International Journal of Computational and Experimental Science and ENgineering (IJCESEN)
Vol. 6-No.3 (2020) pp. 138-144
http://dergipark.org.tr/en/pub/ijcesen

ISSN: 2149-9144

Research Article

Medical Data Analysis for Different Data Types

Süleyman EKEN*

Kocaeli Üniversity, Technology Faculty, Department of Information Systems Engineering, 41001, Kocaeli-Turkey

* **Corresponding Author** : suleyman.eken@kocaeli.edu.tr
ORCID: 0000-0001-9488-908X

Article Info:

DOI: 10.22399/ijcesen.780174
Received : 13 August 2020
Accepted : 17 September 2020

Keywords

Medical data informatics
Electronic health records
Data science
EDA
Reproducible research

Abstract:

Many discoveries and decisions in science are now being made on the basis of analyzing datasets. To gain useful information from raw medical data, data analytic uses insights to benefit the entire lifecycle of medical data. In this paper, medical data analysis notebooks are presented for collaborative and reproducible research. They provide a broad and practical introduction to medical data analysis with different data types such as images and texts. We aim to provide Jupyter notebooks to help those new to the medical data analysis field. Three exploratory coding activities including different data types are introduced: (i) Building, evaluating and interpreting deep learning models with EHR data, (ii) 2D mammogram medical imaging data analysis using CNNs for dense breasts classification, and (iii) Label recognition in radiology reports. Jupyter notebooks are useful for learning how to analyze different medical datasets and identify patterns that will improve any hospitals' and clinicians' computer-aided medical decision-making process. Leveraging advances in exploratory data analysis in healthcare requires collaboration between clinicians and data scientists

1. Introduction

In recent years, the rise in the use of social media and digitalization of social and economic activity was the source of unprecedented amounts of data, mostly in an unstructured form: weblogs, videos, speech recordings, photographs, e-mails, tweets [1, 2] . This data can be analysed to extract relevant and insightful information on business or society, and that is thanks to big data analysis tools and techniques.

This is why we believe that big data analysis and related technologies deep learning, cloud computing and scalable machine learning are a cornerstone to a medical data analysts. That being said, in addition to the benefits it offers, big data comes with many hurdles as well: (i) With continuously increasing interest from different sectors, there is also an increasing demand for skilled data experts, with colleges unable to satisfy that need, nor to incorporate big data within themselves causing millions of students to miss on such an opportunity. (ii) The available machines can not keep up with the rate at which data is generated leading to crushes and lower quality of the analysis. And that, in turn, creates a need for more scalable systems to store and process data. (iii) The sheer size of data in itself makes it prone to mistakes and data loss, for example, think of an average establishment trying keeping records of all its thousands of students across several categories. (iv) Data safety and integrity becomes a paramount worry to both authorities and institutions. The existing security protocols are not developed with big data in mind making them unfit for it, and the nature of the data and its continuous updating make it neither easy nor cheap to manage.

With these difficulties in mind, we develop an exploratory medical data analysis notebooks, aiming at familiarizing them with the key technologies employed to store, manipulate and analyze medical data. In this work, we cover the basic tools for statistical analysis, the Python programming language, and various machine learning algorithms.

2. Material and Methods

Computational notebooks are an open-source interactive web-based application streamlining and simplifying literate programming [3, 4]. Initially, Jupyter notebooks supported only Python. Then it extended the support to different programming languages via Jupyter kernels. Notebooks combine text, code, and visualization in a single JSON document. It is parsed and displayed in a human-friendly format. The code is split into logical sections named "code cells". So, they can be executed interactively. Another worth noting benefit of running individual code chunks is that one can choose to run some parts only once while others can be repeatedly executed. Many researchers and data analysts are rapidly adopting this new medium.

2.1. Python for Data Analysis & Visualization

Pandas is one of the most popular Python libraries that provide high-performance, easy-to-use data structures and data analysis tools [5]. Using Pandas, researchers can handle of missing data, align data automatically or explicitly, perform group-by operations, convert differently-indexed data into DataFrame objects, perform slicing, indexing, and subset of large data sets, merge and join data sets, and reshaped and pivoted of data sets. Also, Python has several visualization libraries, namely Matplotlib [6], Seaborn [7], and Folium [8] for presenting data visually.

2.2. Python for Machine Learning

Python has machine learning library named scikit-learn [9]. It has many learning algorithms such as supervised (regression, classification) and unsupervised (clustering) models. Researchers can analyze the performance of the model using different metrics and deploy methods to select between models. Also it enables users to engineer the right features for a given problem on any dataset.

2.3. Unstructured Data Analysis

We first need to understand what we are dealing with. At this point, we can divide the data structures into three main categories: structured, semi-structured and unstructured data. Any data that we can store, process and access in a relational and meaningful way can be classified as structured data. The data are stored in a relational and clean way in databases. The structure of the data and its relationship with each other is known. As an example of structured data, we can consider the data that e-commerce sites, applications and businesses produce and store in their daily operations. Data that has no relation to each other, is unprocessed and does not have a structure or format can be called unstructured data. The data size is large and it is not easy to make the data meaningful and structured. The data can include a mix of text files, videos and images. We can describe semi-structured as a combination of structural and non-structural data. If the data can be associated and easily processed but not configured, it can be classified as a file, not on a system, or semi-structured data if on a service. Semi-structured data includes data that contains structured data properties but is not stored in traditional databases. Examples of semi-structured data are XML, text and web services [10]. Python Natural Language Toolkit (NLTK) [11] is mostly used for processing and handling text and Scikit-learn [13] includes machine learning applications so it can be used all areas where machine learning can be applied.

3. Coding Activities

This section introduces the three exploratory coding activities. Traditionally, an educational setting had a well-defined learning domain, with students as users at the receiving end, while the domain experts and teachers guide them through the learning process with the help of suitable technology prepared by them. Following sub-sections include a brief description of the laboratories.

3.1. Setup Data Analysis Environment

Here are instructions for getting their system ready to follow along with the core tools and libraries we use: Python has become the go-to programming language for data science and machine learning. So, the first step is installing Python. Anaconda is a data science distribution for Python and R. It is also a package manager and it will also help us to create environment for data science. Also, Anaconda is the recommended way to install Jupyter (IPython) Notebooks [4]. Jupyter Notebook is the interactive environment where you will be writing all your code, creating files and doing visualizations as well. Anaconda comes with a lot of required data science packages pre-installed. A new virtual environment (medical_data_analysis) is created and all the required packages are installed. After activating virtual environment, some Python packages are installed for data science which are most frequently required for other exploratory coding activities: aequitas [12], sqlite3, ipython-sql, numpy, pandas, matplotlib, seaborn, plotly, scipy, scikit-learn,

scikit-image, tensorflow, nltk, transformers, and gensim.

3.2. Building, Evaluating and Interpreting Models with EHR Data

This sub-section include interesting tools to work with EHR (Electronic Health Record) data in artificial intelligence. Firstly, we get hands-on with using Tensorflow DenseFeatures for building a simple regression model. Next, we first review some common evaluation metrics for EHR models and then learn to implement brier scores for model evaluation. Then, we conduct a demographic bias analysis and become familiar with a framework out of the University of Chicago called Aequitas [13]. We use this Aequitas for group bias and fairness disparity analysis.

In this part, we focus on just building a simple Tensorflow regression model with TF DenseFeatures on Heart Disease Data Set to predict resting blood pressure (trestbps field in the dataset). Tensorflow DenseFeatures, combining features, like those from the TensorFlow Feature Columns API, into a dense representation for the model. We can only use certain TF Feature Columns with DenseFeatures: numeric, embedding, bucketized, and indicator columns. We use the Sequential API. Replacing unknown values to NaN, dataset concatenation, droping missing values, and selecting important features are implemented in pre-processing step. After this step, the first five columns of data are as shown in Table 1.

Table 1. *An overview from data*

sex	age	trestbps	thalach
male	32.0	95	127
male	34.0	115	154
male	36.0	110	125
female	38.0	105	166
female	38.0	110	156

Tensorflow feature columns can be combined into a list that can be passed to the DenseFeatures layer. Then, we add this 'dense_feature_layer' as the first layer to the model and this will handle the combining of feature inputs to the model.

The dataset was randomly divided into two independent datasets with 80% and 20% for training and testing, respectively. The batch size, epoch number, and root mean square error propability are 128, 1000, and 0.001 respectively. We evaluate the success/failure of model in terms of two measures such as Mean Absolute Error (MAE) and Mean Squared Error (MSE). Their values are 14.76 and 339.49, respectively. Table 2 shows the converted regression problem to binary classification problem. Here, score is 1 if predicted resting blood pressure is higher than 130. Also, label value is the the truth value. It consists 5 of 46 test samples. Table 3 shows performance results of binary classification.

Table 2. *Five test samples of binary classification*

pred	actual_value	score	Label_value
139.877350	95.0	1	0
132.572052	115.0	1	0
139.877350	170.0	1	1
132.572052	160.0	1	1
126.265846	140.0	0	1

It is important to note that bias within models can restrict or limit patient access to key medical benefits from government aid programs. Programs using machine learning algorithms to help automate approvals of key government benefits are becoming more commonplace. However, it is just as important to consider how bias can unintentionally occur. Another reason that you want to consider bias in models is that in order to create better treatments for patients we need to find better ways to select and recruit patients that represent the wider population that a drug/treatment would be targeted for.

Table 3. *Performance results of binary classification*

	precision	recall	f1-score	support
0	0.22	0.21	0.22	19
1	0.46	0.48	0.47	27
accuracy			0.37	46
macro avg	0.34	0.35	0.34	46
weighted avg	0.36	0.37	0.37	46

In many cases, there may be systemic biases for key groups and while this cannot always be prevented, bringing awareness of limitations and biases can give a more accurate picture of a treatment's effectiveness across different demographics. Unintended bias that is not intentional and often is not even apparent to the creator of a model and it represents the unconscious or unintentional biases that come with the artificial intelligence models. Demographic Group Bias Analysis includes selecting groups to be analyzed, preparing data, and analyzing different metrics with the groups. Here, 'score' and 'label_value' fields are used for boilerplate preprocessing input data. After this step, Table 4 shows summarized metric view for group counts and absolute ones. Opening forms of abbreviations in Table 4 are as following: tpr: true positive rate, tnr: true negative rate for :

Table 4. *Summarized metric view*

attribute_name	attribute_value	tpr	tnr	for	fdr	fpr	fnr	npv	precision	ppr	pprev	prev
pred	126.82-128.04	0.0	1.00	0.67	NaN	0.00	1.0	0.33	NaN	0.00	0.00	0.67
pred	128.04-134.04	1.0	0.00	NaN	0.50	1.00	0.0	NaN	0.50	0.57	1.00	0.50
pred	136.34-141.51	1.0	0.00	NaN	0.42	1.00	0.0	NaN	0.58	0.43	1.00	0.58
actual_value	120.00-134.00	0.4	0.33	0.60	0.67	0.67	0.6	0.40	0.33	0.21	0.55	0.45
actual_value	134.00-141.50	0.7	NaN	1.00	0.00	NaN	0.3	0.00	1.00	0.25	0.70	1.00
actual_value	141.50-178.00	0.5	NaN	1.00	0.00	NaN	0.5	0.00	1.00	0.21	0.50	1.00
actual_value	95.00-120.00	NaN	0.31	0.00	1.00	0.69	NaN	1.00	0.00	0.32	0.69	0.00

false omission rate, fdr: false discovery rate, fpr: false positive rate, fnr: false negative rate, npv: negative predictive value, ppr: predictive positive ratio. Figure 1 illustrates the all group metrics. Reference group information is as following: 'pred':'126.82-128.04', 'actual_value':'134.00-141.50'. Figure 2(a) shows false positive rate disparity for actual_value field. Figure 2(b) shows true positive rate disparity for pred field.

Preds valued between 128.04-134.04 and 136.34-141.51 are over 10x more likely to be truely identified than reference group (preds valued between 126.82-128.04). Figure 3 shows FPR fairness. Red bars mean false/not fair determination.

3.3. 2D Medical Imaging Data Analysis

Different clinical imaging tools are available. X-rays, computed tomography (CT), magnetic resonance imaging (MRI), and ultrasound are such technologies. X-rays down at the body from a single direction to capture a single image.

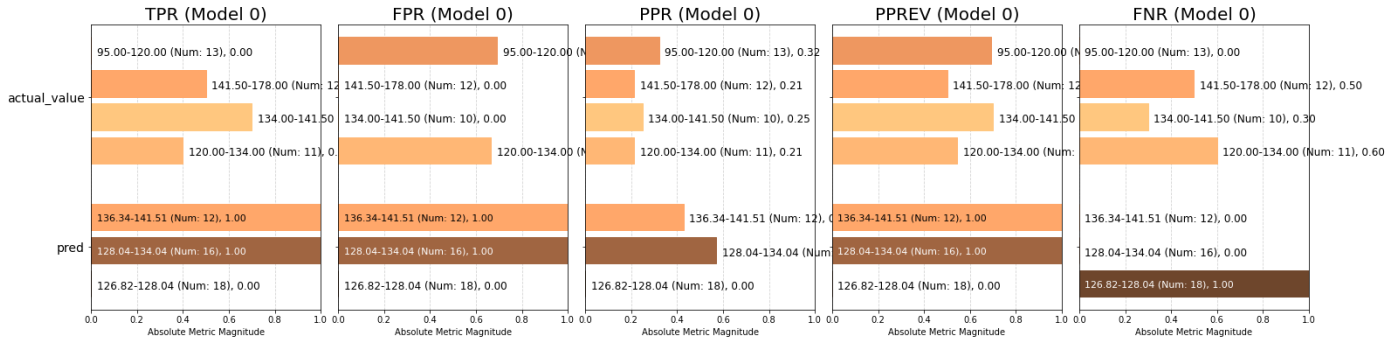

Figure 1. *All group metrics plot*

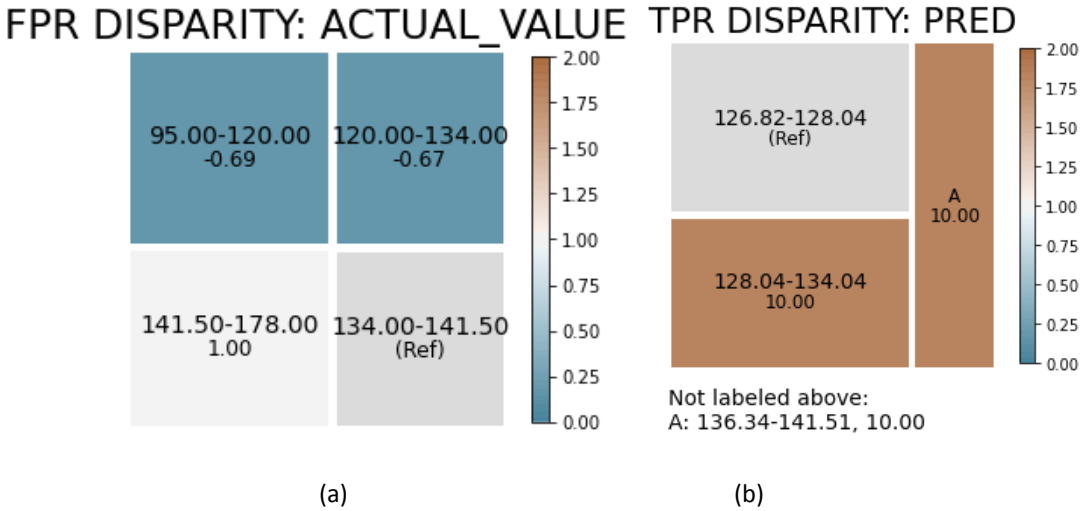

Figure 2. *FPR and TPR disparity for different attributes*

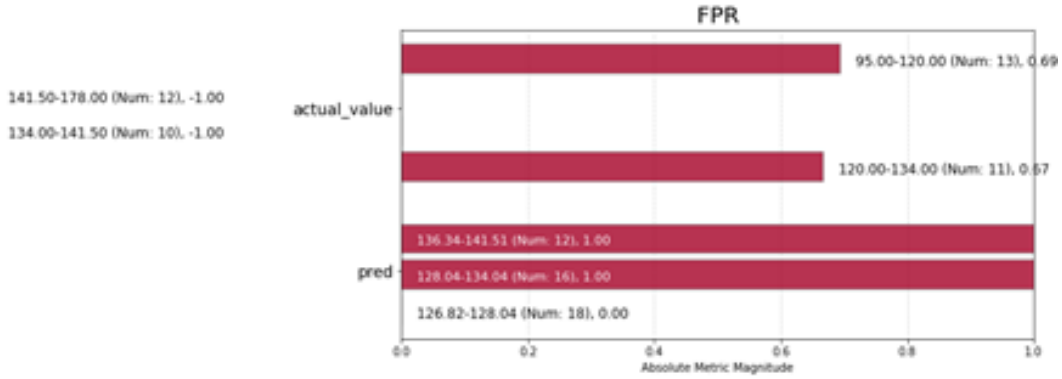

Figure 3. *FPR fairness plot*

CT uses x-ray, but they emit x-rays from many different angles around the human body to capture more detail from more different angles. MRI uses strong magnetic fields and radio waves to create images of areas of the body from all different angles. Ultrasound utilizes high-frequency sound waves beyond the audible limit of human hearing to generate images. Out of these different

imaging tools, x-ray and ultrasound are the only 2D imaging tool. Every imaging center and hospital have a Picture Archiving and Communication System (PACS). These systems allow for all medical imaging to be stored in the hospital's servers and transferred to different departments throughout the hospital. Classification, segmentation [14], and localization [15] the most important types of 2D imaging algorithms. A mammogram shows how dense breasts are. Women with dense breasts have a higher risk of getting breast cancer. In this part, we classify mammogram images into two classes: dense and fatty. Firstly, some preprocessing steps such as removing potential noise from images (e.g. background extraction), normalization (zero-mean, standardization), image augmentation, and resizing imaged for CNN architecture's required input. Here, image augmentation includes horizontal flip, height shift and width shift, rotation, shearing and zooming. After augmentation step, Figure 4 shows some examples of our augmented training data. We use VGG16 model [16] with pre-trained ImageNet weights. Then, fine-tuning is implemented. Layers types and other parameters of used model is given in Figure 5. Optimizer, loss function, learning rate, and epoch number are Adam(lr=1e-4), binary crossentropy, binary accuracy, and 10 respectively. Binary accuracy of the model is 0.66.

3.4. Label Recognition in Radiology Reports

In this part, we extract disease labels information

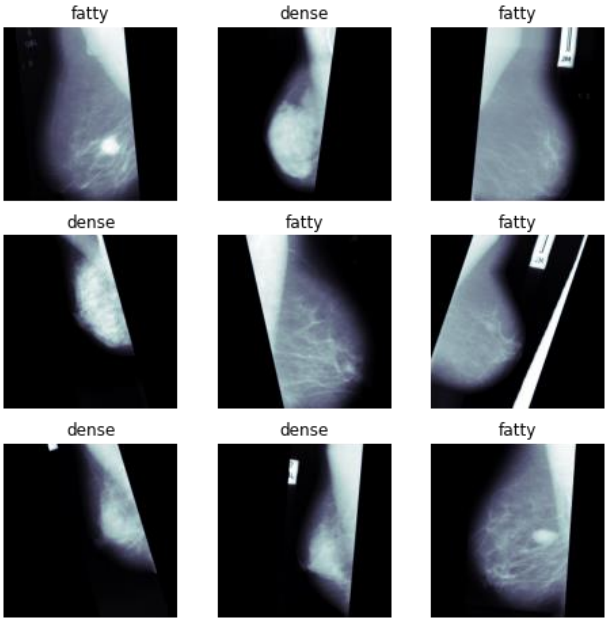

Figure 4. *Manipulated data prior to training*

from unstructured medical text (clinical reports) for patients [17]. We use Stanford report test document. It consists of 1,000 X-ray reports that have been manually labeled by a board certified radiologist for the presence or lack of presence of different pathologies (see Figure 6 for an example report). Figure 7 shows the first five rows of dataset. To label dataset is to search for presence of different keywords in the impression text. A list of relevant keywords for each pathology is prepared for detecting the presence of each label. For instance, related keywords for airspace opacity are: opaci, decreased translucency, increased density, airspace disease, air-space disease, air space disease, infiltrate, infiltration, interstitial marking, interstitial pattern, interstitial lung, reticular pattern, reticular marking, reticulation, parenchymal scarring, peribronchial thickening, wall thickening, scar.

Layer (type)	Output Shape	Param #
model_1 (Model)	(None, 7, 7, 512)	14714688
flatten_1 (Flatten)	(None, 25088)	0
dropout_1 (Dropout)	(None, 25088)	0
dense_1 (Dense)	(None, 1024)	25691136
dropout_2 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 512)	524800
dropout_3 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 256)	131328
dense_4 (Dense)	(None, 1)	257

Total params: 41,062,209
Trainable params: 28,707,329
Non-trainable params: 12,354,880

Figure 5. *Used CNN model architecture*

Comparison: None

Indication: Burmese male has complete TB treatment

Findings: Both lungs are clear and expanded with no infiltrates. Basilar focal atelectasis is present in the lingula. Heart size normal. Calcified right hilar XXXX are present

Impression: No active disease.

MeSH

Major

Pulmonary Atelectasis / lingula / focal
Calcinosis / lung / hilum / right

CHEST 2V FRONTAL/LATERAL XXXX, XXXX XXXX PM

Figure 6. *An example of clinical report from dataset*

	Report Impression	Cardiomegaly	Lung Lesion	Airspace Opacity	Edema	Consolidation	Pneumonia	Atelectasis	Pneumothorax	Pleural Effusion	Pleural Other	Fracture
0	\n \n1.mild pulmonary edema, and cardiomegaly....	True	False	False	True	False	False	True	False	True	False	False
1	\n \n1.unremarkable cardiomediastinal silhouet...	False	False	True	False	False	False	False	False	False	False	True
2	\n1. lines and tubes are unchanged in position...	False	False	True	False	False	False	False	False	False	False	False
3	\n1. postoperative portable film with a right-...	False	False	True	True	False	False	True	True	False	False	False
4	\n \n1.single frontal view of the chest demons...	False	False	True	False	False	False	False	False	True	False	False

Figure 7. *Samples from unstructured clinical reports*

For impression texts, all label can be retrieved. For impression text of "Diffuse Reticular Pattern, which can be seen with an atypical infection or chronic fibrotic change. no Focal Consolidation.", retrieved labels are Airspace Opacity, Consolidation, and Pneumonia. Performance of labeler can be evaluated with F1-score metric regarding retrieved and expected labels. Table 5 shows the labeler performance. Using some preprocessing operations such as converting pattern such as "and/or" to "or", replacing repeated whitespace with just one space, and removing redundant punctuation, performance can be improved. Also, negative mentions and dependency parsing can be used to improve the performance of labeler.

Table 5. *Samples from unstructured clinical reports*

Label	F1-score
Cardiomegaly	0.718
Lung Lesion	0.641
Airspace Opacity	0.923
Edema	0.708
Consolidation	0.270
Pneumonia	0.369
Atelectasis	0.646

4. Discussions

4.1. Study Limitations

Our study has several limitations. There are many types of medical imaging technologies. First, we use only mammogram images in example of 2D imaging data process. Maybe, other 2D medical images or 3D images can be used to extend this study. Second, we present three reproducible notebook activities. We aim to encourage new to the field begin their informatics projects. We plan to make more collaboration with clinicians, radiologist, pathologists, and biomedical engineering about medical diagnosis, prognosis, and treatment activities.

4.2. Study Significance

Working in multidisciplinary teams is a long championed concept to address challenges such as diabetic retinopathy, skin cancer detection from images, smoking status and medication identification from unstructured patient records for artificial intelligence research, and to increase its impact in medicine. In this context, we present three different activities with this study. As a results, multidisciplinary teams can set clear vision

for collaborative and reproducible research, create clear data inclusion and exclusion criteria for initial data draw, and insight for what natural language processing/machine learning methods might be better at addressing different clinical problems.

5. Conclusion

Exploratory notebooks specifically focused on the medical data analysis have been presented in this paper. We firstly give theoretical background information about medical data analysis using different environments/structures such as spreadsheets, SQLite, and Python based. Then, three exploratory coding activities are presented. The first one focuses on EHR data analysis using Python wrangling skills and deep learning models. The second one focuses on image processing based medical data analysis. The last one is unstructured data analysis on clinical reports. As a result, we aim to help those new to the field begin their informatics projects.

Also, we will be taking into account the clinical and industry stakeholders' requirements while presenting the notebooks. These requirements can be listed as: innovative medical data analysis applications for different purposes with different types of input, methods for pattern analysis, supporting for big data platforms that offer distributed run environments.

References

[1] Aggarwal, A. K. (2019). Opportunities and challenges of big data in public sector. In Web services: Concepts, methodologies, tools, and applications (pp. 1749-1761). IGI Global. doi: 10.4018/978-1-4666-9649-5.ch016

[2] Oussous, A., Benjelloun, F. Z., Lahcen, A. A., & Belfkih, S. (2018). Big Data technologies: A survey. Journal of King Saud University-Computer and Information Sciences, 30(4), 431-448. doi: 10.1016/j.jksuci.2017.06.001

[3] Shen, H. (2014). Interactive notebooks: Sharing the code. Nature, 515(7525), 151-152. doi: 10.1038/515151a

[4] Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B. E., Bussonnier, M., Frederic, J., ... & Ivanov, P. (2016, May). Jupyter Notebooks-a publishing format for reproducible computational workflows. In ELPUB (pp. 87-90). doi:10.3233/978-1-61499-649-1-87

[5] McKinney, W. (2011). pandas: a foundational Python library for data analysis and statistics. Python for High Performance and Scientific Computing, 14(9). doi:

[6] Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing In Science Engineering 9, 3. 90-95. doi: 10.1109/MCSE.2007.55

[7] Embarak, O. (2018). Data Visualization. In Data Analysis and Visualization Using Python (pp. 293-342). Apress, Berkeley, CA. 10.1007/978-1-4842-4109-7_7

[8] Cuttone, A., Lehmann, S., & Larsen, J. E. (2016). geoplotlib: a Python Toolbox for Visualizing Geographical Data. arXiv preprint arXiv:1608.01933.

[9] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, 2825-2830.

[10] Leung, C. K. S. (2019). Big data analysis and mining. In Advanced Methodologies and Technologies in Network Architecture, Mobile Computing, and Data Analytics (pp. 15-27). IGI Global. doi: 10.4018/978-1-5225-2255-3.CH030

[11] Loper, E., & Bird, S. (2002). NLTK: the natural language toolkit. arXiv preprint cs/0205028.

[12] Saleiro, P., Kuester, B., Hinkson, L., London, J., Stevens, A., Anisfeld, A., ... & Ghani, R. (2018). Aequitas: A bias and fairness audit toolkit. arXiv preprint arXiv:1811.05577.

[13] Detrano, R., Janosi, A., Steinbrunn, W., Pfisterer, M., Schmid, J. J., Sandhu, S., ... & Froelicher, V. (1989). International application of a new probability algorithm for the diagnosis of coronary artery disease. The American journal of cardiology, 64(5), 304-310. doi: 10.1016/0002-9149(89)90524-9

[14] Strzelecki, M., Szczypinski, P., Materka, A., & Klepaczko, A. (2013). A software tool for automatic classification and segmentation of 2D/3D medical images. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 702, 137-140. doi: 10.1016/j.nima.2012.09.006

[15] de Vos, B. D., Wolterink, J. M., de Jong, P. A., Viergever, M. A., & Išgum, I. (2016, March). 2D image classification for 3D anatomy localization: employing deep convolutional neural networks. In Medical imaging 2016: Image processing (Vol. 9784, p. 97841Y). International Society for Optics and Photonics. doi: 10.1117/12.2216971

[16] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[17] Irvin, J., Rajpurkar, P., Ko, M., Yu, Y., Ciurea-Ilcus, S., Chute, C., ... & Seekins, J. (2019, July). Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, pp. 590-597). 10.1609/aaai.v33i01.3301590