partições baseadas em como seus dados no S3 estão organizados;
- Pense com antecedência sobre como você consultará seu data lake no S3;
- Exemplo: dispositivos enviam dados de sensores a cada hora;
  - Você consulta principalmente por intervalos de tempo?
    - Em caso afirmativo, organize seus intervalos como S3://my-bucket/dataset/aaaa/mm/dd/dispositivo
  - Você consulta principalmente por dispositivo?
    - Em caso afirmativo, organize seus intervalos como S3://my-bucket/dataset/device/aaaa/mm/dd

Glue ETL

- Transformar dados, limpar dados, enriquecer dados (antes de fazer a análise)
  - Gera código ETL em Python ou Scala, você pode modificar o código;
  - Ou você pode fornecer seus próprios scripts Spark ou Pyspark;
  - O destino pode ser S3, JDBC (RDS, RedShift) ou o catálogo de dados do Glue;
- Totalmente gerenciado, econômico, paga apenas pelos recursos consumidos;
- Os jobs são executados em uma plataforma spark serverless;
- Glue Scheduler para agendar os trabalhos
- Glue Triggers para automatizar execuções de trabalho com base em "eventos"

Glue ET: Transformações

- Transformações empacotadas (Bundled):
  - DropFields, DropNullFields: Remover campos nulos;
  - Filter: Específica uma função para filtrar registros;
  - Join: Para enriquecer os dados;
  - Map: Adicionar campos, excluir campos, realizar pesquisas externas;
- Transformações para aprendizado de máquinas:
  - FindMatches ML: Identifica registros duplicados ou correspondentes em seu conjunto de dados, mesmo quando os registros não têm um identificador único comum e nenhum campo que corresponde exatamente;
  - Conversões de formato: CSV, JSON, AVRO, Parquet, ORC, XML;
  - Transformações do Apache Spark (exemplo: K-means);

1.6 AWS Data Stores Para Machine Learning

- RedShift - Data Warehousing, SQL Analytics (OLAP - Online Analytical Processing):
  - Carrega dados do S3 para o RedShift;
  - Use o RedShift Spectrum para consultar dados diretamente no S3 (sem necessitar carregar);
- RDS, Aurora:
  - Armazenamento Relacional, SQL (OLTP - Online Transaction Processing);
  - Servidores devem ser provisionados com antecedência;
- DynamoDB:
  - Armazenamento NoSQL, serverless, provisão com capacidade de leitura e escrita;
  - Útil para armazenar o modelo de machine learning servido pela sua aplicação;
- S3:
  - Armazenamento de objetos;
  - Serverless, armazenamento infinito;
  - Integração com a maioria dos serviços AWS;
- ElasticSearch:
  - Indexação de dados;
  - Pesquisa nos dados;
  - Clickstream Analytics;
- ElastiCache:
  - Mecanismo de cache;
  - Não é usado para aprendizado de máquina;

1.7 AWS Data Pipeline

- Os destinos incluem S3, RDS, DynamoDB, RedShift e EMR;
- Gerencia dependências entre tarefas;
- Tenta novamente e notifica falhas;
- As fontes de dados podem ser locais (on-premises);
- Altamente disponível;

AWS Data Pipeline vs Glue

- Glue:
  - Glue ETL - Executa o código Apache Spark, baseado em Scala ou Python, com foco no ETL;
  - Não se preocupa em configurar ou gerenciar os recursos;
  - Tem um catálogo de dados para disponibililzar os dados para o Athena ou RedShift Spectrum;

- Data Pipeline:
  - Serviço de Orquestração;
  - Mais controle sobre o ambiente, recursos de computação que executam código;
  - Permite acesso à instâncias EC2 ou EMR (cria recursos em sua própria conta);

2 Análise de dados exploratória

2.1 AWS Athena

- Serviço de consulta interativa para S3 (SQL);
  - Não há necessidade de carregar dados, ele permanece no S3;
  - Utiliza Presto;
  - Serverless;
  - Suporta muitos formatos de dados:
    - CSV (legível para humanos);
    - JSON (legível para humanos);
    - ORC (colunar, divísivel);
    - Parquet (colunar, divísivel);
    - Avro (divísivel);
    - Não estruturado, semiestruturado ou estruturado;
  - Alguns exemplos:
    - Consultas ad-hoc de logs da web;
    - Consultas de dados de teste antes de carregar no RedShift;
    - Analisa olos do cloudtrail/cloudfront/VPL/ELB, etc no S3;
    - Integração com notebookos do Jupyter, Zeppelin, RStudio;
    - Integração com o QuickSight;
    - Integração via ODBC/JDBC com ferramentas de visualização;

  Modelo de custo do Athena;

  - Paga conforme o uso
    - $5 por TB;
    - Consultas bem-sucedidas ou canceladas contam, consultas com falha não;
  - Sem custo para DDL (CREATE, ALTER, DROP, etc.)
  - Economize muito dinheiro usando formatos em colunas:
    - ORC, Parquet;
    - Economize 30-90% e obtenha melhor desempenho;
  - Glue e S3 têm seus próprios custos;

Segurança no Athena

- Controle de Acesso
  - IAM, ACLs, Políticas de bucket S3;
  - /AmazonAthenaFullAccess/AWSQuicksightAthenaAccess
- Criptografia resultados "at rest" no diretórios de staging do S3;
- Croptografia "server side" com chave KMS (CSE-KMS);
- Croptografia "server side" com chave KMS (SSE-KMS);
- Possibilidade de acesso entre contas na política de Bucket S3;
- Criptografia Transport Layer Security (TLS) "em trânsito". (entre Athena e S3)

Quando não usar o Athena

- Relatório/Visualização altamente formatados
  - É para isso que serve o Quicksight
- ETL
  - Em vez disso, use o Glue;

2.2 Amazon QuickSight

- Serviço de análise de dados rápido, fácil e baseado na nuvem;
- Permite que todos os funcionários de uma organização:
  - Crie visualizações;
  - Realizar análise ad-hoc;
  - Obtenha rapidamente insight de negócios a partir de dados;
  - A qualquer hora, em qualquer dispositivo (navegadores, celular);
  - Serverless;
- Fontes de dados QuickSight
  - RedShift;
  - Aurora/RDS;
  - Athena;
  - EC2: Hosted databases
  - Arquivos (S3 ou on-premises)
    - Excel;
    - CSV, TSV
    - Formatos de log
  - A preparação de dados permite ETL limitado;
- SPICE
  - Os conjuntos de dados são importados para o SPICE

        * Mecanismo de cálculo in-memory super rápido e paralelo;
        * Usa armazenamento colunar in-memory, geração de código de máquina;
        * Acelera consultas interativas em grandes conjuntos de dados;
    - Cada usuário recebe 10GB de SPICE;
    - Altamente disponível durável;
    - Escala para centenas de milhares de usuários;

Casos de uso do QuickSight

- Exploração/Visualização ad-hoc interativa de dados;
- Painéis e KPIs;
- Histórias
    - Visitas guiadas por visualizações específicas de uma análise
    - Transmitir os pontos-chave, processo de pensamento, evolução de uma análise;
    - Analisar/Visualizar dados de:
        * Logs no S3;
        * Bancos de dados on-premises
        * AWS (RDS, RedShift, Athena, S3)
        * Aplicativos SaaS, como salesforce;
        * Qualquer fonte de dados JDBC/ODBC
- Machine Learning Insights
    - Detecção de anomalia;
    - Previsão;
    - Autonarrativas

QuickSight quando não usar

- Relatórios prontos formatados;
    - Quicksight é para consultas e visualizações ad-hoc;
- ETL
    - Use Glue, embora Quicksight possa fazer algumas transformações;
- Segurança do Quicksight;
    - Autenticação multifator em sua conta
    - Conectividade VPC;
        * Adicione o intervalo de endereços IP do Quicksight aos seus grupos de segurança de banco de dados;
    - Segurança de nível de linha;
    - Acesso VPC privado
        * Interface de rede "Elastic", AWS Direct Connect;

Gestão de usuários no QuickSight

- Usuários definidos via IAM ou inscrição por e-mail;
- Integração do Active Directory com QuickSight Enterprise Edition;
- Precificação do QuickSight:
  - Pagamento anual:
    - Standard: $9/Usuário/mês
    - Enterprise: $18/Usuário/mês
  - Capacidade extra do spice (além do 10GB)
    - $0,25 (standard) $0,38 (enterprise)/GB/mês
  - Mês a mês
    - Standard: $12/GB/mês
    - Enterprise: $24/GB/mês
  - Edição Enterprise:
    - Criptografia "at rest"
    - Integração com Microsoft Active Directory

2.3 O que é EMR?

- Elastic Map Reduce;
- Estrutura Gerenciada do Hadoop em instâncias EC2;
- Inclui Spark, HBase, Presto, Flink, Hive e mais;
- EMR notebooks;
- Vários pontos de integração com AWS;

Cluster EMR

- Master Node: Gerencia o cluster
  - Instâncias EC2 única;
- Core Node: hospeda dados HDFS e executa tarefas
  - Pode ser escalado para cima e para baixo, mas com algum risco;
- Task Node: Executa tarefas, não hospeda dados;
  - Sem risco de perda de dados ao remover;
  - Bom uso de instâncias pontuais;

Uso do EMR

- Clusters transitórios versus clusters de longa duração
  - Pode alternar task nodes usando "Spot Instances" para capacidade temporária;
  - Pode usar instâncias reservadas em clusters de longa duração para economizar dinheiro;
- Conecta diretamente ao mestre para executar ao jobs;
- Envie as etapas ordenadas por meio do console;

Integração EMR / AWS

- Amazon EC2 para as instâncias dos nós que fazem parte do cluster
- Amazon VPC para configurar a rede virtual na qual você inicia suas instâncias;
- Amazon S3 para armazenar dados de entrada e saída;
- Amazon CloudWatch para monitorar o desempenho do cluster e configurar alarmes;
- AWS IAM para configurar permissões;
- AWS CloudTrail para auditar as solicitações feitas ao serviço;
- AWS Data Pipeline para agendar e iniciar seus clusters;

EMR Storage

- HDFS
- EMRFS: Acessa S3 como se fosse HDFS
  - Visualização consistente EMRFS - Opcional para consistência S3;
  - Usa DynamoDB para monitorar consistência;
- Sistema de arquivos local
- EBS para HDFS
- O armazenamento é temporário, então quando você encerrar o cluster os dados serão perdidos;
- Então ele é útil como armazenamento em cache para resultados temporários;
- Cobrança do EMR por hora
  - Mais cobrança EC2
- Provê novos nós se um nó central falhar;
- Pode adicionar e remover nós de tarefas rapidamente;
- Pode redimensionar os nós centrais de um cluster em execução;

Então, o que é Hadoop?

- O Hadoop é um ecossistema de bigdata.
- Ele é composto pelos seguintes módulos:
  - Hadoop Core, que tem as bibliotecas e utilitários necessários para os outros módulos do Hadoop;
  - HDFS, sistema de arquivos distribuídos e escalável para Hadoop, ele distribui os dados que armazena entre todas as instâncias do cluster. Neste caso aqui, o HDFS é não persistente, portanto, se você encerra o cluster, os dados são apagados;
  - Hadoop YARN (Yet Another Resource Negotiator) é um componente para gerenciar de forma centralizada os recursos do cluster;
  - E por fim, o MapReduce, que é uma estrutura de software que permite que você crie aplicativos que são capazes de processar grandes volumes de dados, em paralelo, em clusters, ao mesmo tempo tolerante à falhas;

Apache Spark

- O Apache Spark é um sistema de processamento distribuído de código aberto, e ele é frequentemente utilizado para processamento de dados;
- Ele utiliza cache na memória e otimiza a execução das consultas;
- Com esses elementos, você tem um resultado das análises de maneira mais rápido, mesmo com um grande volume de dados sendo processados;
- Além disso, ele tem APIs, para você desenvolver as suas aplicações em Java, Scala, Python e R;
- Ele suporta a reutilização de código, para vários tipos de tarefas;

Como ele é utilizado?

- Ele pode ser utilizado para o processamento de streaming de dados;
  - Por exemplo, ele pode ser utilizado para processar dados em tempo real coletados do Amazon Kinesis;
  - Ou do Apache Kafka, ou do Apache Streaming;
- A análise do Apache Spark é feita de maneira totalmente tolerante à falhas e os resultados são gravados no S3, ou num cluster do HDFS;
- Com relação ao aprendizado de máquina:
  - O Spark inclui o ML Library, ou Machine Learning Library, uma biblioteca para fazer aprendizado de máquina em dados de grandes volumes;
  - Ele possui ainda o Spark SQL, que é uma ferramenta de SQL interativo que é usado para consultas interativas de baixa latência utilizando linguagem SQL ou SQL Hive;
  - Ele não é usado para OLTP, e também não é utilizado para processar dados em lote (batch). Ele é uma ferramenta para processar dados à medida que os dados são carregados;
- Como funciona o Spark?
  - Os aplicativos Spark são executados como conjuntos independentes de processos em um cluster;
  - Eles são coordenados pelo objeto Spark Context, no programa principal;
  - Logo, a primeira coisa a ser feita quando vamos processar dados utilizando o Spark é criar o contexto do Spark;
  - O Spark Context se conecta em diferentes gerenciadores de cluster que alocam recursos entre os aplicativos que estão usando os contextos;
  - E quando eles se conectam, o spark adquire executores nos nós que fazem parte do cluster;
  - Esses executores são proessos que executam cálculos e armazenam dados para os aplicativos que estão executando este código;
  - Na etapa final, o Spark Context envia a tarefa para os executores então executarem;

Spark MLLib

- Classificação: logistic regression, naive bayes;
- Regressão;
- Árvore de Decisão;
- Recommendation engine (ALS);
- Clustering (K-Means);
- LDA (topic modeling);
- Utilitários de fluxo de trabalho de ML (pipelines, transformação de atributos, persistência)
- SVD, PCA, estatísticas;

Zeppelin + Spark

- Pode executar o código do spark interativamente (como no shell do spark);
- Isso acelera seu ciclo de desenvolvimento;
- Permite fácil experimentação e exploração de seu big data;
- Pode executar consultas SQL diretamente no SparkSQL;
- Os resultados da consulta podem ser visualizados em tabelas e gráficos;
- Faz com que o Spark pareça mais uma ferramenta de ciência de dados;

EMR Notebook

- Conceito semelhante ao Zeppelin, com mais integração AWS;
- Notebooks com backup para S3;
- Provisione clusters a partir do notebook;
- Hospedado em um VPC;
- Acessado apenas por meio do console AWS;

Segurança no EMR

- IAM policies: Concedem ou negam permissão e determinam quais ações um usuário pode tomar no EMR, além de outros recursos do AWS. Também podemos combinar políticas IAM com tag para controlar o acesso aos clusters;
- Kerberos: É um protocolo de autenticação de rede, que garante que as senhas e outras credenciais não sejam enviadas através da rede sem ser criptografadas;
- SSH: Fornece uma forma segura para os usuários se conectarem por linha de comando em instâncias de um cluster. Ele também fornece tunelamento para que as interfaces web dos aplicativos hospedados no Master Node sejam acessadas;
- IAM roles: Controla como o EMR pode acessar outros serviços do AWS. Cada cluster deve ter uma role de serviço e uma role por perfil de instância do EC2, e as políticas de IAM anexadas a essa role fornece permissões para o cluster interoperar com outros serviços do AWS e isso vai ser feito com o nome do próprio usuário;

EMR: Escolhendo os tipos de Instância

- Master Node:
  - m4.large se <50 nós, m4.xlarge se >50 nós;
- Core & task nodes:
  - m4.large geralmente é suficiente
  - Se o cluster esperar muito por dependências externas (por exemplo um web crawler), t2.medium;
  - Desempenho maior: m4.xlarge;
  - Aplicativos de computação intensiva: instâncias com mais CPU;
  - Banco de dados, aplicativos de cache em memória: instâncias com mais memória;
  - Rede/uso intensivo de CPU (NLP, ML) - instâncias de cluster de computador;
- Spot Instances
  - Bons para nós de tarefa;
  - Use apenas no core & master se você estiver testando ou muito sensível ao custo, você está arriscando a perda parcial de dados;

2.4 Engenharia de Atributos

O que é a Engenharia de Atributos?

- Aplicar seu conhecimento dos dados - e do modelo que você está usando - para criar características melhores para treinar o seu modelo.
  - Quais características devo usar?
  - Preciso transformar estas características de alguma forma?
  - Como faço para lidar com dados faltantes?
  - Devo criar novas características a partir das existentes?
- Você não pode simplesmente inserir dados brutos e esperar bons resultados;
- Esta é a arte do aprendizado de máquina, onde a experiência é aplicada;

A Maldição da Dimensionalidade

- Muitas características podem ser um problema - leva a dados esparsos;
- Cada característica é uma nova dimensão
- Grande parte da engenharia de atributos é selecionar os atributos mais relevantes para o problema em questão
  - Muitas vezezé aqui que o conhecimento do domínio entra em jogo
- Técnicas de redução de dimensionalidade não supervisionadas também podem ser empregadas para transformar muitas características em menos características;
- PCA;
- K-Means;

Imputando dados ausentes: substituição pela média

- Substituir os valores ausentes pelo valor médio das colunas (colunas, não linhas! Uma coluna representa um único atributo. Só faz sentido tirar a média de outras amostras do mesmo atributo);
- Rápido e fácil, não afeta a média ou o tamanho da amostra do conjunto de dados geral;
- A mediana pode ser uma escolha melhor do que a média quando há outliers;
- De uma maneira geral pode não ser uma boa ideia:
  - Funciona apenas a nível da coluna, perde correlações entre as características;
  - Não pode ser usado em características categóricas (inputar com valor mais frequente pode funcionar neste caso, no entando...);
  - Não muito preciso;
  - Ainda mais profundo - mantém o desempenho por meio skip connections;

2.5

- Codificador -> Decodificador;
  - Sequência -> Vetor -> Sequência;
  - Ou seja, tradução automática;

Treinando RNN

- Backpropagation através do tempo;
  - Assim como a backpropagation em MLPs, mas aplicada a cada etapa de tempo;
- Todas essas etapas de tempo aumentam rapidamente;
  - Acaba parecendo uma rede neural muito, muito profunda;
  - Pode limitar a backpropagation a um número limitado de estapas de tempo (Backpropagation truncada ao longo do tempo);
- Estado de etapas de tempo anteriores se dilui ao longo do tempo;
  - Isso pode ser um problema, por exemplo, ao aprender estruturas de frases;
- Célula LSTM;
  - (Long Short-Term Memory Cell) Célula de Memória Longa de Curto Prazo;
  - Mantém estados separados de curto e longo prazo;
- Célula GRU;
  - Unidade recorrente bloqueada - Gated Recurrent Unit;
  - Célula LSTM simplificada com desempenho quase igual;
- É realmente difícil;
  - Muito sensível a topologias, escolha de hiperparâmetros;
  - Uso intensivo de recursos;
  - Uma escolha errada pode levar a uma RNN que não converge;

2.6 Deep Learning no EC2/EMR

- EMR é compatível com Apache MXNet e tipos de instância GPU;
- Tipos de instância apropriados para aprendizado profundo:
  - P3: 8 GPUs Tesla V100;
  - P2: 16 GPUs K80;
  - G3: 4GPUs M60 (todos os chips Nvidia);
- Deep Learning AMI's (TensorFlow, PyTorch, Apache MXNet, Chainer, Gluon e Keras);
  - Não tem custo adicional;
- Sagemaker;

2.7 Tuning de redes neurais

Taxa de aprendizagem (Learning Rate)

- As redes neurais são treinadas por gradiente descent (ou técnicas semelhantes);
- Começamos em algum ponto aleatório, e amostramos soluções diferentes (pesos) buscando minimizar alguma função de custo, ao longo de muitas epochs;
- A distância entre essas amostras é a taxa de aprendizagem;

Efeito da taxa de aprendizagem

- Uma taxa de aprendizado muito alta significa que você pode ultrapassar a solução ideal;
- Uma taxa de aprendizado muito pequena demorará muito para encontrar a solução ideal;
- A taxa de aprendizagem é um exemplo de um hiperparâmetro;

Batch size

- Define quantas amostras de treinamento são usada em cada epoch;
- Um tanto contra-intuitivo:
  - Batch size pequenos podemo ultrapassar os "mínimos locais" mais facilmente;
  - Batch size muito grandes podem acabar ficando presos na solução errada;

Para recapitular

- Batch sizes pequenos tendem a não ficar presos nos mínimos locais;
- Batch sizes grandes podem convergir para a solução errada aleatoriamente;
- Grandes taxas de aprendizagem podem ultrapassar a solução correta;
- Pequenas taxas de aprendizagem aumentam o tempo de treinamento;

2.8 Técnicas de regularização

O que é regularização?

- Prevenindo o overfitting;
  - Modelos que são bons em fazer previsões sobre os dados em que foram treinados, mas não sobre novos dados que não tenham visto antes;
  - Modelos superdimensionados aprenderam padrões nos dados de treinamento que não se generalizam para o mundo real;
  - Frequentemente visto como alta precisão no conjunto de dados de treinamento, mas menor precisão no conjunto de dados de teste ou avaliação;
    - Ao treinar e avaliar um modelo, usamos conjunto de dados de treinamento, avaliação e teste;
- As técnicas de regularização têm como objetivo evitar o overfitting;

Algumas técnicas de regularização

- Simplificação do modelo:
  - Muitas vezes um modelo pode estar sendo construído de uma maneira muito mais complexa do que deveria ser, por um exemplo, imagine que numa rede neural nós temos centenas de camadas ocultas e cada comada oculta possui milhares de neurônios, tudo isso para obter um simples modelo de classificação binária que poderia ser feito utilizando uma árvore de decisão;
  - É claro que isso é uma maneira um tanto quanto exagerada de expor o problema e na vida real, a linha entre o quão complexo e o mínimo de simplificação que ele tem que ter para ser eficiente não é tão evidente. Mas num caso como esse, diminuir o número de camadas e o número de neurônios por camada pode ajudar e muito em reduzir as chances de o modelo superajustar;
  - Isso porque reduzir a complexidade do modelo também reduz as chances de ele aprender padrões complexos (lê-se ruídos) da base de treinamento;
- Dropout:
  - O dropout é uma técnica que remove neurônios da sua rede de maneira completamente aleatória;
  - Isso força a sua rede neural a aprender padrões genéricos da base de dados, pois diminui a possibilidade de algum desses neurônios aprenderem padrões específicos (provavelmente associados a ruídos) da base de treinamento;
- Parada Antecipada;
  - O que podemos observar no treinamento de uma rede neural é que ao decorrer das suas epochs, as suas métricas de classificação podem acabar por estabilizar em um valor específico, sem apresentar melhoras reais com novos treinamentos;
  - Nesses casos, você continuar o treinamento pode acarretar em problemas para o seu algoritmo, pois este pode acabar por gerar um modelo superajustado;

2.9 Gradientes

- Apenas recordando, uma rede neural ela funciona através do ajuste de pesos, então num processo de treinamento os dados passam pela rede neural, as funções de ativação vão produzir valores e na camada de saída, uma loss function vai calcular o erro e esse erro vai ser propagado de volta e neste processo de propagação, os pesos devem ser ajustados;
- Os pesos eles devem ser ajustados de quê forma? Mais ou menos? Qual é o valor do ajuste?
- Nesse aspecto, o gradiente descendente tem um papel fundamental, pois ele vai responder essa pergunta;

O problema do "Vanishing Gradient"

- Quando a inclinação da curva de aprendizado se aproxima de zero, o processamento pode "travar";
- Acabamos trabalhando com números muito pequenos que tornam o treinamento mais lento, ou mesmo introduzem erros numéricos;
- Torna-se um problema com redes mais profundas e RNNs à medida que se propagam para camadas mais profundas;
- Problema oposto: "exploding gradients";

Corrigindo o problema do "Vanishing Gradient"

- Hierarquia multinível;
  - Divida em níveis com suas próprias sub-redes, treinando individualmente;
- Long Short-Term Memory (LSTM);
- Residual Networks;
  - ResNet;
  - Conjunto de redes mais curtas;
- Melhor escolha da função de ativação;
  - ReLU é uma boa escolha;

Verificação de gradiente

- Uma técnica de depuração;
- Verifica numericamente as derivadas calculadas durante o treinamento;
- Útil para validar o código de treinamento de rede neural;
  - Mas provavelmente você não vai querer escrever este código;

2.10 Regularização L1 e L2

O que é?

- Prevenir overfitting em ML em geral;
- Um "termo" de regularização é adicionado à medida que os pesos são aprendidos;
- O termos L1 é a soma dos pesos

$$\lambda \sum_{i=1}^{k} |w_i| \,, \tag{1}$$

- O termo L2 é a soma do quadrado dos pesos

$$\lambda \sum_{i=1}^{k} w_i^2, \tag{2}$$

- A mesma ideia pode ser aplicada a funções de perda;

Qual é a diferença?

- L1: soma dos pesos
  - Executa a seleção de atributos - atributos inteiros vão para 0;
  - Computacionalmente ineficiente;
  - Saída esparsa;
- L2: soma do quadrado dos pesos
  - Todos os atributos permanecem, são apenas podenrados;
  - Computacionalmente eficiente;
  - Saída densa;

Por que você iria usar o L1?

- A seleção de atributos pode reduzir a dimensionalidade;
  - De 100 atributos, talvez apenas 10 acabem com coeficientes diferentes de zero;
  - A dispersão resultante pode compesar sua ineficiência computacional;
- Mas se você acha que todos os seus recursos são importantes, L2 é provavelmente é uma escolha melhor;

2.11 Matriz de confusão

Às vezes, precisão não mostra tudo

- Um teste para uma doença rara pode ter 99.9% de precisão, bastando adivinhar "não" o tempo todo;
- Precisamos entender os verdadeiros positivos e verdadeiros negativos, bem como os falsos positivos e falsos negativos;
- Uma matriz de confusão mostra isso.

	SIM real	Não real
Previsto SIM	VERDADEIROS POSITIVOS	FALSOS POSITIVOS
Previsto NÃO	FALSOS NEGATIVOS	VERDADEIROS NEGATIVOS

Figure 1: Matriz de confusão binária.

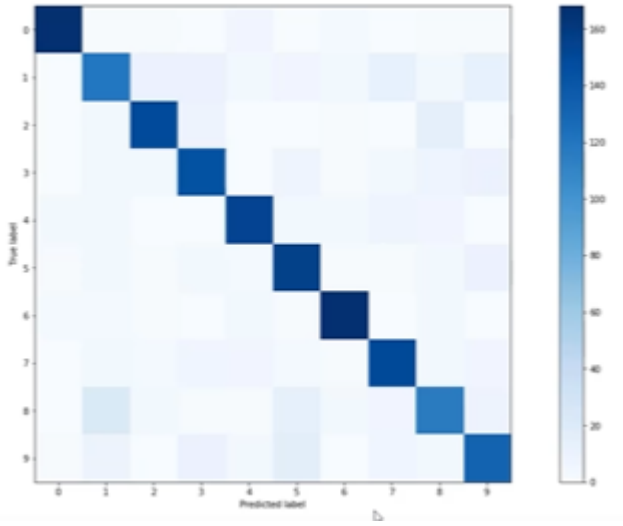

Figure 2: Matriz de confusão multiclasse visualizada a partir de um heatmap.

Matriz de confusão binária

Matriz de confusão multiclasse + heatmap

2.12 Métricas de classificação

Recall ou Sensibilidade

$$= \frac{\text{Verdadeiros Positivos}}{\text{Verdadeiros Positivos} + \text{Falsos Negativos}}$$

- Sensbilidade, taxa de verdadeiro positivo, integridade;
- Porcentagem de positivos previstos corretamente;
- Boa escolha de métrica quando você se preocupa muito com falsos negativos;
  - Ou seja, detecção de fraude;

Precision/Precisão

$$= \frac{\text{Verdadeiros Positivos}}{\text{Verdadeiros Positivos} + \text{Falsos Positivos}}$$

- Positivos corretos;
- Porcentagem de resultados relevantes
- Boa escolha de métrica quando você se preocupa muito com falsos positivos;
  - Ou seja, triagem médica, teste de drogas;

Outras métricas

- Especificidade
$$= \frac{\text{VN}}{\text{VN} + \text{FP}} = \text{"Taxa de verdadeiros negativos"}$$
- F1-Score
$$\frac{2\text{VP}}{2\text{VP} + \text{FP} + \text{FN}} = 2\frac{\text{Precisão} \times \text{Recall}}{\text{Precisão} + \text{Recall}}$$
  - Média harmônica de precisão e sensibilidade;
  - Quando você se preocupa com precisão e recall;
- RMSE
  - Raiz do erro quadrático médio;
  - Medição de precisão;
  - Só se preocupa com as respostas certas e erradas;
- Curva ROC
  - Receiver Operating Characteristic Curve (Curva de Característica de Operação do Receptor);
  - Gráfico da taxa de verdadeiro positivo (recall) vs taxa de falso positivo em várias configurações;

  - Os pontos acima da diagonal representam uma boa classificação (melhor do que aleatória);
  - A curva ideal seria apenas um ponto no canto superior esquerdo;
  - Quanto mais "dobrado" em direção ao canto superior esquerdo, melhor;
- Curva AUC
  - A área sob a curva ROC - Area Under the Curve (AUC);
  - Probabilidade de um classificador classificar uma instância positiva escolhida aleatoriamente melhor do que uma negativa escolhida aleatoriamente;
  - ROC AUC de 0.5 é um classificador inútil, já 1.0 é perfeito.
  - Métrica comumente usada pra comparar classificadores;

2.13 Ensemble Learning

Métodos de grupo

- Exemplo comum: floresta aleatória.
  - Árvores de decisão são propensas a overfitting;
  - Então, faça várias árvores de decisão e deixe que todos votem no resultado;
  - Esta é uma floresta aleatória;

Bagging

- Gere N novos conjuntos de treinamento por amostragem aleatória com reposição;
- Cada modelo reamostrado pode ser treinado em paralelo;

Boosting

- As observações são ponderadas;
- Alguns participarão de novos conjuntos de treinamento com mais frequência;
- O treinamento é sequencial;
- Cada classificador leva em consideração o sucesso do anterior;

Bagging vs Boosting

- Boosting:
  - XGBoost é a "última moda";
  - Boosting geralmlente produz melhor precisão;
- Bagging
  - Evita overfitting;
  - Mais fácil de paralelizar;
- Então, depende do seu objetivo;

2.14 Amazon SageMaker

O SageMaker foi desenvolvido para lidar com todo o workflow de ML

- Implantar modelo, avaliar resultados na produção;
- Buscar, limpar e preparar os dados;
- Treine e avalie um modelo;

Treinamento e implantação do SageMaker

- Em termos de arquitetura, como isso funcionará?
  - Você vai ter dados de treinamento, que eles devem estar no S3 do Amazon;
  - Você vai ter o treinamento do seu modelo a partir dos dados de treinamento;
  - O código de treinamento deste modelo vai estar no ECR (Elastic Container Register), que vem num container em Docker;
  - Uma vez o modelo treinado, você vai fazer o deploy deste modelo, e ele também vai ficar no S3, os artefatos desse modelo estará no S3;
  - Então, você pode fazer a implantação e hospedagem desse modelo e aí você vai ter lá seu código de previsão, que também vai vir de um container em docker;
  - Uma vez implantado, o modelo estará pronto para servir uma aplicação do cliente via endpoint;

Notebooks SageMaker podem direcionar o processo

- Instâncias de notebook no EC2 são executadas a partir do console;
  - Acesso a dados S3;
  - Scikit_learn, Spark, Tensorflow;
  - Grande variedade de modelos integrados;
  - Capacidade de acelerar instâncias de treinamento;
  - Capacidade de implantar modelos treinados para fazer previsões em escala;

Preparação de dados no SageMaker

- Os dados devem vir do S3;
  - O formato ideal varia com o algoritmo
    - Geralmente é RecordIO/Protobuf
- Apache Spark integra-se ao SageMaker
- Scikit_learn, numpy, pandas, tudo à sua disposição dentro de um notebook;

Treinamento em SageMaker

- Crie um job de treinamento;
  - URL do bucket S3 com dados de treinamento;
  - Recursos de computação de ML;
  - URL do bucket S3 para saída;
  - Caminho ECR para o código de treinamento;
- Opções de treinamento;
  - Algoritmos de treinamento integrados;
  - Spark MLLib
  - Código Python Tensorflow/MXNet personalizado;
  - Sua própria imagem do Docker;
  - Algoritmo adquirido do Marketplace do AWS;

Impantando modelos treinados

- Salve seu modelo treinado no S3;
- Pode implantar de duas maneiras:
  - Endpoint persistente para fazer previsões individuais sob demanda;
  - SageMaker Batch Transform para obter previsões para todo um conjunto de dados;
- Muitas opções legais:
  - Pipelines de inferência para processamento mais complexo;
  - SageMaker Neo para implantação em dispositivos na ponta;
  - Elastic Inference para acelerar modelos de aprendizado profundo;
  - Escalonamento automático (aumente o número de endpoints conforme necessário);

2.15 Algoritmos integrados do SageMaker

Aprendizado linear

- Regressão Linear
  - Ajusta uma linha para seus dados de treinamento (linha de regressão);
  - Predicações baseadas nessa linha;
- Pode lidar com previsões de regressão (númericas) e classificação;
  - Para classificação, uma função de limite linear é usada;
  - Pode ser binária ou multiclasse;
- Que tipo de treinamento se espera?
  - RecordIO/Protobuf;
    - Somente dados Float32;
  - CSV;
    - A primeira coluna é considerada o rótulo;

  - Suporta o modo file ou pipe;
    - File: copia todos os dados de treino em um único arquivo;
    - Pipe: stream do S3 conforme necessário - mais eficiente;

- Como é usado?
  - Pré-processando:
    - Os dados de treinamento devem ser normalizados (para que todos os atributos tenham o mesmo peso);
    - O linear learner pode fazer isso para você automaticamente;
    - Os dados de entrada devem ser "misturados";
  - Treinamento:
    - Usa stochastic gradient descent;
    - Escolha um algoritmo de otimização (Adam, AdaGrad, SGD, etc.);
    - Vários modelos são otimizados em paralelo;
    - Regularização L1, L2 para ajuste;
  - Validação:
  - O modelo mais otimizado é selecionado;

- Hiperparâmetros importantes:
  - Balance_multiclass_weights
    - Dá a cada classe igual importância nas funções de perda (loss function);
  - Learning_rate, mini_batch_size;
  - L1;
    - Regularização;
  - Wd;
    - Weight decay (Regularização L2);
  - Aprendizado linear: tipos de instância
    - Treinamento
      - CPU ou GPU de uma ou várias máquinas;
      - Multi-GPU não melhora a performance do modelo;

2.16 XGBoost

Para que serve?

- O eXtreme Gradient Boosting:
  - Grupo de árvores de decisão;
  - Novas árvores feitas para corrigir os erros das árvores anteriores;
  - Usa gradient descent para minimizar a perda conforme novas árvores são adicionadas;

- Tem ganhado muitas competições do Kaggle;
  - E também é muito rápido;

- Pode ser usado para classificação;

- E também para regressão;
  - Usando árvores de regressão;

Que tipo de treinamento ele espera?

- XGBoost não foi feito para o SageMaker, XGBoost Open Source;
- Portanto, é necessária a entrada CSV ou lbsvm;
- O AWS recentemente melhorou para aceitar também RecordIO/Protubuf e Parquet;

Como é usado?

- Os modelos são serializados/desserializados com Pickle;
- Pode ser usado comom uma estrutura em notebooks;
  - Sagemaker.xgboost
- Ou como um algoritmo SageMaker integrado;

Hiperparâmetros importantes

- Existem muitos, alguns exemplos são:
  - Subsample:
    - Previne o overfitting;
  - ETA:
    - Reduz o tamanho do step, evitando overfitting;
  - Gama:
    - Redução de perda mínima para criar uma partição; maior=mais conservador;
  - Alfa:
    - Termo de regularização L1; maior = mais conservador;
  - Lambda:
    - Termo de regularização L2; maior = mais conservador;

Tipos de instância

- Usa apenas CPU;
- É limitado pela memória, não pelo computador;
- Então, M4 é uma boa opção;

2.17 Seq2Seq

Para que serve?

- A entrada é uma sequência de tokens, a saída é uma sequência de tokens;
- Tradução por máquina;
- Resumo de texto;
- Fala para texto;
- Implementado com RNNs e CNNs

Que tipo de treinamento ele espera?

- RecordIO/Protobuf:
  - Os tokens devem ser inteiros (isso é incomum pois a maioria dos algoritmos espera dados de pontos flutuante);
- Comece com arquivos de texto tokenizados;
- Converter para protobuf usando código de exemplo:
  - Pacotes de tensores inteiros com arquivos de vocabulário;
  - Muito parecido com o lab TF/IDF que fizemos anteriormente;
- Deve fornecer dados de treinamento, dados de vallidação e arquivos de vocabulário;

Como é usado?

- O treinamento para tradução automática pode levar dias, mesmo no SageMaker;
- Modelos pré-treinados estão disponíveis:
  - Veja o notebook de exemplo;
- Conjunto de dados de treinamento públicos estão disponíveis para tarefas de tradução específicas;

Hiperparâmetros importantes

- Batch_size;
- Optimizer_type(adam, sgd, rmsprop);
- Learning_rate;
- Num_layers_encoder;
- Num_layers_decoder;
- Pode otimizar em:
  - Precisão:
    - Vs. conjunto de dados de validação fornecido
  - BLEU score:
    - Compara com várias traduções de referência
  - Perplexity:
    - Cross-entropy

Tipos de instância

- Só pode usar tipos de instância de GPU (P3 por exemplo)
- Pode usar uma única máquina para treinamento;
  - Mas pode usar multi-GPUs em uma máquina;

2.18 DeepAR

Para que serve?

- Previsão de dados de série temporal unidimensionais;
- Usa RNN's;
- Permite treinar o mesmo modelo em várias séries temporais relacionadas;
- Encontra frequências e sazonalidade;

Que tipo de treinamento ele espera?

- JSON
  - Gzip ou Parquet;
- Cada registro deve conter:
  - Início: a hora de início;
  - Objetivo: os valores da série temporal;
- Cada registro pode conter:
  - Dynamic_feat: recursos dinâmicos (como uma promoção aplicada a um produto em uma série temporal de compras de produtos);
  - Cat: características categóricas;

Como é usado?

- Sempre inclua séries temporais inteiras para treinamento, teste e inferência;
- Use o conjunto de dados inteiro como conjunto de treinamento, remova os últimos dados de tempo para teste. Avalie os valores retidos;
- Não use valores muito grandes para previsão (>400)
- Treine em várias séries temporais e não apenas em uma, quando possível;

Hiperparâmetros importantes

- Context_length
  - Número de dados temporais que o modelo vê antes de fazer uma previsão;
  - Pode ser menor do que sazonalidades; o modelo vai ter um atraso de um ano de qualquer maneira;
- Epochs;
- Mini_batch_size;
- Learning_rate;
- Num_cells;

Tipos de instância

- Pode usar CPU ou GPU;
- Máquina única ou multi-máquina;
- Comece com CPU (C4.2xlarge, C4.4xlarge);
- Mova para GPU, se necessário;
  - Só ajuda com modelos maiores;
- Apenas CPU para inferência;
- Pode precisar de instâncias maiores para ajuste;

2.19 BlazingText

Para que serve?

- Classificação de texto:
  - Prever textos para uma frase;
  - Útil em pesquisas na web, recuperação de informações;
  - Supervisionado;
- Word2vec:
  - Cria uma representação vetorial de palavras;
  - Palavras com semânticas semelhantes são representadas por vetores próximos uns dos outros;
  - Isso é chamado de incorporação de palavras;
  - É útil para PLN, mas não é um algoritmo de PLN em si;
  - Usado em tradução automática, análise de sentimento;
  - Funciona com palavras individuais, mas não com frases ou documentos;

Que tipo de treinamento ele espera?

- Para modo supervisionado (classificação de texto):
  - Uma frase por linha;
  - A primeira "palavra" na frase é a string __label__ seguida pelo rótulo de fato da sentença;
- Além disso, "augmented manifest text format";
- O Word2vec espera apenas um arquivo de texto com uma frase de treinamento por linha;

Como isso é usado?

- Word2vec tem várias formas:
  - Cbow (Bag of words);
  - Skip-gram;
  - Batch skip-gram:
    - Computação distribuída em muitos nós;

Hiperparâmetros importantes

- Word2vec:
  - Mode (batch_skipgram, skipgram, cbow);
  - Learning_rate;
  - Window_size;
  - Vector_dim;
  - Negative_samples;
- Classificação de texto:
  - Epochs;
  - Learning_rate;
  - Word_ngrams;
  - Vector_dim;

Tipos de instâncias

- Para cbow e skipgram, recomenda-se um único ml.p3.2xlarge;
  - Qualquer CPU ou instância única de GPU funcionará;
- Para batch_skipgram, pode usar uma ou várias instâncias de CPU;
- Para classificação de texto, é recomendado C5 se tiver menos de 2GB de dado de treinamento;
- Para conjuntos de dados maiores, use uma única instância de GPU (ml.p2.xlarge ou ml.p3.2xlarge);

2.20 Object2Vec

Para que serve?

- Lembra do word2vec do BlazingText? É semelhante, mas serve também para objetos arbitrários;
- Ele cria "embeddings" densos de baixa dimensão de objetos de alta dimensão;
- É basicamente um word2vec generalizado, para lidar com outras coisas além de palavras;
- Calcula os vizinhos mais próximos de objetos;
- Visualiza clusters;
- Previsão de gênero;
- Recomendações (itens ou usuários semelhantes)
- Não supervisionado;

Que tipo de treinamento ele espera?

- Os dados devem ser tokenizados em inteiros;
- Os dados de treinamento consistem em pares de tokens e/ou sequências de tokens;
  - Sentença-sentença;
  - Sequência de rótulos (gênero para descrição);
  - Cliente-cliente;
  - Produto-produto;
  - Item de usuário;

Como é usado?

- Processa dados em linhas JSON e os embaralha;
- Treina com dois canais de entrada, dois encoders e um comparador;
- Opções de encoders:
  - Embeddings com pool médio;
  - CNN's
  - LSTM bidirecional;
- O comparador é seguido por uma rede neural feed-forward;

Hiperparâmetros importantes

- Os usuais de deep learning;
  - Dropout, early stopping, epochs, learning rate, batch size, layers, activation function, optimizer, weight decay;
- Enc1_network, enc2_network
  - Opções de hcnn, bilstm, pooled_embedding;

Tipos de instância

- Só pode treinar em uma única máquina (CPU ou GPU, multi-GPU)
  - Ml.m5.2xlarge;
  - Ml.p2.xlarge;
  - Se necessário, vá até ml.m5.4xlarge ou ml.m5.12xlarge;
- Inferência: use ml.p2.2xlarge;
  - Use a variável de ambiente INFERENCE_PREFERRED_MODE para otimizar para embeddings do codificador em vez de classificação ou regressão;

2.21 Detecção de objetos

Para que serve?

- Identifique todos os objetos de uma imagem com caixas delimitadoras;
- Detecta e classifica objetos com uma única rede neural profunda;
- As classes são acompanhadas por pontuações de confiança;
- Pode treinar do zero ou usar modelos pré-treinados com base na ImageNet;

Que tipo de treinamento ele espera?

- RecordIO ou formato de imagem (Jpg ou Png)
- Com o formato de imagem, fornça um arquivo JSON para dados de anotação para cada imagem;

Como isso é usado?

- Obtém uma imagem como entrada, produz todas as instâncias de objetos na imagem com categorias e pontuações de confiança;
- Usa uma CNN com o algoritmo Single Shot Multibox Detector (SSD);
  - O CNN básico pode ser VGG-16 ou ResNet-50;
- Transfer Learning/Treinamento Incremental;
  - Use um modelo pré-treinado para os pesos básicos da rede, em vez de pesos iniciais aleatórios;
- Usa flip, jitter e rescale internamente para evitar o overfitting;

Hiperparâmetros importantes

- Mini-batch_size
- Learning_rate
- Optimizer
  - Sgd, adam, rmsprop, adadelta

Tipos de instâncias

- Use instâncias de GPU para treinamento (multi-GPU e multi-máquina)
  - ml.p2.xlarge, ml.p2.8xlarge, ml.p2.16xlarge, ml.p3.2xlarge, ml.p3.8xlarge, ml.p3.16xlarge;
- Use CPU para inferência;
  - C5, M5, P2, P3 todos ok;

2.22 Classificação de imagens

Para que serve?

- Atribuir um ou mais rótulos a uma imagem;
- Não diz onde os objetos estão, apenas quais objetos estão na imagem;

Que tipo de treinamento ele espera?

- Apache MXNet RecordIO:
  - Não o protobuf;
  - Isso é para interoperabilidade com outras estruturas de aprendizado profundo;
- Ou imagens jpg ou png;
- O formato de imagem requer arquivos .lst para assosciar índice de imagem, rótulo de classe e caminho para a imagem;
- O formato "Augmented Manifest Image" habilita o modo Pipe;

Como é usada?

- ResNet CNN nos bastidores;
- Modo de treinamento completo;
  - Rede inicializada com pesos aleatórios;
- Modo de transferência de aprendizagem;
  - Inicializado com pesos pré-treinados;
  - A camada superior totalmente conectada é inicializada com pesos aleatórios;
  - A rede é ajustada com novos dados de treinamento;
- O tamanho padrão da imagem é de 3 canais 224x224 (conjunto de dados da imageNet);

Hiperparâmetros importantes

- Os de costume para aprendizado profundo:
  - Batch size, learning rate, optimizer
- Parâmetros específicos do otimizador
  - Weight decay, beta 1, beta 2, eps, gamma;

Tipos de instância

- Instâncias de GPU para treinamento (P2, P3) multi-GPU e multi-máquina OK;
- CPU ou GPU para inferência (C4, P2, P3);

2.23 Segmentação semântica

Para que serve?

- Classificação de objetos em nível de píxel;
- Diferente da classificação de imagem, que atribui rótulos a imagens inteiras;
- Diferente da detecção de objetos, que atribui rotulos a caixa delimitadoras;
- Útil para veículos autônomos, diagnósticos por imagens médicas, detecção de robôs;
- Produz uma máscara de segmentação;

Que tipo de treinamento ele espera?

- Imagens JPG ou anotações PNG;
- Para treinamento e validação;
- Formato Augmented Manifest Imagem suportado para o modo Pipe;
- Imagens JPG aceitas para inferência;

Como é usada?

- Construído em MXNet Gluon e Gluon CV;
- Escolha entre 3 algoritmos:
  - Fully-conbolutional Network (FCN);
  - Pyramid Scene Parsing (PSP);
  - DeepLabV3;
- Escolha entre backbones:
  - ResNet50;
  - ResNet101;
  - Ambos treinados na ImageNet;
- Treinamento incremental, treinamento do zero também suportado;

Hiperparâmetros importantes

- Epochs, learning rate, batch size, optimizer, etc.;
- Algoritmo;
- Backbone;

Tipos de instância

- Apenas GPU para treinamento (P2 ou P3) em uma única máquina apenas;
  - Especificamente ml.p2.xlarge, ml.p2.8xlarge, ml.p2.16xlarge, ml.p3.2xlarge, ml.p3.8xlarge, ml.p3.16xlarge;
- Inferência na CPU (C5 ou M5) ou GPU (P2 ou P3);

2.24 Random Cut Forest

Para que serve?

- Detecção de anomalia;
- Não supervisionado;
- Detecta picos inesperados nos dados de uma série temporal;
- Divide em periodicidade;
- Pontos de dados não classificáveis;
- Atribui uma pontuação de anomalia a cada dado;
- Baseado em um algoritmo desenvolvido pela Amazon (do qual eles se orgulham muito!);

Que tipo de treinamento espera?

- RecordIO/Protobuf ou CSV;
- Pode usar modo File ou Pipe para ambos;
- Canal de teste opcional para calcular acurácia, precisão, recall e F2 em dados rotulados (anomalia ou não);

Como é usado?

- Cria uma floresta de árvores onde cada árvore é uma partição dos dados de treinamento; olha para a mudança esperada na complexidade da árvore como resultado da adição de um dado nela;
- Os dados são amostrados aleatoriamente;
- Então é treinada;
- O RCF também aparece no Kinesis Analytics, ele também pode funcionar em streaming de dados;

Hiperparâmetros importantes

- Num_trees;
  - Aumentar reduz o ruído;
- Num_samples_per_tree;
  - Deve ser escolhido de forma que 1/num_samples_per_tree se aproxime da proporção de dados entre anômalos e normais;

Tipos de instância

- Não tem vantagem usar GPUs;
- Use M4, C4 ou C5 para treinamento;
- ml.c5.xl para inferência;

2.25 Neural Topic Model

Para que serve?

- Organiza documentos em tópicos;
- Classifica ou resume documentos com base em tópicos;
- Não é apenas TF/IDF;
  - "Bicicleta", "carro", "trem", "quilometragem" e "velocidade" podem classificar um documento como "transporte", por exemplo (embora não seja rotulado exatamente desta maneira);
- Sem supervisão;
  - Algoritmo é "Inferência Neural Variacional";

Que tipo de treinamento ele espera?

- Quatro canais de dados:
  - "Treinar" é necessário;
  - "Validação", "teste" e "auxiliar" opcional;
- RecordIO/Protobuf ou CSV;
- As palavras devem ser convertidas em números inteiros;

  - Cada documento deve conter uma contagem para cada palavra do vocabulário em CSV;
  - O canal "auxiliar" é para vocabulário;
- Modo file ou pipe;

Como é usado?

- Você define quantos tópicos deseja;
  - Isto é um hiperparâmetro que você define quando irá configurar o algoritmo;
- Esses tópicos são uma representação latente com base nas palavras mais bem classificadas;
- Um dos dois algoritmos de modelagem de tópicos no SageMaker - você pode tentar os dois;

Hiperparâmetros importantes

- Reduzir mini_batch_size e learning_rate pode reduzir validation_loss
  - À custa do tempo de treinamento;
- Num_topics;

Tipos de instâncias

- GPU ou CPU
  - GPU é recomendado para treinamento;
  - CPU para inferência;
  - CPU é mais barato;

2.26 LDA

Para que serve?

- Latent Dirichlet Allocation;
- Outro algoritmo de modelagem de tópico;
  - Não é deep learning;
- Sem supervisão;
  - Os próprios tópicos não estão rotulados;
  - Eles são apenas agrupamentos de documentos com um subconjunto compartilhado de palavras;
- Pode ser usado para outras coisas além de palavras;
  - Agrupar clientes baseados em suas compras;
  - Análise harmônica na música;

Que tipo de treinamento ele espera?

- Canal de treinamento, canal de teste opcional;
- RecordIO/Protobuf ou CSV;
- Cada documento tem contagens para cada palavra do vocabulário (em formato CSV);
- Modo Pipe suportado apenas com RecordIO;

Como é usado?

- Sem supervisão: gera quantos tópicos você especificar;
- O canal de teste opcional pode ser usado para pontuar os resultados;
  - Probabilidade de registro por palavra;
- Funcionalmente semelhante ao NTM, mas baseado em CPU;
  - Portanto, pode ser mais barato/mais eficiente;

Hiperparâmetros importantes

- Num_topics;
- Alpha0;
  - Estimativa inicial para o parâmetro de concentração;
  - Valores menores geram misturas de tópicos esparsos;
  - Valores maiores (>1.0) produzem misturas uniformes;

Tipos de instância

- Treinamento de CPU de instância única;

2.27 KNN

Para que serve?

- K-Nearest-Neighbours;
- Algoritmo de Classificação ou Regressão Simples;
- Classificação;
  - Encontre os K pontos mais próximos de um ponto de amostra e retorne o rótulo mais frequente;
- Regressão;
  - Encontre os K pontos mias próximos de um ponto de amostra e retorne o valor médio;

Que tipo de treinamento ele espera?

- O canal de treinamento contém seus dados;
- O canal de teste mede acurácia ou MSE;
- Treinamento com recordIO/protobuf ou CSV;
  - A primeira coluna é o rótulo;
- Modo file ou pipe em ambos;

Como é usado?

- Os dados são primeiro amostrados;
- O SageMaker inclui um estágio de redução de dimensionalidade;
  - Evita dados esparsos ("maldição da dimensionalidade");
  - A custo de ruído/precisão;
  - Métodos de "sign" ou "fjlt";
- Cria um índice para procurar vizinhos;
- Serializa o modelo;
- Consulta o modelo para um determinado K;

Hiperparâmetros importantes

- K;
- Sample_size;

Tipos de instância

- Treinamento de CPU ou GPU;
  - ml.m5.2xlarge;
  - ml.p2.xlarge;
- Inferência;
  - CPU para menor latência;
  - GPU para maior rendimento em grandes batchs;

2.28 K-Means

Para que serve?

- Agrupamento não supervisionado;
- Divida os dados em K grupos, onde os membros de um grupo são semelhantes quanto possível entre si;
  - Você define o que "semelhante" significa;
  - Medida pela distância euclideana;
- Clustering K-Means em escala da web;

Que tipo de treinamento ele espera?

- Canal de treinamento, teste opcional;
  - Treino ShardedByS3Key, teste FullyReplicated;
- RecordIO/protobuf ou CSV;
- File ou Pipe em qualquer um;

Como é usado?

- Cada observação mapeada para o espaço n-dimensional (n=número de características);
- Trabalha para otimizar o centro dos k clusters;
  - "Centros dos clusters extras" podem ser especificados para melhorar a precisão (que acabam sendo reduzidos para k);
  - $K = k * x$;
- Algoritmo:
  - Determina os centros de cluster iniciais;
    - Abordagem aleatória ou k-means ++
    - K-means ++ tenta separar os clusters iniciais;
- Execute interação nos dados de treinamento e calcule os centros de cluster;
- Reduza os clusters de K para k;
  - Usando o método Lloyd's com kmeans ++;

Hiperparâmetros importantes

- K
  - Escolher K é complicado;
  - Use o método elbow;
  - Plotar a soma dos quadrados dentro do cluster em função de K;
  - Otimiza as dimensões dos clusters;
- Mini_batch_size;
- Extra_center_factor;
- Init_method;

Tipos de instância

- CPU ou GPU, mas CPU é recomendada;
  - Apenas uma GPU por instância usada na GPU;
  - Portanto, use p*.xlarge se for usar GPU;

2.29 PCA

Para que serve?

- Principal Component Analysis;
- Redução de dimensionalidade;
  - Projete dados com alta dimensionalidade (muitos atributos) em dimensões inferiores (como um gráficos 2D) enquanto minimiza a perda de informações;
  - As dimensões reduzidas são chamadas de componentes;
  - O primeiro componente tem a maior variabilidade possível;
  - O segundo componente tem a próxima maior e assim por diante;
- Sem supervisão;

Que tipo de treinamento ele espera?

- RecordIO/Protobuf ou CSV;
- File ou Pipe ou ambos;

Como é usado?

- Matriz de covariância é criada, então usa uma técnica de singular value decomposition (SVD);
- Dois modos;
  - Regular:
    - Para dados esparsos e número moderado de observações e recursos;
  - Randomizado:
    - Para um grande número de observações e atributos;
    - Usa algoritmo de aproximação;

Hiperparâmetros importantes

- Algorithm_mode;
- Subtract_mean;
  - Dados não-enviesados;

Tipos de instância

- GPU ou CPU
  - Depende "das especificações dos dados de entrada";

2.30 Factorization Machines

Para que serve?

- Lida com dados esparsos;
  - Previsão de clique;
  - Recomendações de itens;
  - Como um usuário individual não interage com a maioria das págainas / produtos, os dados são esparsos;
- Supervisionado
  - Classificação ou regressão
- Limitado a interação em pares;
  - Usuário -> item por exemplo;

Que tipos de treinamento ele espera?

- RecordIO/Protobuf com Float32;
  - Dados esparsos significam que CSV não é prático;

Como é usado?

- Encontra fatores que podemos usar para prever uma classificação (clique ou não? Compra ou não?) ou valor (avaliação prevista?). Dada uma matriz que representa algumas coisas (usuários e itens?);
- Normalmente usado no contexto de sistemas de recomendação;

Hiperparâmetros importantes

- Métodos de inicialização para bias, fatores e termos lineares;
  - Uniforme, normal ou constante;
  - Pode ajustar as propriedades de cada método;

Tipos de instância

- CPU ou GPU
  - CPU é recomendado;
  - GPU só funciona com dados densos;

2.31 IP Insights

Para que serve?

- Aprendizagem não supervisionada de padrões de uso de endereço IP;
- Identifica comportamento suspeito de endereços IP;
  - Identifica logins de IPs anômalos;
  - Identifique contas que criam recursos de IPs anômalos;

Que tipo de treinamento ele espera?

- Nomes de usuários e IDs de contas podem ser inseridos diretamente;
- Não há necessidade de pré-processamento;
- Canal de treinamento, validação opcional (calcula a pontuação AUC);
- CSV apenas;
  - Entity, IP;

Como é usado?

- Usa uma rede neural para aprender representações vetoriais latentes de entidades e endereços IP;
- Entidades são transformadas em hashs;
  - Hash devem ter um tamanho grande o suficiente;
- Gera automaticamente amostras negativas durante o treinamento, pareando aleatoriamente entidades e IPs;

Hiperparâmetros importantes

- Num_entity_vectors
  - Tamanho do hash;
  - Definido para duas vezes o número de identificadores de entidade exclusivos;
- Vector_dim;
  - Tamanho dos vetores de incorporação;
  - Ajusta o tamanho do modelo;
  - Muito grande gera overfitting;
- Epochs, learning rate, batch size, etc.;

Tipos de instância

- CPU ou GPU
  - GPU é recomendado;
  - ml.p3.2xlarge ou superior;
  - Pode usar várias GPUs;
  - O tamanho da instância da CPU depende de vector_dim e num_entity_vectors;

2.32 Aprendizagem por reforço

Introdução

- Na aprendizagem por reforço, você tem um tipo de agente que "explora" algum espaço;
- À medida que avança, aprende o valor de diferentes mudanças de estado em diferentes condições;
- Esses valores informam o comportamento subsequente do agente;
- Exemplos: jogo Pac-Man, Cat & Mouse (Jogo AI);
  - Gestão da cadeia de suprimento;
  - Sistema HVAC;
  - Robótica industrial;
  - Sistemas de diálogo;
  - Veículos autônomos;
- Leva a um desempenho on-line mais rápido depois que o espaço foi explorado;

Q-Learning

- Uma implementação específica de aprendizagem por reforço;
- Você tem:
  - Um conjunto de estados do ambiente;
  - Um conjunto de ações possíveis nesses estados - s;
  - Um valor para cada estado / ação - Q

- Comece com valores $Q = 0$;
- Explore o espaço;
- Quando coisas ruins acontecerem após um determinado estado/ação, reduza seu Q;
- À medida que as recompensas acontecem após um determinado estado/ação, aumente seu Q;
- Quais são alguns estados/ações aqui?
  - Pac-man tem um muro a esquerda;
  - Pac-man morre se der um passo para baixo;
  - Pac-man continua vivo se for para cima, ou direita;
- Você pode "olhar para frente" mais de uma etapa usando um fator de desconto ao calcular Q (aqui s é o estado anterior, s' é o estado atual);
- $Q(s,a) += \text{desconto} * (\text{recompensa}(s,a) + \max(Q(s')) - Q(s,a))$

O problema da exploração

- Como podemos explorar com eficiência todos os estados possíveis?
  - Abordagem simples: sempre escolha a ação para um determinado estado com o Q mais alto. Se houve empate, escolha aleatoriamente;
    - Mas isso é realmente ineficiente, e você pode perder muitos caminhos dessa forma;
- Melhor maneira: introduza um elemento épsilon;
  - Se um número aleatório for menor que épsilon, não siga o Q mais alto, mas escolha aleatoriamente;
  - Dessa forma, a exploração nunca para totalmente;
  - Escolher épsilon pode ser complicado;

Palavras complicadas

- Markov Decision Process
  - Da wikipedia: os processos de decisão de Markov (MDPs) fornecem uma estrutura matemática para modelar a tomada de decisão em situações em que os resultados são parcialmente aleatórios e parcialmente sob o controle de um tomador de decisão;
  - Soa familiar? MDPs são apenas uma maneira de descrever o que acabamos de ver usando notação matemática;
  - Os estados ainda são descritos como s e s';
  - As funções de transição de estado são descritas como $P_a(s,s')$;
  - Nossos valores "Q" são descritos como uma função de recompensa $R_a(s,s')$;
- Uma MDP é um processo de controle estocástico de tempo discreto;

Para recapitular

- Você pode fazer um pac-man inteligente em algumas etapas:
  - Faça com que ele explore de forma semi-aleatória diferentes opções de movimento (ações) dadas as diferentes condições (estados);
  - Acompanhe a recompensa ou penalidade associada a cada escolha para um determinado estado/ação (Q);
  - Use esses valores Q armazenados para informar suas escolhas futuras;
- Conceito muito simples. Agora você pode dizer que compreende o aprendizado por reforço, o Q-learning, os processos de decisão de Markov e a programação dinâmica;

Aprendizagem por reforço no SageMaker

- Usa uma estrutura de aprendizado profundo com Tensorflow e MXNet;
- Suporta Intel Coach e Ray Rlib toolkits;
- Suporta ambientes personalizados, de código aberto ou comerciais;
  - MATLAB, Simulink;
  - EnergyPlus, RoboSchool, PyBullet;
  - Amazon Sumerian, AWS RoboMaker;

Treinamento distribuído com SageMaker RL

- Pode distribuir treinamento e/ou implanatação de ambiente;
- Multi-core e multi-instância;

Termos chave

- Ambiente: O layout do tabuleiro, labirinto, etc.
- Estado: onde o jogador/peças estão;
- Ação: Mova-se em uma determinada direção, etc.;
- Recompensa: Valor associado à ação daquele estado;
- Observação: ou seja, arredores em um labirinto, estado de tabuleiro de xadrez;

Principais hiperparâmetros

- Os parâmetros de sua escolha podem ser abstraídos;
- O ajuste de hiperparâmetros no SageMaker pode otimizá-los;

Tipos de instância

- Nenhuma orientação específica fornecida no guia do desenvolvedor;
- Mas é um aprendizado profundo, então as GPUs são melhores;
- E sabemos que ele oferece suporte a várias instâncias e núcleos;

2.33 Ajuste automático de modelo no SageMaker

Ajuste de hiperparâmetros

- Como você sabe os melhores valores de learning rate, batch size, depth, etc?
- Frequentemente você tem que experimentar;
- O problema cresce rapidamente quando você tem muitos hiperparâmetros diferentes; precisa experimentar todas as combinações de todos os valores possíveis de alguma forma, treinar modelo e avaliá-lo;

Ajuste automático de modelo

- Defina os hiperparâmetros de seu interesse, os intervalos que deseja testar e as métricas para as quais está otimizando;
- O SageMaker cria um "Job de ajuste de parâmetros" que treina quantas combinações você configurar;
  - Instâncias de treinamento são aceleradas conforme necessário, potencialmente muitas delas;
- O conjunto de hiperparâmetros que produzir os melhores resultados pode então ser implantados como um modelo;
- Aprende à medida que avança, por isso, não tem de tentar todas as combinações possíveis;

Melhores práticas

- Não otimize muitos hiperparâmetros de uma vez;
- Limite seus intervalos ao menor intervalo possível;
- Use escalas logarítimicas quando apropriado;
- Não execute muitos jobs de treinamento simultaneamente;
  - Isso limita a capacidade do processo aprender à medida que avança;
- Certifique-se de que os jobs de treinamento em execução em várias instâncias gerem a métrica correta no final;

2.34 SageMaker e Spark

Integrando o SageMaker e Spark

- Pré-processe os dados normalmente com o Spark;
  - Gerar DataFrames;
- Use a biblioteca sagemaker-spark;
- SageMakerEstimator;
  - KMeans, PCA, XGBoost;
- SageMaker Model;
- Conecte o notebook a um cluster EMR remoto executando Spark (ou use o Zeppelin);

- O dataframe de treinamento deve ter:
  - Uma coluna de atributos que é um vetor de doubles;
  - Uma coluna opcional de rótulos de doubles;
- Chame fit em seu SageMakerEstimator para obter um SageMakerModel;
- Chame transform no SageMakerModel para fazer inferências;
- Também funciona com Spark Pipelines;

Por que se importar?

- Permite combinar pré-processsamento de big data no Spark com treinamento e inferência no SageMaker;

2.35 Novos Recursos do SageMaker

SageMaker Studio

- IDE visual para machine learning;
- Integra muitos dos recursos que estamos prestes a abordar;

SageMaker Notebooks

- Crie e compartilhe notebooks Jupyter com SageMaker Studio;
- Alterna entre as configurações de hardware (sem infraestrutura para gerenciar);

SageMaker Experiments

- Organiza, captura, compara e pesquisa seus Jobs de ML;

SageMaker Debugger

- Salva o estado do modelo interno em intervalos periódicos;
  - Gradientes/tensores ao longo do tempo conforme um modelo é treinado;
  - Defina regras para detectar condições indesejadas durante o treinaamento;
  - Um trabalho de depuração é executado para cada regra que você configurar;
  - Registra e dispara um evento CloudWatch quando a regra é atingida;

SageMaker Autopilot

- Automatiza:
  - Seleção de algoritmo;
  - Pré-processamento de dados;
  - Ajuste de modelo;
  - Toda infraestrutura;
- Faz todas as tentativas e erros para você;
- Em geral, é conhecido como AutoML;

SageMaker Model Monitor

- Receba alertas sobre mudanças de qualidade em seus modelos implantados
- Visualize as mudanças de dados;
  - Exemplo: o modelo de empréstimo começa a dar às pessoas mais crédito devido a desvios ou falta de recursos de entrada;
- Nenhum código necessário;

2.36 Serviços de IA/ML de Alto Nível

Amazon Comprehend

- Processamento de linguage natural e análise de texto;
- Insira mídias sociais, e-mails, páginas da web, documentos, transcrições, registros médicos (Comprehend Medical);
- Extrai "frases-chave", entidades, sentimentos, linguagem, sintaxe, tópicos e classificações de documentos;
- Pode treinar com seus próprios dados;

Amazon Translate

- Usa aprendizado profundo para tradução;
- Suporta terminologia personalizada;
  - Em formato CSV ou TMX;
  - Apropriado para nomes próprios, marcas e etc.;

Amazon Transcribe

- Fala para texto
  - Entrada em FLAC, MP3, MP4 ou WAV, em um idioma específico;
  - Streaming de áudio compatível (HTTP/2 ou WebSocket);
    - Francês, inglês, espanhol, português e outros;
- Identificação do locutor;
  - Especifique o número de locutores;
- Identificação de canal;
  - Ou seja, dois locutores podem ser transcritos separadamente mesclando com base no tempo de "enunciados";
- Vocabulários personalizados;
  - Listas de vocabulário (uma lista de palavras especiais, nomes, acrônimos);
  - Tabelas de vocabulário (podem incluir "SoundsLike", "IPA" e "DisplayAs");

Amazon Polly

- Transforma texto para fala neural, muitas vozes e idiomas;
- Lexicons;
  - Personalize a pronúncia de palavras e frases específicas;
  - Exemplo: “World Wide Web Consortium” em vez de “W3C”;
- SSML;
  - Alternativa para texto simples;
  - Linguagem de marcação de síntese de fala;
  - Dá controle sobre a ênfase, a pronúncia, a respiração, o sussuro, a velocidade da fala, o tom, as pausas;
- Marcas de fala (Speech Marks);
  - Pode codificar quandao a frase/palavra começa e termina no stream de áudio;
  - Útil para animação com sincronização labial;

Rekognition

- Visão computacional;
- Detecção de objeto e cena;
  - POde usar sua própria coleção de rosto;
- Moderação de imagem;
- Análise facial;
- Reconhecimento de celebridades;
- Comparação de rosto;
- Texto na imagem;
- Análise de vídeo;
  - Objetos/pessoas/celebridades maracdos na linha do tempo;
  - People pathing;
- As imagens vêm do S3 ou fornecer bytes de imagens como parte da solicitação;
  - S3 será mais rápido se a imagem já estiver lá;
- O reconhecimento facial depende de boa iluminação, ângulo, visibilidade dos olhos, resolução;
- O vídeo deve vir de Streams de vídeo Kinesis;
  - H.264 H.264 encoded;
  - 5-30 FPS;
  - Privilegiar a resolução em vez da taxa de quadros;
- Pode ser usado com labda para acionar análise de imagem durante o upload;

- Rótulos personalizados do Rekognition;
  - Treine com um pequeno conjunto de imagens rotuladas;
  - Use seus próprios rótulos para itens exclusivos;
  - Exemplo: a NFL (National Football League nos EUA) usa rótulos personalizados para identificar logotipos de times, postes e dedos de espuma em imagens;

Amazon Forecast

- Seviço totalmente gerenciado para fornecer previsões altamente precisas com ML;
- "AutoML" escolhe o melhor modelo para seus dados de série temporal;
  - ARIMA, DeepAR, ETS, NPTS, Profeta;
- Funciona com qualquer série temporal;
  - Preço, promoções, performance econômica, etc.;
  - Pode combinar com dados associados para encontrar relacionamentos;
- Planejamento de estoque, planejamento financeiro, planejamento de recursos;
- Com base em "grupos de conjuntos de dados", "preditores" e "previsões";

Amazon Lex

- Motor chatbot de linguagem natural;
- Um bot é construído em torno de intents;
  - Enunciados invocam intenções ("Eu quero pedir uma pizza");
  - As funções Lambda são invocadas para cumprir a intenção;
  - Os slots especificam informações extras necessárias para a intenção;
    - Tamanho da pizza, coberturas, tipo de crosta, aonde entregar, etc.
- Pode implantar no AWS Mobile SDK, Facebook Messenger, Slack e Twilio;

Outros serviços de ML

- Amazon Personalize;
  - Sistema de recomendação;
- Amazon Textract;
  - OCR com suporte a formulários, campos e tabelas;
- AWS DeepRacer;
  - Aprendizado por reforço com carro de corrida em escala 1/18;
- DeepLens;
  - Câmera de vídeo habilitada para aprendizado profundo;
  - Integrado com Rekognition, SageMaker, Polly, Tensorflow, MXNet, Caffe;

AWS DeepComposer

- Teclado alimentado por IA;
- Compõe uma melodia em uma música inteira;
- Para fins educacionais;

Amazon Fraud Detector

- Carregue seus próprios dados históricos de fraude;
- Constrói modelos personalizados a partir de um modelo que você escolher;
- Dispõe uma API para seu aplicativo online;
- Avalie o risco de:
  - Novas contas;
  - Check-out de convidado;
  - Abuso de "Experimente antes de comprar";
  - Pagamentos online;

Amazon CodeGuru

- Revisões de código automatizadas;
- Encontra linhas de código que afetam o desempenho;
- Vazamentos de recursos;
- Oferece recomendações específicas;
- Implementado por ML;

Contact Lens para Amazon Connect

- Para centros de atendimento ao cliente;
- Insere dados de áudio de chamadas gravadas;
- Permite pesquisa em chamadas/chats;
- Análise de sentimentos;
- Encontre "expressões" que se correlacionam com chamadas bem-sucedidas;
- Categorizar chamadas automaticamente;
- Meça a velocidade da conversa e as interrupções;
- Detecção de tema: descobre problemas emergentes;

Amazon Kendra

- Pesquisa corporativa com linguagem natural;
- Por exemplo: “Onde fica o balcão de suporte de TI?”, “Como faço para conectar à minha VPN?”;
- Combina dados de sistemas de arquivos, SharePoint, Intranet, serviços de compartilhamento (JDBC, S3) em um repositório pesquisável;
- Com tecnologia ML - usa feedback positivo/negativo;
- Ajuste de relevância - aumenta a intensidade da atualização do documento, contagem de visualizações, etc.;

Amazon Augmented AI (A2I)

- Revisão humana das previsões de ML;
- Cria fluxos de trabalho para revisar previsões de baixa confiança;
- Acesse a Mechanical Turk;
- Integrado ao Amazon Textract e Rekognition;
- Integra-se com o SageMaker;
- Muito semelhante ao Ground Truth;

Juntando tudo

- Construa sua própria Alexa;
  - Transcribe -> Lex -> Polly!
- Faça um tradutor universal;
  - Transcribe -> Translate -> Polly
- Construa um detector de celebridades;
  - DeepLens -> Rekognition;
- As pessoas ao telefone estão felizes?
  - Transcribe -> Compreender;