Professor Eduardo Kugler Viegas
Frameworks de Big Data

Prática Apache Spark

Considerando o caso a seguir, implemente as soluções em Python para extrair o conjunto de informações solicitadas

Descrição:

Você foi contratado por uma empresa para efetuar uma análise de dados. Esta empresa possui acesso a uma base de dados com dados sobre incidentes criminais na cidade de Chicago desde 2001. Nesse cenário, para cada incidente criminal presente na base de dados são fornecidos os seguintes campos:

Campo	Descrição
Dia	Dia da ocorrência
Mês	Mês da ocorrência
Ano	Ano da ocorrência
Bloco	Região da ocorrência
Tipo	Tipo da ocorrência criminal
Descrição	Breve descrição da ocorrência
Descrição da localização	Descrição sobre a localização da ocorrência, e.g. Rua
Latitude	Localização da ocorrência
Longitude	Localização da ocorrência

No total, a base de dados possui mais de 13 milhões de ocorrências criminais. A base de dados foi fornecida no formato CSV, sendo que cada entrada (ocorrência criminal), é representada por uma linha no arquivo. Enquanto cada linha possui os campos listados previamente, estes separados pelo caractere ";". A imagem a seguir exibe as 5 primeiras ocorrências criminais presentes na base.

```
18;03;2015;047XX W OHIO ST;BATTERY;AGGRAVATED: HANDGUN;STREET;41.891398861;-87.744384567
18;03;2015;066XX S MARSHFIELD AVE;OTHER OFFENSE;PAROLE VIOLATION;STREET;41.773371528;-87.665319468
18;03;2015;044XX S LAKE PARK AVE;BATTERY;DOMESTIC BATTERY SIMPLE;APARTMENT;41.81386068;-87.596642837
18;03;2015;051XX S MICHIGAN AVE;BATTERY;SIMPLE;APARTMENT;41.800802415;-87.622619343
18;03;2015;047XX W ADAMS ST;ROBBERY;ARMED: HANDGUN;SIDEWALK;41.878064761;-87.743354013
```

Diante desse contexto, você foi encarregado pelo desenvolvimento de um conjunto de soluções no Apache Spark, que permitam a extração das seguintes informações sobre a base:

1. Quantidade de crimes por ano
2. Quantidade de crimes por ano que sejam do tipo NARCOTICS
3. Quantidade de crimes por ano, que sejam do tipo NARCOTICS, e tenham ocorrido em dias pares;
4. Mês com maior ocorrência de crimes;
5. Mês com menor ocorrência de crimes;
6. Mês por ano com a maior ocorrência de crimes;
7. Mês com a maior ocorrência de crimes do tipo "DECEPTIVE PRACTICE"
8. Dia do ano com a maior ocorrência de crimes;
9. Quantidade de crimes por ano que sejam do tipo NARCOTICS, que ocorreram na localização descrita como STREET
10. Quantidade de crimes por ano que sejam do tipo NARCOTICS, que ocorreram na localização descrita como STREET, no raio de tamanho 2 da latitude 41 e longitude -87;
11. Dia da semana com maior quantidades de ocorrência criminal;
12. Dia da semana com maior quantidades de ocorrência criminal, do tipo "DECEPTIVE PRACTICE";
13. Dia da semana com maior quantidades de ocorrência criminal, do tipo "DECEPTIVE PRACTICE", que a região da ocorrência possua "PARK";
14. Dia da semana com maior quantidades de ocorrência criminal, do tipo "DECEPTIVE PRACTICE", que a região da ocorrência possua "PARK", no raio de tamanho 2 da latitude 41 e longitude -87;
15. Quantidade de ocorrências em 2015 que ocorreram no raio da média e desvio padrão das ocorrências do tipo "DECEPTIVE PRACTICE";

16. A média da latitude e a média da longitude dos crimes que ocorreram em 2015, do tipo NARCOTICS. *Sem utilizar a função mean*();
17. A média da latitude e a média da longitude da base gerada a partir de: 10% primeiros e 10% últimas ocorrências na base;
18. A média da latitude e a média da longitude dos grupos de: 10% primeiros e 10% últimas ocorrências que ocorreram em 2015;
19. Quantidade de ocorrências em 2015 que ocorreram no raio da média e desvio padrão das ocorrências que possuam a palavra "ASSAULT" em seu tipo;
20. A quantidade de ocorrências de acordo com a localização agrupadas em um raio de 0.5;

Com base em seu conhecimento forneça a resposta utilizando o Apache Spark. Para tanto, para cada informação requisitada forneça:

1. **O código fonte da função**
2. **O resultado da função (fornecer apenas os 5 primeiros resultados, quando aplicável)**

Dicas

- *Localização dos arquivos*

Máquina	***Base parcial***	***Base completa***
NameNode	*/data/ocorrencias_criminais/ocorrencias_criminais_sample.csv*	*/data/ocorrencias_criminais/ocorrencias_criminais.csv*
Host	*/home/docker/Desktop/data/ocorrencias_criminais/ocorrencias_criminais_sample.csv*	*/home/docker/Desktop/data/ocorrencias_criminais/ocorrencias_criminais.csv*

- *Copie o código gerado no notebook para o documento*
- *Utilize o arquivo ocorrencias_criminais_sample.csv para testar a sua solução*
- *O resultado deve ser obtido sobre o arquivo base_inteira.csv através* ***da execução no cluster hadoop!***
- *Lembre-se de exibir os resultados ordenados!*

Código Apache Spark

	Função	Entrada	Saída	Descrição
Transformações	map(função)	RDD <Valores>	RDD <Valores>	Executa a função sobre todo valor de entrada, deve gerar um valor de saída
	filter(função)	RDD <Valores>	RDD <Valores>	Executa a função sobre todo valor de entrada, gera um RDD apenas com valores que retornaram True
	flatMap(função)	RDD <Valores>	RDD <Valores>	Executa a função sobre todo valor de entrada, pode gerar 0 ou mais valores de saída
	sample(*withReplacement*, *fraction*)	RDD <Valores>	RDD <Valores>	Gera um RDD com *fraction* % do RDD de entrada
	distinct()	RDD <Valores>	RDD <Valores>	Gera um RDD com valores únicos do RDD de entrada
	groupBy(função)	RDD <Valores>	RDD <Chave, Valores>	Agrupa os valores de acordo com o valor retornado pela função
	groupByKey()	RDD <Chave, Valor>	RDD < Chave, Valores>	Agrupa os valores de acordo com a chave
	reduceByKey(função)	RDD <Chave, Valor>	RDD <Chave, Valor>	Gera um RDD com os valores agrupados de acordo com a chave (função recebe 2 parâmetros)
	sortBy(função)	RDD <Valores>	RDD <Valores>	Gera um RDD ordenado de acordo com os valores retornado pela função
	sortByKey()	RDD < Chave, Valores>	RDD <Chave, Valores>	Gera um RDD ordenado de acordo com a chave
Ações	reduce(função)	RDD <Valores>	Valor	Gera um valor agregando o RDD de entrada (função recebe 2 parâmetros)
	sampleStdev()	RDD <Valores>	Valor	Gera o desvio padrão do RDD de entrada
	max()	RDD <Valores>	Valor	Gera o maior valor do RDD de entrada
	min()	RDD <Valores>	Valor	Gera o menor valor do RDD de entrada
	collect()	RDD <Valores>	Lista (Valores)	Recebe todos os valores do RDD como uma lista no driver
	count()	RDD <Valores>	Inteiro	Recebe um inteiro com a quantidade de elementos no RDD
	first()	RDD <Valores>	Valor	Recebe o primeiro valor do RDD
	take(n)	RDD <Valores>	Lista (Valores)	Recebe os N primeiros valores do RDD em uma lista
	countByKey()	RDD <Chave, Valores>	Dicionário (chave, inteiro)	Recebe um dicionário com a quantidade de valores para cada chave
	saveAsText(caminho)	RDD <Valores>	-	Gera um arquivo no caminho especificado com o RDD de entrada (pode ser escrito no HDFS)