Nanodegree Engenheiro de Machine Learning

Projeto Final

Rodney N. Silva
Dezembro de 2018

I. Definição

Visão Geral

Este projeto analisa dados de turbinas eólicas da Siemens Wind Power, apresentados no Concurso de Análise de Dados na Universidade Central Florida em 2017(DATA MINING PROGRAM-University of Central Florida,2017).Esta análise é importante porque pode minimizar o custo das visitas de manutenção e fornecer informações importantes sobre falhas nas turbinas.Este tema foi escolhido devido ao grande crescimento no número de parques eólicos no Brasil durante a última década. O país já é o oitavo maior país do mundo em capacidade instalada de geração eólica e existem pouquíssimas bases de dados abertas de turbinas eólicas. O Aprendizado de Máquina será utilizado neste problema porque pode prever a duração aproximada de futuras visitas de manutenção, através do treinamento de um modelo com os dados existentes de duração de visitas passadas. A maior parte dos trabalhos científicos existentes na área de manutenção preditiva de aerogeradores se baseia em dados contínuos, como temperatura de componentes da turbina, velocidade do vento e potência instantânea gerada. Porém, o artigo de Lijie(2011) fornece um modelo preditivo para informar o tipo de falha a partir de dados contínuos e discretos: dados de sensores e códigos de eventos. O trabalho mostrado aqui, que contém apenas dados discretos, se baseia no artigo citado acima.

Descrição do Problema

As turbinas são utilizadas em parques eólicos de geração de energia e durante a operação geram alarmes,informações e falhas que podem causar o desligamento da turbina e necessitam de intervenção para religamento.Estas turbinas são visitadas por técnicos,quando necessário, com tempo de viagem entre o prédio de controle e as turbinas de aproximadamente 30 minutos.É importante analisar os dados das turbinas de forma que se identifique se existe algum padrão de falhas em algum local, em determinados tipos de turbina, sazonalidade de falhas ou período do dia em que ocorrem em maior número. Os dados existentes no arquivo principal são:

nome do parque eólico, número da turbina, número da visita, 4 fatores mascarados e relacionados com o tipo de parque(FactorsA-D), horário de início da visita, e duração.Também são fornecidos códigos dos eventos,advertências ou falhas relacionados com cada visita.O código 1020 indica que uma visita está em andamento, ele é acionado por um botão que os técnicos acionam ao chegar no aerogerador para a visita técnica. O volume de dados compreende 171929 registros de eventos em 37 parques eólicos e 641 códigos de evento distintos.

O objetivo principal é investigar através dos dados de cada turbina informações que possam identificar o motivo da visita técnica, apresentando os padrões de códigos exibidos antes,durante ou após a visita e categorizando estes padrões. O segundo objetivo é prever, com base nos códigos anteriores à uma visita e demais informações, o tempo aproximado de duração da manutenção. Para isto é necessário analisar os códigos de eventos, de forma que se busque sequências de códigos indicativas das visitas. Este problema difere de casos semelhantes de manutenção preditiva, onde existe um conjunto de dados com eventos relacionados à falhas e aqueles relacionados à máquinas onde não houve nenhuma falha, já que o conjunto de dados fornecido pela Siemens só possui eventos relacionados à falhas. Existem códigos que, embora estejam relacionados à uma visita técnica, são meramente informativos e ocorrem também durante a operação normal das turbinas. Para separar os códigos indicativos de falha daqueles que ocorrem normalmente será utilizado o índice TFIDF, bastante conhecido em processamento de linguagem natural, onde auxilia a classificação de documentos de acordo com o número de palavras raras encontradas em cada um.
A estratégia para a resolução do problema é a seguinte:

- Cálculo do índice TFIDF dos códigos
- Agrupamento de códigos com valores altos de índice TFIDF em sequências
- Treinamento do modelo preditivo de duração das visitas com os códigos obtidos acima e outras variáveis contidas nos dados
- Avaliação do modelo

O resultado esperado é um modelo preditivo de duração das visitas e as sequências de códigos representativos de falha.

Métricas

O objetivo do modelo preditivo é identificar visitas com duração menor que 10 minutos e visitas cuja duração é superior à 20 minutos. As métricas escolhidas para o modelo preditivo de tempo de manutenção são *precision e recall*. *Precision* indica em um número de previsões a quantidade de acertos que foram feitos de acordo com o rótulo duração da visita, curta ou longa. *Recall* indica no conjunto de dados a

quantidade de visitas curtas e longas que foram corretamente identificadas. Segue abaixo a definição matemática de cada métrica, que pode assumir valores de 0 até 1:

TP: True positives- prever um evento que realmente ocorreu na realidade
FN: False negatives- prever inexistência de um evento, quando na realidade ele ocorreu
FP: False positives- prever um evento, quando na realidade ele não ocorreu

$$Precision = \frac{TP}{TP + FP} \quad , Recall = \frac{TP}{TP + FN}$$

Neste caso dois modelos serão definidos: um para identificar visitas com duração inferior a 10 minutos e outro para identificar aquelas superiores a 20 minutos. *Precision* foi escolhida como métrica porque é suficiente para avaliar se as previsões têm um número alto de acertos, prevendo visitas curtas ou longas quando elas realmente se confirmaram. Já o *Recall* foi escolhido porque é necessário identificar se cada uma das classes é capaz de ser identificada corretamente pelo modelo.Por exemplo, no modelo que identifica visitas inferiores a 10 minutos o *Recall* é uma indicação das visitas que são inferiores a 10 minutos e foram corretamente identificadas. Desta forma, o objetivo é obter um modelo com valores de *precision e recall* próximos de 1.

II. Análise

Explorando os Dados

Arquivos de dados:

- wind.xlsx (13.3 MB): arquivo principal de dados
- codes.xlsx (21.5 kB): arquivo auxiliar com códigos
- sites.xlsx (8.3 kB): arquivo auxiliar com tamanho dos parques

A figura1 mostra o conjunto de dados principal, abaixo são descritos os dados contidos neste conjunto.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Park_Name	FactorA	FactorB	FactorC	FactorD	StationID	VisitType	VisitId	ManualStop	VisitStartTime	VisitDurMinutes	Code	ManualStop	TimeOn	TimeOff
2	Park024	4	GGS	BBC	B	8894	PL	178	yes	25/08/2016 13:53	4	3173	FALSO	24/08/2016 08:33	24/08/2016 11:49
3	Park024	4	GGS	BBC	B	8894	PL	178	yes	25/08/2016 13:53	4	63027	FALSO	24/08/2016 08:33	24/08/2016 11:50
4	Park024	4	GGS	BBC	B	8894	PL	178	yes	25/08/2016 13:53	4	13	FALSO	24/08/2016 11:33	
5	Park024	4	GGS	BBC	B	8894	PL	178	yes	25/08/2016 13:53	4	14	FALSO	24/08/2016 11:39	
6	Park024	4	GGS	BBC	B	8894	PL	178	yes	25/08/2016 13:53	4	13	FALSO	24/08/2016 11:39	
7	Park024	4	GGS	BBC	B	8894	PL	178	yes	25/08/2016 13:53	4	14	FALSO	24/08/2016 11:44	
8	Park024	4	GGS	BBC	B	8894	PL	178	yes	25/08/2016 13:53	4	13	FALSO	24/08/2016 11:49	
9	Park024	4	GGS	BBC	B	8894	PL	178	yes	25/08/2016 13:53	4	1001	VERDADEIRO	24/08/2016 11:49	24/08/2016 11:52
10	Park024	4	GGS	BBC	B	8894	PL	178	yes	25/08/2016 13:53	4	63027	FALSO	24/08/2016 11:50	24/08/2016 11:52
11	Park024	4	GGS	BBC	B	8894	PL	178	yes	25/08/2016 13:53	4	2	FALSO	24/08/2016 11:50	
12	Park024	4	GGS	BBC	B	8894	PL	178	yes	25/08/2016 13:53	4	7	FALSO	24/08/2016 11:50	
13	Park024	4	GGS	BBC	B	8894	PL	178	yes	25/08/2016 13:53	4	18	FALSO	24/08/2016 11:50	

Figura1- Conjunto principal de dados

Park_Name – designação do parque eólico
FactorA, FactorB, FactorC, FactorD – características mascaradas dos parques
StationID – identificador único da turbina
VisitType – indicação dos técnicos referente ao tipo de visita
VisitID – identificador de cada visita
Manual Stop during Visit – indica se houve código de parada manual
VisitStartTime – data e horário de início da visita
VisitDurMinutes – duração das visitas em minutos
Code – número que indica o evento,falha ou alarme no log do histórico da turbina
ManualStop – indica se o código foi gerado por uma pessoa(localmente ou remotamente por software)
TimeOff – quando o código desapareceu (formato data/horário)
TimeOn – quando o código apareceu (formato data/horário)

Um conjunto auxiliar de dados possui a descrição para os códigos mostrados na coluna 'Code' do conjunto principal. Este segundo conjunto é mostrado na figura2.

	A	B	C	D
1	Code	EventWarningStop	IsManualStop?	StopUrgency
2	2	Event	FALSO	0
3	7	Event	FALSO	0
4	8	Event	FALSO	0
5	9	Event	FALSO	0
6	13	Event	FALSO	0
7	14	Event	FALSO	0
8	18	Event	FALSO	0
9	59	Event	FALSO	0
10	1001	Stop	VERDADEIRO	4
11	1002	Stop	VERDADEIRO	5
12	1003	Stop	VERDADEIRO	1
13	1004	Stop	FALSO	5
14	1005	Stop	FALSO	5
15	1007	Stop	VERDADEIRO	5
16	1008	Stop	VERDADEIRO	5
17	1010	Stop	FALSO	1

Figura2- Conjunto auxiliar de dados

Code - número do código
EventWarningStop - evento,parada ou alarme
IsManualStop - se foi uma parada gerada por intervenção humana
StopUrgency - urgência(0= não é uma parada;1=menor urgência,até 6= maior urgência)

A figura3 mostra o segundo conjunto auxiliar de dados, com o número de aerogeradores por parque.

	A	B
1	Park_Name	#Assets
2	Park002	76
3	Park032	22
4	Park025	32
5	Park028	15
6	Park033	11
7	Park027	15
8	Park030	10
9	Park007	26
10	Park022	21

Figura3- Quantidade de aerogeradores por parque

Como mostrado na figura4 , o conjunto principal de dados possui 171929 linhas e 15 colunas.Os tipos de dados existentes são: string,int, bool e datetime64. Estes tipos estão de acordo com o esperado para cada variável. A única variável que possui valores nulos é 'TimeOff'(29988 nulos), nas outras não existem valores nulos ou inexistentes.

```
RangeIndex: 171929 entries, 0 to 171928
Data columns (total 15 columns):
Park_Name                 171929 non-null object
FactorA                   171929 non-null int64
FactorB                   171929 non-null object
FactorC                   171929 non-null object
FactorD                   171929 non-null object
StationID                 171929 non-null int64
VisitType                 171929 non-null object
VisitId                   171929 non-null int64
ManualStop during Visit   171929 non-null object
VisitStartTime            171929 non-null datetime64[ns]
VisitDurMinutes           171929 non-null int64
Code                      171929 non-null int64
ManualStop                171929 non-null bool
TimeOn                    171929 non-null datetime64[ns]
TimeOff                   141941 non-null datetime64[ns]
dtypes: bool(1), datetime64[ns](3), int64(5), object(6)
memory usage: 18.5+ MB
```

Figura4- Estatísticas básicas para o conjunto principal de dados

O primeiro conjunto auxiliar possui 642 linhas e 4 colunas, os tipos de dados são string e float e estão também de acordo com o esperado para as variáveis. Existe apenas um valor nulo para as categorias 'Code' ,'EventWarningStop' e 'IsManualStop'.O segundo conjunto auxiliar de dados possui 37 linhas e nenhum valor nulo.No conjunto de dados principal alguns códigos precisam ser transformados em dummys, para utilização no modelo preditivo.É necessário também unir os conjuntos de dados auxiliares ao conjunto principal. Estes dados também precisam ser agrupados por visita.Como informado pela Siemens, nem todas as visitas são reais, elas podem ter sido indicadas ao software de monitoramento por causa de algum ruído elétrico nos cabos, indicando que uma equipe de manutenção estava em um aerogerador,quando não havia ninguém lá.

Aquelas que ocorrem antes das 6h e após 20h são falsas,já que as equipes de manutenção não trabalham nestes horários. Visitas sinalizadas em muitos parques diferentes quase ao mesmo tempo também são falsas, já que existem poucas equipes.

Visualização Exploratória

Após as operações de união de conjuntos de dados,limpeza de visitas falsas e agrupamento por visita foi produzido um gráfico para a análise exploratória dos dados, mostrado na figura5. Na diagonal principal deste gráfico é mostrado um histograma de frequência de valores e fora da diagonal são mostradas as correlações entre variáveis.Descrição das variáveis e principais conclusões:

- Número do Parque -Os parques de numeração 1 até 5 têm a maior frequência de visitas. Os parques de numeração 6 até 37 têm frequência menor.
- Hora da Visita - As visitas têm uma frequência que cresce das 6h até 9h da manhã, quando começam a diminuir até as 20h.
- Mês da Visita - Para o período de um ano o mês de Março teve o maior número de visitas e Dezembro o menor.
- Duração da Visita - A maior parte das visitas têm duração inferior a 10 minutos.
- Percentual Visitado em um Dia - Esta variável mostra o percentual de aerogeradores de um parque que foi visitado em um dia. Na maior parte dos dias as visitas ocorreram em até 20% dos aerogeradores dos parques visitados naquele dia.

Foi encontrada uma leve correlação negativa entre a variável ‘Percentual Visitado em um Dia’ e ‘Duração da Visita’, indicando que parques pouco visitados em um dia têm visitas mais longas.Quando o parque é muito visitado estas visitas são curtas.As variáveis ‘Percentual Visitado em um Dia’ , ‘Duração da Visita’ têm distribuição chi-quadrada e ‘Hora da Visita’ normalmente distribuída,inclinada para a direita.

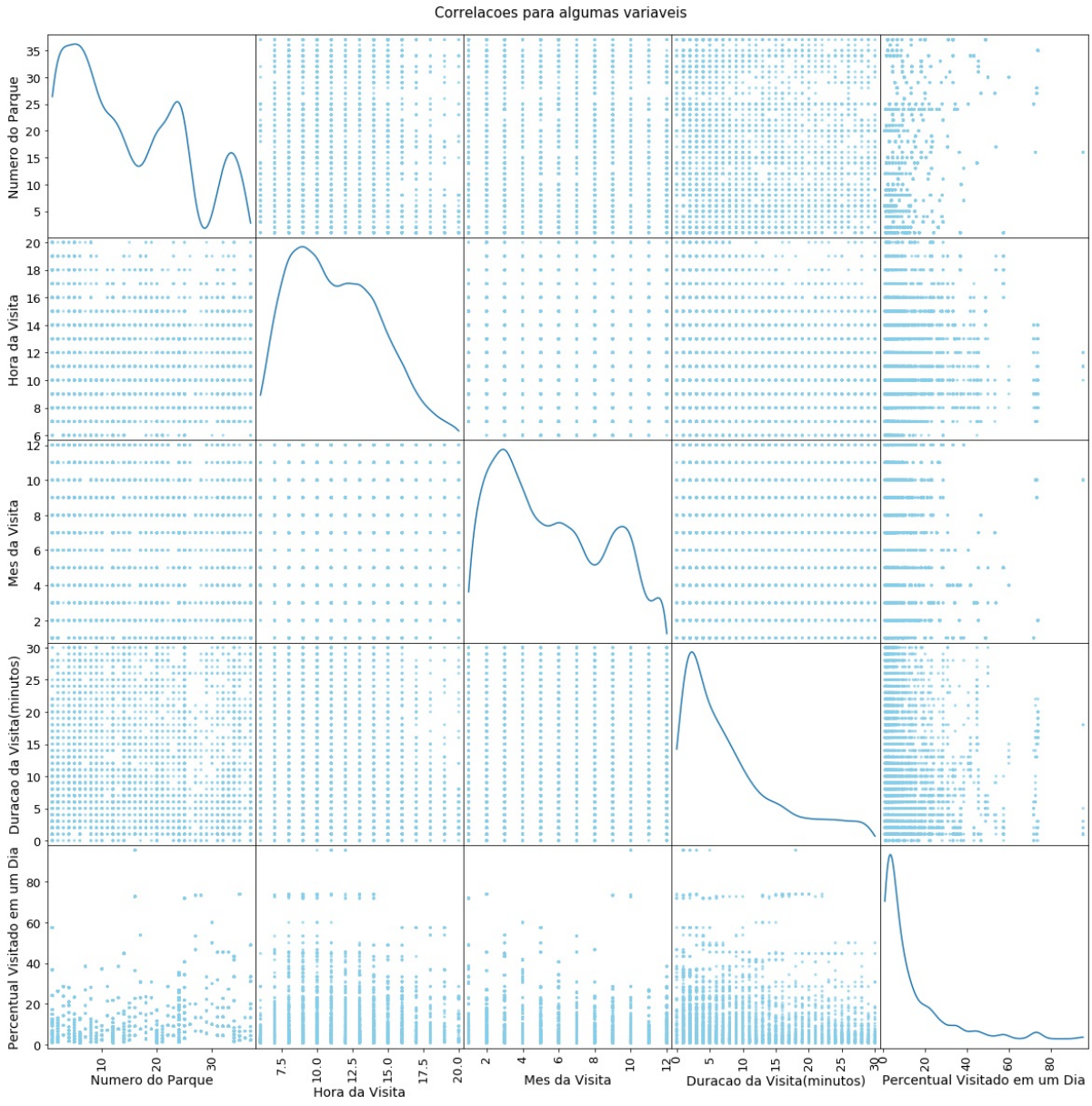

Figura5- Visualização Exploratória

Algoritmos e Técnicas

TFIDF

A técnica escolhida é a mesma utilizada por Lijie (2011) ,para analisar os códigos de falha e criar um modelo preditivo: o cálculo do índice *term frequency and inverse document frequency* (TFIDF) para todos os códigos do conjunto de dados e utilização destes códigos no modelo. Este índice é utilizado para a classificação de textos: a idéia é que se uma palavra ocorre em mesmo número em vários documentos sua força de identificação é baixa com relação à uma palavra que ocorre muito em um documento alvo e pouco nos outros documentos. Da mesma forma, se um código ocorre em muitas visitas sua força de identificação é baixa e

certamente este código aparece durante a operação normal dos aerogeradores, não sendo relacionado com a falha. Mas se o código tem frequência alta em eventos alvo e baixa no restante ele está relacionado com a falha no conjunto alvo de eventos. Cálculo do índice TFIDF:

$$TFIDF = \frac{ni}{N} log \frac{D}{1+di}$$

ni - número de eventos de falha no conjunto alvo com código i
N - número de eventos de falha no conjunto alvo
D - número de eventos de falha no conjunto que não é alvo
di - número de eventos de falha com código i no conjunto que não é alvo

Neste caso os eventos serão divididos primeiramente entre aqueles que ocorrem para visitas inferiores a 10 minutos(conjunto alvo), considerando o restante dos eventos como conjunto que não é alvo. A segunda etapa considera os eventos que ocorrem para visitas superiores a 20 minutos(conjunto alvo) e o restante como não sendo alvo. Após este cálculo, os códigos com altos índices TFIDF serão utilizados em um algoritmo XGBoost, descrito abaixo.

XGBOOST

XGBoost é um algoritmo de aprendizado supervisionada de máquina baseado em árvores de decisão com aumento de gradiente(*gradient boosting*), é altamente flexível e versátil, além de escalável e rápido. XGBoost pode trabalhar tanto com regressões como com classificações, ganhou popularidade nos anos recentes devido à compatibilidade com soluções distribuídas.Resumidamente, XGBoost é uma variação de *boosting*, um método de algoritmo combinado que tenta ajustar dados usando modelos 'fracos', tipicamente árvores de decisão. A idéia é a de que um modelo 'fraco',com baixa precisão e revocação,pode ser melhorado para se tornar um modelo 'forte' que tem performance melhor.A cada iteração,cada modelo 'fraco' tenta reduzir o erro do conjunto de modelos, produzindo um modelo combinado que seja melhor que o anterior. XGBoost é um modelo do tipo *gradient boosting*, onde modelos fracos são generalizados otimizando uma função de erro, que usa o gradiente descendente para minimizar o erro entre o valor previsto e real.

Existem muitas vantagens do XGBoost em relação a outros métodos de classificação:

- Trabalha com tamanho elevado de dados: o algoritmo tem muitas funções que facilitam o trabalho com muitos dados, que simplesmente nao cabem na memória, utilizados em computação distribuída. Ele também tem

gerenciamento automático de dados faltantes e permite a continuidade do treinamento.

- Regularização embutida: O algoritmo tem várias opções para controle de regularização e ajuste contra overfitting, incluindo *gamma*,regularização L1 e L2, máxima profundidade da árvore,mínima soma de pesos de todas as observações de uma ramificação,etc.
- Otimização para velocidade e performance: o XGBoost tem opções para reduzir o tempo de treinamento, mantendo a acurácia do modelo usando paralelização com CPU de múltiplos processadores e otimização de cache.

Será necessária a criação de *dummies* dos códigos com maior índice TFIDF para que estas variáveis possam ser utilizadas pelo algoritmo preditivo.Desta forma, com os valores rótulo de tempo de duração da visita serão treinados dois algoritmos: um para prever visitas com duração inferior a 10 minutos e outro para prever visitas com duração superior a 20 minutos.

Benchmark

O Benchmark para visitas com duração inferior a 10 minutos considera a previsão de que todas pertencem à esta classe, já que a média de duração das visitas de todo o conjunto de dados é 9.6 minutos. Já o Benchmark para visitas superiores a 20 minutos considera como previsão uma escolha aleatória, já que apenas 17% das visitas no conjunto de dados pertencem à esta classe.Os valores para a classe1 sofrem uma pequena variação, já que são calculados de forma aleatória.

	Precision	Recall
class0	0.60	1.00
class1	0.84	0.51

Figura6- Benchmarks-class 0: (<10 min) , class 1: (>20 min)

III. Metodologia

Pré-processamento

Como mostrado anteriormente na seção Explorando Dados, a única limpeza necessária nos dados é a retirada de visitas falsas, que foi feita antes de gerar os gráficos de visualização exploratória.Para a utilização do algoritmo preditivo, a coluna ‘Code’, com os códigos de evento, foi transformada em variáveis *dummy*

para cada código.A variável ‘TimeOff’ tem 29988 valores nulos, mas ela não será utilizada.

Implementação

Como descrito na seção Algoritmos e Técnicas, é necessário primeiramente o cálculo do índice TFIDF para os eventos com duração inferior a 10 min. Os códigos com seu respectivo índice são mostrados na figura7, onde os códigos à esquerda são aqueles que ocorrem muito no conjunto de dados alvo e pouco no restante, tendo grande poder de distinção. Já os códigos à direita são aqueles que ocorrem em igual proporção nos dois conjuntos,tendo pouco poder de distinção. O maior índice obtido tem valor 0.45 ,que é considerado baixo para que seja possível distinguir entre grupos de dados. Para os eventos com duração superior a 20 minutos os códigos com índices calculados são mostrados na figura8. Neste caso o maior índice obtido tem valor 0.6, um pouco mais significativo que no caso anterior.

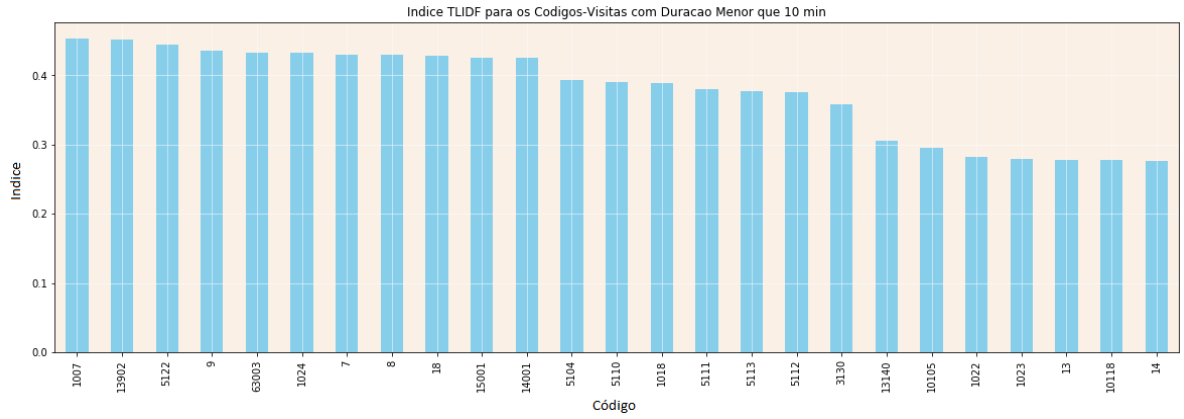

Figura7- Índice dos códigos para visitas curtas

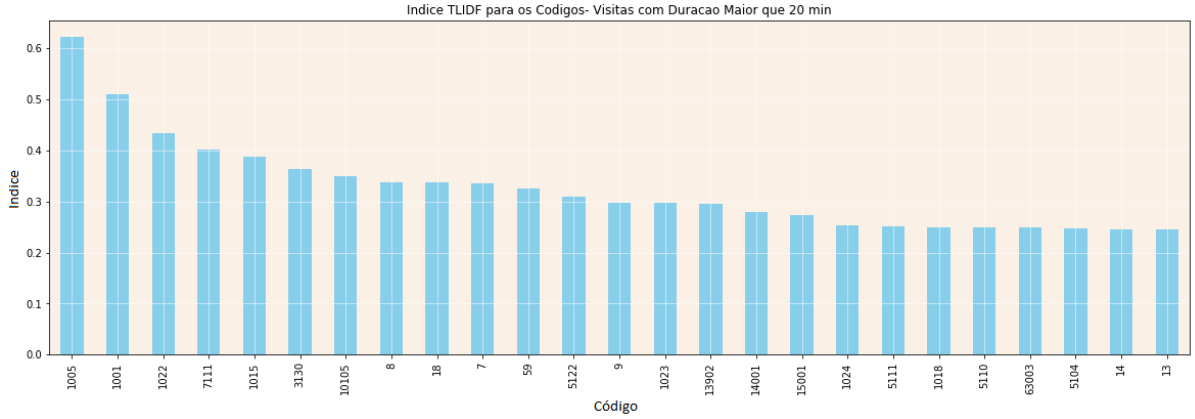

Figura8- Índice dos códigos para visitas longas

Com o objetivo de identificar as sequências que mais ocorrem em visitas curtas e longas para estes códigos com maior índice TFIDF, foram contadas as ocorrências de códigos de interesse que aparecem juntos para um período de uma hora.Desta forma, foi criada uma nova dimensão para o gráfico TFIDF, normalizada pela maior contagem obtida. As figuras 9 e 10 mostram estes gráficos, onde a contagem foi feita apenas para os códigos com maiores índices TFIDF. Para as visitas curtas,observa-se na figura9 que os códigos 7,8 e 18 formam uma sequência, já que têm o mesmo índice temporal de contagem. Já para as visitas longas, a figura10 mostra que os códigos 8,18 e 17 formam uma sequência.

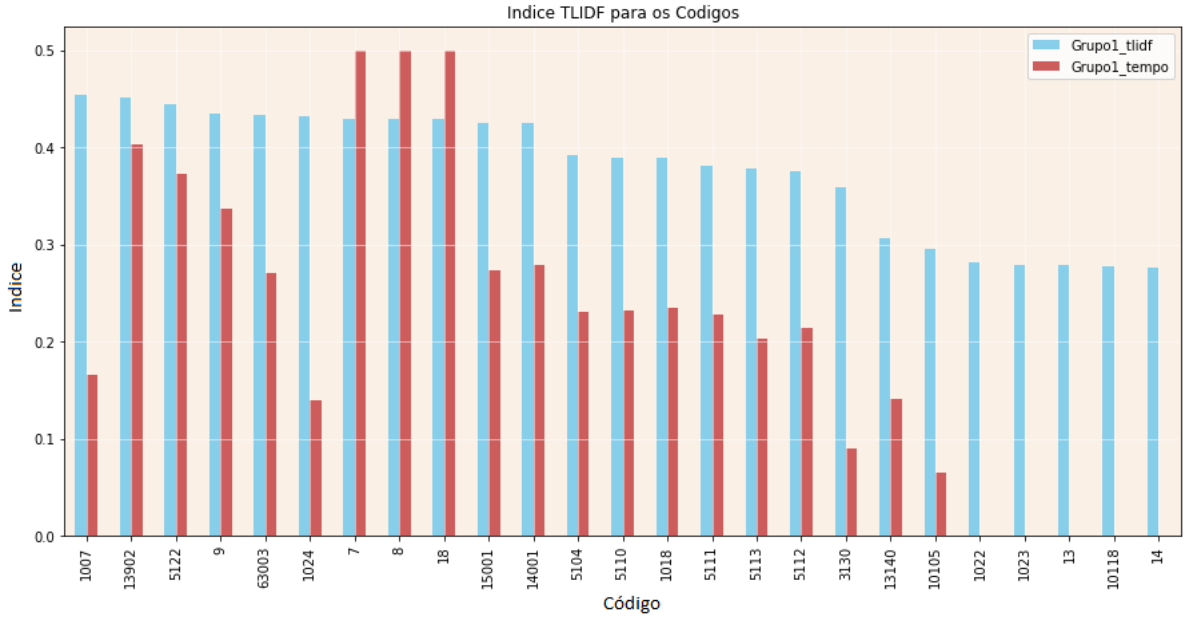

Figura9- Índice dos códigos para visitas curtas

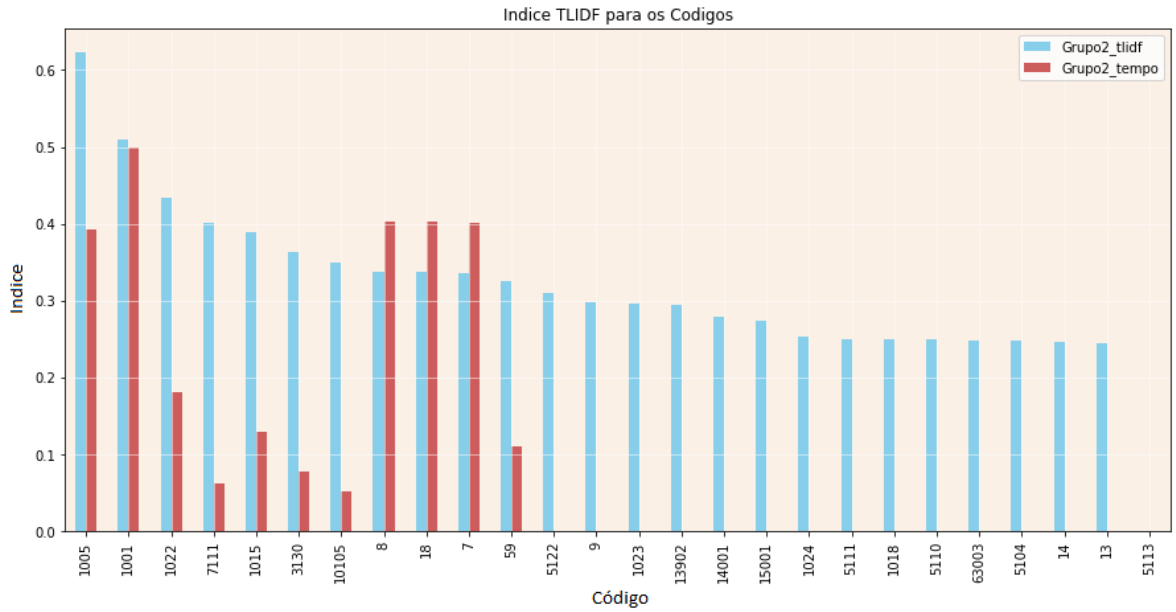

Figura10- Índice dos códigos para visitas longas

O cálculo de índices TFIDF também foi realizado com a divisão do conjunto de dados entre códigos que aconteceram antes das visitas e aqueles que aconteceram durante e depois.Na primeira etapa, os códigos anteriores à visita representam o conjunto alvo,códigos que ocorrem durante ou depois não são alvo. Na segunda etapa, ,códigos que ocorrem durante ou depois são alvo, enquanto aqueles anteriores à visita representam o conjunto não alvo. Esta análise é importante, pois códigos semelhantes que aconteceram antes das visitas demonstram causa semelhante também.Já uma semelhança entre os códigos que ocorreram durante e depois indicam que foi realizado serviço similar.

Com relação ao código para cálculo do índice TFIDF não houve nenhuma dificuldade, mas com relação à parte de contagem foi um pouco mais complicado já que era necessário criar também uma lista com as sequências encontradas. Foi criado um dicionário, usando como chave o número do código, sendo que o valor dos elementos é a contagem de quantas vezes o código de interesse apareceu seguido de outro.Também foi necessário gerar uma lista com as sequências obtidas, de modo que pudessem ser contadas depois. Este código precisa identificar sequências de qualquer tamanho que acontecem em períodos inferiores a uma hora. Foi necessário incluir várias condições ‘if’ no código e testar se os valores obtidos para as contagens estavam corretos.

O modelo XGBoost foi treinado com os códigos como variáveis de entrada e outras variáveis de interesse. Os parâmetros do modelo são mostrados abaixo:

- booster='gbtree'
- class_weight='balanced'
- objective='multi:softmax'
- tree_method = ‘auto’

Refinamento

O ajuste de hiper-parâmetros foi realizado no XGBoost com ajuda do GridSearchCV, do Sklearn. O espaço de procura está listado abaixo:

- max_depth: Máxima profundidade da árvore, já que valores maiores produzem aprendizado de relações mais específicas.Valores: [3,6,9]
- colsample_bytree: Controla a quantidade de variáveis amostradas por cada árvore. Valores: [0.7,0.8,0.9]
- subsample: Proporção de subamostragem, pode prevenir overfitting. Valores: [0.5,0.6,0.7]
- n_estimators: Número de árvores criadas. Valores: [2,4,8]
- learning_rate: Controla a velocidade de convergência. Valores: [0.1,0.2,0.3]
- min_child_weight: mínimo peso de observações em um nó derivado. Valores: [3,4,5]
- max_delta_step: controla o passo de atualização. Valores: [10,15,20]

Resultados antes do gridsearch: Modelo1-F1: 0.62 , Modelo2- F1: 0.78

Resultados após o grid search:

Modelo1:

```
XGBClassifier(base_score=0.5, booster='gbtree', class_weight='balanced',
    colsample_bylevel=1, colsample_bytree=0.9, gamma=0,
    learning_rate=0.3, max_delta_step=15, max_depth=9,
    min_child_weight=3, missing=None, n_estimators=4, n_jobs=1,
    nthread=-1, num_class=2, objective='multi:softmax', random_state=42,
    reg_alpha=0, reg_lambda=0.9, scale_pos_weight=1, seed=None,
    silent=True, subsample=0.7)
```

F1: 0.64

Modelo2:

```
XGBClassifier(base_score=0.5, booster='gbtree', class_weight='balanced',
    colsample_bylevel=1, colsample_bytree=0.9, gamma=0,
    learning_rate=0.3, max_delta_step=15, max_depth=9,
    min_child_weight=3, missing=None, n_estimators=4, n_jobs=1,
    nthread=-1, num_class=2, objective='multi:softmax', random_state=42,
```

reg_alpha=0, reg_lambda=0.9, scale_pos_weight=1, seed=None,
silent=True, subsample=0.7)
F1: 0.78

Houve apenas melhora no escore F1 do Modelo1, que passou de 0.62 para 0.64, após o grisearch. O escore F1 do Modelo2 se manteve em 0.78. Este escore F1 considera a média ponderada para duas classes em cada modelo, mas em cada modelo só existe interesse em prever uma classe apenas.

IV. Resultados

Avaliação dos Modelos

O modelo final para prever visitas com duração inferior a 10 minutos contém as seguintes variáveis: “Eventos Antes das Visitas”, “Mês da Visita”, ”Nome do Parque”,”Percentual_Visitado_em_um_dia”,”1007-1024-9-63003”(códigos),”13902-5 122”(códigos). Este modelo obteve um escore F1 de 0.74 para a classe de visitas abaixo de 10 minutos. Já o modelo que prevê visitas superiores a 20 minutos possui as variáveis: “Eventos Antes das Visitas”, “Mês da Visita”, ”Nome do Parque”,”Percentual Visitado em um dia”,”1005-1001”(códigos).O escore F1 obtido para este modelo foi de 0.91 para a classe de visitas acima de 20 minutos. As variáveis de códigos foram obtidas a partir da análise TFIDF, que indicou os códigos mais significativos de cada tipo de visita. Para analisar a robustez dos modelos,foi feito um novo treinamento usando 50% do tamanho do conjunto de dados utilizado antes.O escore F1 obtido para os modelos foi de 0.72 e 0.91 respectivamente. Isto mostra que os modelos obtidos são bastante robustos, pois têm baixa sensibilidade à mudanças no tamanho dos dados de treinamento.

Justificativa

A Tabela1 mostra a comparação dos resultados obtidos pelos modelos com o benchmark. Para as visitas com duração inferior a 10 min houve uma melhora na precisão de 0.60 para 0.67. O modelo proposto para esta classificação obteve um Recall alto, de 0.82,embora distante do valor 1 obtido no benchmark.
Já o segundo modelo, proposto para obter as classificações de visitas acima de 20 minutos, obteve uma melhora pouco significativa na precisão mas bastante alta no recall, que passou de 0.51 para 0.99. Desta forma, os modelos propostos para o problema de previsão de duração das visitas se mostram no geral melhores que o benchmark.

Classe	Benchmark		Modelos Propostos	
	Precision	Recall	Precision	Recall
Duração < 10 min	0.60	1.00	0.67	0.82
Duração > 20 min	0.84	0.51	0.85	0.99

Tabela1- Comparação dos resultados com o benchmark

V. Conclusão

Visualizações Finais

Através da metodologia descrita no item 'Implementação', foram obtidas as sequências de códigos de interesse que mais ocorrem antes das visitas. Estas sequências são mostradas na Figura11, onde o tamanho dos retângulos é proporcional ao número de visitas em que estas sequências apareceram. Visitas que foram precedidas por estas sequências indicam a mesma causa. Já a figura12 mostra as sequências que ocorrem durante e depois das visitas. Visitas que tiveram estas sequências durante ou após a manutenção indicam que o mesmo tipo de serviço foi realizado.

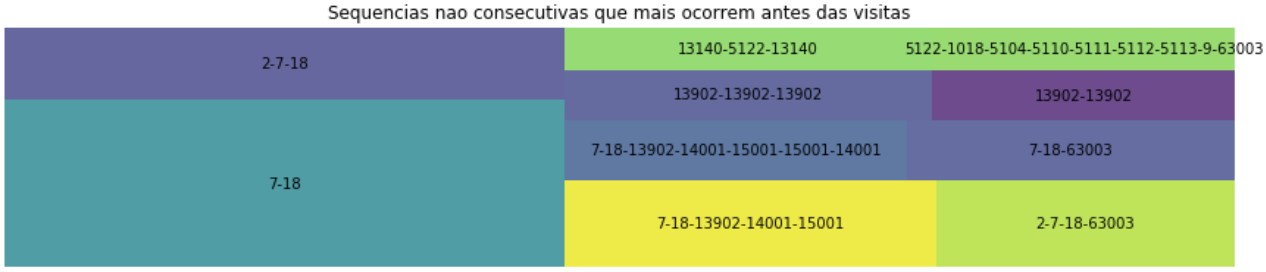

Figura11- Sequências anteriores às visitas

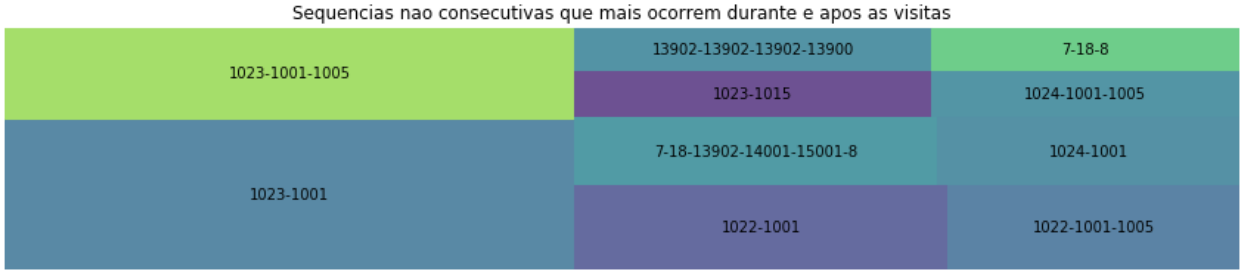

Figura12- Sequências durante a após as visitas

As figuras 13 e 14 mostram a importância de cada variável utilizada nos modelos.O número de códigos que ocorrem antes de uma visita é a variável que possui maior importância nos dois modelos, seguida do mês da visita. O nome do parque e percentual do parque visitado em um único dia são indicadores com grande importância também, seguidos dos códigos que caracterizam visitas longas ou curtas.Conclui-se portanto que a duração das visitas está muito mais relacionada ao número de códigos anteriores e mês de ocorrência do que à ocorrência de códigos de interesse.Esta análise é de grande importância para a gerência de manutenção de um parque eólico, pois indica quais variáveis influenciam mais a duração das visitas.

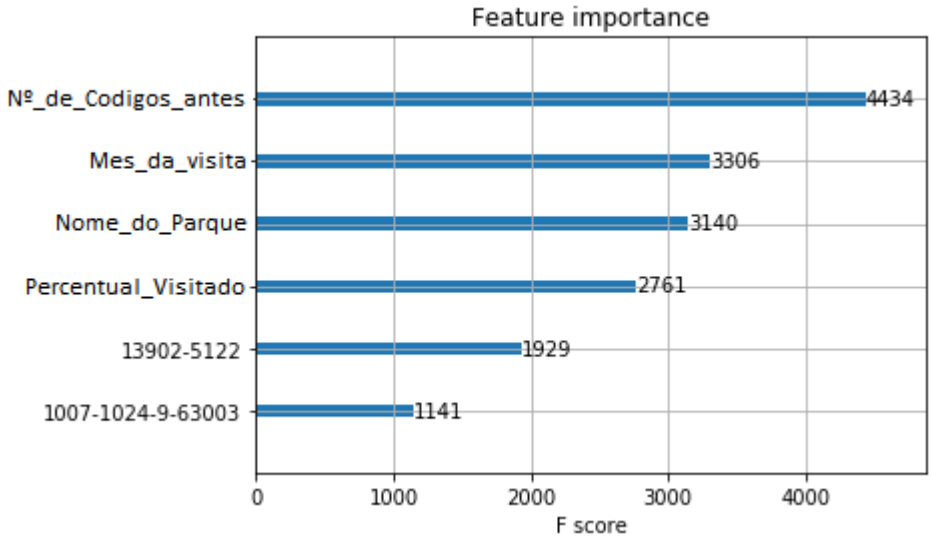

Figura13- Importância das variáveis no modelo de visitas curtas

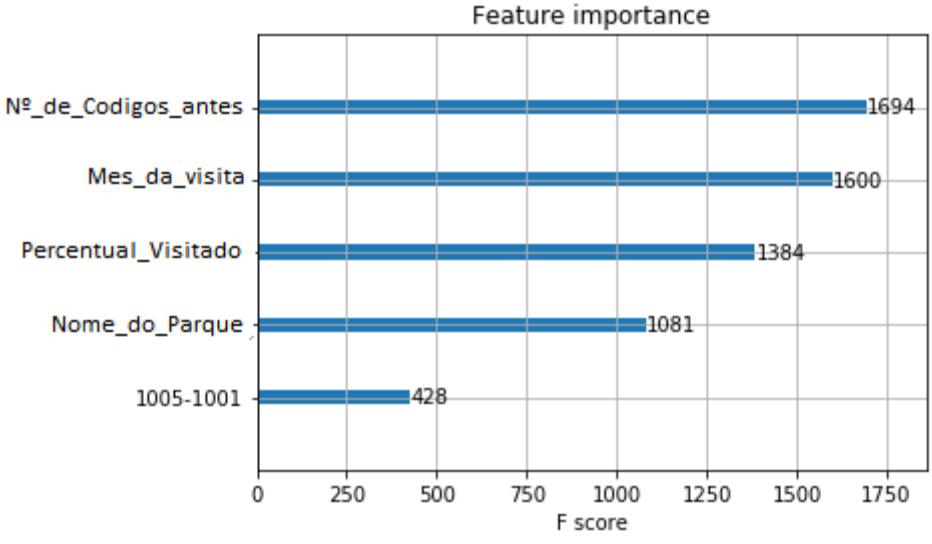

Figura14- Importância das variáveis no modelo de visitas longas

Reflexão

Este projeto analisa vários dados de manutenção de um parque eólico, na tentativa de auxiliar a gerência de manutenção a identificar padrões que possam indicar a causa das visitas e sua duração. A quantidade de códigos analisados é enorme, desta forma foi necessária a utlização do índice TFIDF para diferenciar códigos de operação normal de códigos realmente relacionados com as visitas.A solução mostrada aqui incluiu os seguintes passos:

- Estabelecimento das estatísticas descritivas: Nesta etapa foram mostrados os dados básicos do conjunto a ser analisado, como formato dos dados, correlações entre variáveis e distribuições de frequência. Foi necessária apenas a limpeza de visitas falsas.
- Criação de novas variáveis: Aqui foi calculado o índice TFIDF, foram mostrados os códigos que obtiveram os maiores índices.
- Treinamento e teste do Modelo: O treinamento foi realizado com dois modelos XGBoost utilizando as variáveis criadas, com o objetivo de prever visitas curtas e longas. O escore F1 para as classes de interesse foi de 0.74 e 0.91.
- Refinamento: A utilização do gridsearch não provocou melhora significativa no desempenho do modelo
- Sequências de maior ocorrência: Foram mostradas as sequências obtidas através do índice TFIDF que explicam a ocorrência das visitas.

Foram encontrados muitos desafios durante este projeto, listados abaixo:

- A análise PCA, inicialmente prevista para ser utilizada no grande número de códigos não apresentou bons resultados e foi abandonada.
- Análises similares ao PCA, como MCA ou MFA também não resolveram o problema
- A previsão de tempo das visitas através de uma regressão numérica não mostrou bons resultados e não foi utilizada.
- A implementação do código para contagem das sequências de interesse exigiu vários testes até que estivesse correto.
- A criação de novas variáveis foi bastante demorada, devido ao tamanho do conjunto de dados e diferentes formas de agrupamento e ordenação dos dados que foram necessárias para esta tarefa.
- Compreender e replicar o artigo de Lijie(2011) foi uma tarefa bastante desafiadora.

Aperfeiçoamento

Os valores obtidos de Precision e Recall para o Modelo1 ainda podem ser aperfeiçoados com a criação de novas variáveis, já que o escore F1 obtido foi de 0,74. Uma das idéias é utilizar a variável “Factor”, que mostra as características mascaradas de cada parque. Poderia ser testada a técnica ‘sequence mining’ ou ‘sequential pattern mining’, para analisar se as sequências resultantes são as mesmas obtidas aqui com TFIDF. O único artigo encontrado sobre manutenção preditiva de parques eólicos não cita esta técnica e desconheço sua teoria, mas seria interessante testá-la. Caso as sequências fossem diferentes elas poderiam também ser utilizadas como variáveis para o modelo preditivo. O resultado potencial desses aperfeiçoamentos seria uma melhor representação das sequências de falha apresentadas e maior escore F1 para o modelo de visitas curtas.

Referências

UCF-DATA MINING PROGRAM. **Siemens Wind Analytics Contest**. Disponível em: <https://sciences.ucf.edu/statistics/dms/siemens-2017-wind-analytics-contest/> Acesso em: 05 de novembro de 2018

LIJIE,Y. **Wind Turbine Data Analytics for Drive-Train Failure Early Detection and Diagnostics**.ASME 2011 Turbo Expo- Turbine Technical Conference and Exposition,Vancouver, 2011.