Testes ao Modelo:

Os testes de modelos são uma parte fundamental da construção de modelos. Quando feito correctamente, os testes asseguram que o seu modelo é estável e não está sobreajustado. Os três métodos mais conhecidos de teste de modelos são o teste aleatório de treino dividido, a validação cruzada em K, e deixar um de fora da validação cruzada.

<https://builtin.com/data-science/machine-learning-models-python>

Podemos também especificar um parâmetro chamado random\_state. Este é um parâmetro de embaralhamento que controla a forma como os dados são divididos aleatoriamente. Se lhe dermos um valor inteiro, garantimos que reproduzimos a mesma divisão em cada execução:

Ter uma boa compreensão das ferramentas disponíveis para a construção de modelos robustos de aprendizagem de máquinas é uma competência que todos os cientistas de dados devem ter. Ser capaz de preparar dados para formação e testes, selecionar características, e afinar parâmetros de modelos é necessário para construir modelos estáveis cujas previsões são fiáveis. Além disso, ter estas ferramentas no seu bolso de trás pode poupar horas de trabalho significativas, uma vez que estes métodos automatizam o que de outra forma seria feito manualmente. Estas técnicas, quando usadas corretamente, permitem reduzir o risco de deterioração do modelo que pode custar às empresas milhões de dólares no futuro.

Compreender como estabelecer adequadamente os testes de modelos é também vital para garantir que não se sobrepõe aos seus modelos. Por exemplo, se não dividir corretamente os seus dados para treino e testes, os testes dos seus modelos podem dar-lhe uma falsa sensação de precisão dos modelos, o que pode ser muito caro para uma empresa. No caso de rotatividade, se o seu modelo for enganosamente preciso, pode visar incorretamente os clientes com anúncios e descontos que não são realmente susceptíveis de rotatividade. Isto pode resultar em milhões perdidos em dólares de anúncios.

Para reduzir o risco de sobreajustamento e sobreavaliação do desempenho do modelo, é crucial ter um conjunto de teste de retenção (como o que geramos a partir da divisão aleatória do teste e treino), no qual realizamos um único teste após a afinação do modelo e selecionamos as características nos dados de treino. Além disso, os métodos de validação cruzada dão-nos uma boa compreensão da estabilidade dos nossos modelos, tanto em termos de afinação do modelo como de seleção de características. A validação cruzada permite-nos ver como o desempenho varia entre vários testes. Idealmente, gostaríamos de selecionar as características e os parâmetros do modelo que dão o melhor desempenho com a menor variação no desempenho do modelo. Baixa variância no desempenho do modelo, significa que as previsões do modelo são mais fiáveis e de menor risco.

A seleção das características é também muito importante, uma vez que pode ajudar a filtrar um número potencialmente grande de entradas. Não importa quão preciso seja um modelo, será muito difícil interpretar como os valores para 1.000 características, por exemplo, determinam se um cliente se agitará ou não. A seleção de características, juntamente com a perícia do domínio, pode ajudar os cientistas de dados a selecionar e interpretar as características mais importantes para a previsão de um resultado.

A afinação de hiperparâmetros é também um passo essencial necessário para se obter um desempenho óptimo do modelo. Enquanto os valores dos parâmetros padrão fornecidos pelos pacotes de aprendizagem da máquina Scikit-learn fornecem tipicamente um desempenho decente, a afinação do modelo é necessária para se conseguir modelos muito precisos. Por exemplo, com a rotatividade, quanto mais precisamente se pode visar clientes susceptíveis de deixar uma empresa, menos dinheiro é perdido para clientes incorretamente visados e mais aumenta o valor vitalício dos clientes em alto risco de sair.

Ter uma forte familiaridade com as ferramentas disponíveis para a criação de testes de modelos, seleção de características e afinação de modelos é um conjunto de competências inestimáveis para os cientistas de dados em qualquer indústria. Ter este conhecimento pode ajudar os cientistas de dados a construir modelos robustos e fiáveis que podem acrescentar um valor significativo a uma empresa, resultando em poupanças de recursos em termos de dinheiro e mão-de-obra, bem como aumentar os lucros da empresa. O código deste posto está disponível em GitHub.

Aqui explica: The most comprehensive guide to automated feature selection methods in Python

<https://www.datagraphi.com/blog/post/2019/9/23/feature-selection-with-sklearn-in-python>

<https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74>

**Random Search Training**

linspace 🡪 escolhe 10 numeros entre 200 e 2000

from sklearn.model\_selection import RandomizedSearchCV# Number of trees in random forest  
n\_estimators = [int(x) for x in np.linspace(start = 200, stop = 2000, num = 10)]  
# Number of features to consider at every split  
max\_features = ['auto', 'sqrt']  
# Maximum number of levels in tree  
max\_depth = [int(x) for x in np.linspace(10, 110, num = 11)]  
max\_depth.append(None)  
# Minimum number of samples required to split a node  
min\_samples\_split = [2, 5, 10]  
# Minimum number of samples required at each leaf node  
min\_samples\_leaf = [1, 2, 4]  
# Method of selecting samples for training each tree  
bootstrap = [True, False]

# Create the random grid  
random\_grid = {'n\_estimators': n\_estimators,  
 'max\_features': max\_features,  
 'max\_depth': max\_depth,  
 'min\_samples\_split': min\_samples\_split,  
 'min\_samples\_leaf': min\_samples\_leaf,  
 'bootstrap': bootstrap}pprint(random\_grid)

Now, we instantiate the random search and fit it like any Scikit-Learn model:

# Use the random grid to search for best hyperparameters  
# First create the base model to tune  
rf = RandomForestRegressor()  
# Random search of parameters, using 3 fold cross validation,   
# search across 100 different combinations, and use all available cores  
rf\_random = RandomizedSearchCV(estimator = rf, param\_distributions = random\_grid, n\_iter = 100, cv = 3, verbose=2, random\_state=42, n\_jobs = -1)# Fit the random search model  
rf\_random.fit(train\_features, train\_labels)

rf\_random**.**best\_params\_

#### Evaluate the Default Model

<https://github.com/WillKoehrsen/Machine-Learning-Projects/blob/master/random_forest_explained/Improving%20Random%20Forest%20Part%202.ipynb>

In [14]:

base\_model **=** RandomForestRegressor(n\_estimators **=** 10, random\_state **=** 42)

base\_model**.**fit(train\_features, train\_labels)

base\_accuracy **=** evaluate(base\_model, test\_features, test\_labels)

Model Performance

Average Error: 3.9199 degrees.

Accuracy = 93.36%.

#### Evaluate the Best Random Search Model

In [15]:

best\_random **=** rf\_random**.**best\_estimator\_

random\_accuracy **=** evaluate(best\_random, test\_features, test\_labels)

print('Improvement of {:0.2f}%.'**.**format( 100 **\*** (random\_accuracy **-** base\_accuracy) **/** base\_accuracy))

**Final Model**

The final model from hyperparameter tuning is as follows.

In [26]:

final\_model **=** grid\_search**.**best\_estimator\_

print('Final Model Parameters:\n')

pprint(final\_model**.**get\_params())

print('\n')

grid\_final\_accuracy **=** evaluate(final\_model, test\_features, test\_labels)

RFECV

<https://machinelearningmastery.com/rfe-feature-selection-in-python/>

Hyperparameter Tuning the Random Forest in Python

<https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74>

Problemas dos dados desbalanceados

<https://www.ele.uri.edu/faculty/he/research/ImbalancedLearning/ImbalancedLearning_lecturenotes.pdf>

Feature Selection

Seleção de características é o processo de seleção do número óptimo de características a partir de um conjunto maior de características. Há várias vantagens deste processo de selecção de características e também há várias técnicas disponíveis para este processo de selecção de características. Neste núcleo, vamos analisar estas vantagens e várias técnicas para a selecção de características.

How to deal with imbalanced data

<https://arxiv.org/pdf/2108.00071.pdf>

<https://www.cs.waikato.ac.nz/~ml/publications/1998/Hall-Smith98.pdf>

Feature selection using random forest

1. Prepare the dataset.

2. Train a random forest classifier

3. Identify the most important features

4. Create a new ‘limited featured’ dataset containing only those features

5. Train a second classifier on this new dataset

6. Compare the accuracy of the ‘full featured’ classifier to the accuracy of the ‘limited featured’ classifier

<https://chrisalbon.com/code/machine_learning/trees_and_forests/feature_selection_using_random_forest/>

A modified k-fold cross-validation, called stratified k-fold cross-validation, is more suitable for imbalanced classification problems. Stratified k-fold or Repeated Stratified k-fold preserves the imbalanced class distribution in each fold. Either of these k-fold techniques can then be used in conjunction with cross\_val\_score to predict and evaluate each fold. All the evaluation metrics explained earlier can be used as a scoring parameter in cross\_val\_score. Refer to the full list here. A practical demonstration is included later in the section of cost-sensitive learning.

<https://towardsdatascience.com/how-to-effectively-predict-imbalanced-classes-in-python-e8cd3b5720c4>

AUC ROC curve precision recall curves

<https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/>

chrome-extension://dagcmkpagjlhakfdhnbomgmjdpkdklff/enhanced-reader.html?openApp&pdf=https%3A%2F%2Farxiv.org%2Fpdf%2F2111.12140.pdf

The amount of data for machine learning (ML) applications is constantly growing. Not only the number of observations, especially the number of measured variables (features) increases with ongoing digitization. Selecting the most appropriate features for predictive modeling is an important lever for the success of ML applications in business and research. Feature selection methods (FSM) that are independent of a certain ML algorithm—so-called filter methods—have been numerously suggested, but little guidance for researchers and quantitative modelers exists to choose appropriate approaches for typical ML problems

<https://medium.com/mlearning-ai/feature-selection-using-filter-method-python-implementation-from-scratch-375d86389003>

Diagram

Description automatically generated

<https://towardsdatascience.com/feature-selection-in-python-using-filter-method-7ae5cbc4ee05>

Feature selection in Python using the Filter method

Filter Method for Feature selection

The filter method ranks each feature based on some uni-variate metric and then selects the highest-ranking features. Some of the uni-variate metrics are:

* variance: removing constant and quasi constant features
* chi-square: used for classification. It is a statistical test of independence to determine the dependency of two variables.
* correlation coefficients: removes duplicate features
* Information gain or mutual information: assess the dependency of the independent variable in predicting the target variable. In other words, it determines the ability of the independent feature to predict the target variable

<https://www.analyticsvidhya.com/blog/2020/10/a-comprehensive-guide-to-feature-selection-using-wrapper-methods-in-python/>

O processo de regularização visa adicionar uma penalidade para os parâmetros muito grandes, garantindo dessa forma que o aprendizado do modelo seja generalizado e não específico. LASSO também é conhecido como L1. Utilizando LASSO é possível reduzir a zero os coeficientes de algumas variáveis e dessa forma, descartá-la-á.

E caberá ao profissional decidir qual o valor de corte que proporcionará os rácios de probabilidade e os valores de sensibilidade e especificidade que têm o maior valor clínico no diagnóstico de qualquer desordem.

Em geral, um teste de diagnóstico tende a ser avaliado por duas destas medidas, FVP (sensibilidade) e FVN (especificidade). Metz [58], define sensibilidade como sendo a probabilidade de decidir se a doença em questão está presente quando de facto está presente, e especificidade como sendo a probabilidade de decidir se a doença em questão está ausente quando, de facto está ausente. Em termos de diagnóstico, poder-se-á definir sensibilidade como a capacidade que um teste tem para detectar a doença no indivíduo, e a especificidade como a capacidade que o teste tem para excluir os indivíduos isentos de doença. Assim, valores de corte elevados, conduzem a um teste pouco sensível e muito específico, por outro lado, valores de corte baixos, conduzem a um teste muito sensível e pouco específico

ROC

Devido à capacidade discriminatória intrínseca à metodologia ROC, esta é ainda usada no apoio a decisões médicas com o intuito de prever e classificar problemas de foro médico. Os testes de diagnóstico, e tendo em conta os diversos tipos, numa vertente médica são a primeira fase de controlo, e são cruciais para permitirem geralmente um tratamento adequado. Porém quando há uma tomada de decisão sobre a conduta do tratamento há sempre uma relação de compromisso, designada por Metz por uma relação de custo/benefício [22]. Pois as decisões tomadas podem estar corretas ou incorretas [17] [22].

Como já referido, a curva ROC pode servir para representar o desempenho do classificador, e para comparar classificadores, pode-se transformar o desempenho ROC num valor escalar, representando o desempenho esperado. Um método comum é calcular a área abaixo da curva ROC, ou simplesmente AUC *(Area Under Curve)* (Fan et al., 2006)*.*

Uma das formas de avaliar a capacidade de discriminação da curva ROC é através da medição da área baixo da curva. Quanto mais a curva se aproximar do canto superior esquerdo (área de 1) maior será a sua capacidade de discriminação. Se esta se aproximar da diagonal (área de 0.5) a sua capacidade de discriminação é nula, enquanto, que se assumir valores inferiores a curva ROC apresenta capacidade de discriminação, mas está a discriminar contrariamente (e.g. identifica melhor como não doente aquele que é doente e vice-versa).

Uma vez que a CUA é uma porção da área do quadrado da unidade, o seu valor estará sempre entre 0 e 1,0. No entanto, como a adivinhação aleatória produz a linha diagonal entre (0, 0) e (1, 1), que tem uma área de 0,5, nenhum classificador realista deve ter um CUA inferior a 0,5.

A área abaixo da curva (AUC) está associada ao poder discriminante de um teste diagnóstico, correspondendo à probabilidade de identificar corretamente a ocorrência ou não de *delirium*. Quando o valor da AUC é igual a 0,5, isto é, a FFP é igual à FVP, o classificador não apresenta poder discriminate, porém, à medida que o valor da AUC aumenta, isto é, o valor da FVP aumenta e o valor da FFP diminui, aumenta o poder discriminante de um classificador. Até ao valor máximo que este pode atingir com uma AUC = 1, significando uma discriminação perfeita.

Aquando da observação do gráfico, o cenário ideal implica encontrar uma área sob a curva ROC perto de 1, uma vez que, quanto mais próxima estiver a curva do canto superior esquerdo, mais verdadeiros positivos (VP) e menos falsos negativos (FN) serão obtidos (Stoltzfus, 2011).

A determinação de um valor de corte "ideal" é quase sempre um compromisso entre sensibilidade (verdadeiros positivos) e especificidade (verdadeiros negativos). Como ambos mudam com cada valor de "corte", torna-se difícil para o leitor imaginar qual é o corte ideal. A curva ROC oferece uma ilustração gráfica destes trade-offs em cada "cut-off" para qualquer teste de diagnóstico que utilize uma variável contínua.3 Idealmente, o melhor valor de "cut-off" proporciona tanto a maior sensibilidade como a maior especificidade, facilmente localizada na curva ROC ao encontrar o ponto mais alto no eixo vertical e o mais à esquerda no eixo horizontal (canto superior esquerdo) (Fig. 1)

De um modo geral, a curva ROC é uma representação gráfica bidimensional, onde estão representados a fração de verdadeiros positivos (FVP), no eixo nas ordenadas, e da fração de falsos positivos (FFP), no eixo das abcissas. A análise da curva ROC permite estudar a variação da sensibilidade e especificidade para cada valor de corte*.* O objetivo desta análise é identificar ou confirmar a qualidade do ajustamento do modelo. Aquando da observação do gráfico, o cenário ideal implica encontrar uma área sob a curva ROC (AUC) perto de 1, uma vez que, quanto mais próxima estiver a curva do canto superior esquerdo, mais verdadeiros positivos (VP) e menos falsos negativos (FN) serão obtidos (Stoltzfus, 2011).

<https://repositorium.sdum.uminho.pt/bitstream/1822/40535/3/Maria%20Filipa%20Torres%20Goncalves%20Flores%20Mourao.pdf>

Para o cálculo desta medida utiliza-se a proporção entre os VP e a soma dos VP com os FP, isto é os valores positivos previstos corretamente, e os valores negativos etiquetados incorretamente pelo modelo, traduzindo-se pela equação 6.3 (J. Han et al., 2012). A especificidade corresponde à probabilidade de ocorrer um resultado negativo sabendo que o indivíduo é saudável, ou seja, à fração de

negativos entre os saudáveis.

|  |  |  |
| --- | --- | --- |
|  |  | (6.5.6) |

z

De forma a contornar este problema, foram desenhados métodos que permitem uma redução da dimensão dos dados.