Data Science Pastime

1

Why?

1) A lot of legacy projects
2) Low developers level in field of math
3) Low developers level in field of technologies which not related to the company projects
4) Unstable situation of company

For what?

1) To increase math skills
2) To increase technologies stack of developers (company)
3) The occasion to learn something new
4) Not waste time
5) To play on Kaggle (for example)
6) Just for fun

Agenda

1) Intro

2) Technologies stack
   2.1) Python
   2.2) Numpy
   2.3) Pandas
   2.4) Matplotlib, Seaborn
   2.5) Sklearn
   2.6) TensorFlow

3) Data
   3.1) Data example
   3.2) Analyze by describing data
   3.3) Assumption based on data analysis
   3.4) Analyze by visualization data
   3.5) Wrangle data

4) Machine learning
   4.1) Model
   4.2) Model examples
   4.3) Model training
   4.4) Model evaluation

5) TensorFlow
   5.1) Short overview
   5.2) Custom graph

6) Conclusions

7) References

8) Q&A

Intro

Data science

Is an interdisciplinary field about scientific methods, processes, and systems to extract knowledge or insights from data in various forms, either structured or unstructured.

Data science is a "concept to unify statistics, data analysis and their related methods" in order to "understand and analyze actual phenomena" with data.

Data mining

Is the computing process of discovering patterns in large data sets involving methods at the intersection of machine learning, statistics, and database systems.

1. Anomaly detection
2. Association rule learning
3. Clustering
4. Classification
5. Regression
6. Summarization

Technologies stack

1) Python

Python is a widely used high-level programming language for general-purpose programming.

Python features a dynamic type system and automatic memory management and supports multiple programming paradigms, including object-oriented, imperative, functional programming, and procedural styles. It has a large and comprehensive standard library

```
print("Hello, World!")
```

Zen of Python

```
>> import this
```

Technologies stack

2) Numpy

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- a powerful N-dimensional array object
- sophisticated (broadcasting) functions
- tools for integrating C/C++ and Fortran code
- useful linear algebra, Fourier transform, and random number capabilities

```
a = np.array([[1,2,3], [4,5,6], [7,8,9], [10, 11, 12]])
print(a)
b = np.array([0, 2, 0, 1])
print(a[np.arange(4), b])
a[np.arange(4), b] += 10
print(a)
```

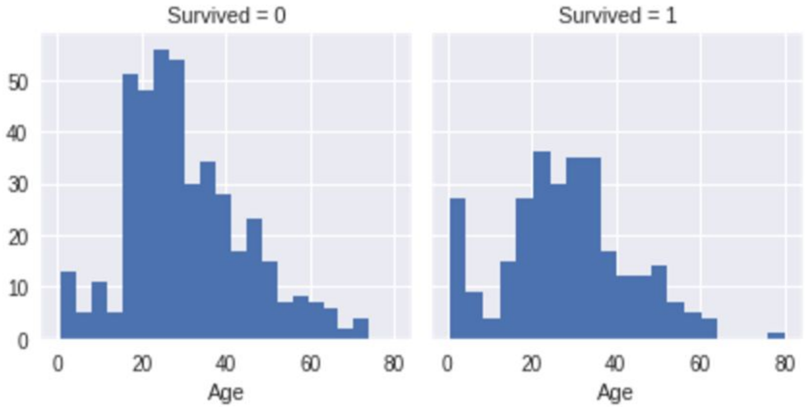
Technologies stack

4) Matplotlib, Seaborn

Is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Seaborn is a Python visualization library based on matplotlib.

```
import seaborn as sns
import matplotlib.pyplot as plt


g = sns.FacetGrid(data, col='Survived')
g.map(plt.hist, 'Age', bins=20)
```



Technologies stack

3) Pandas

Is an open source library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

```
data = pd.read_csv(r"Titanic/data/train.csv")
print(data.columns.values)
data.head()
data.tail()
data.info()
data.describe()
data.describe(include=['O'])
data[['Pclass', 'Survived']].groupby(['Pclass'], as_index=False).mean().sort_values(by='Survived', ascending=False)
data = data.drop(['Ticket', 'Cabin'], axis=1)
print(data)
```

Technologies stack

5) Sklearn

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib

```
from sklearn.tree import DecisionTreeRegressor


def get_mae(max_leaf_nodes, predictors_train, predictors_val, targ_train, targ_val):
   model = DecisionTreeRegressor(max_leaf_nodes=max_leaf_nodes, random_state=0)
   model.fit(predictors_train, targ_train)
   preds_val = model.predict(predictors_val)
   mae = mean_absolute_error(targ_val, preds_val)
   return(mae)


for max_leaf_nodes in [5, 50, 500, 5000]:
   my_mae = get_mae(max_leaf_nodes, input_x, input_y, output_x, output_y)
   print("Max leaf nodes: %d  \t\t Mean Absolute Error:  %f" %(max_leaf_nodes, my_mae))
```

Technologies stack

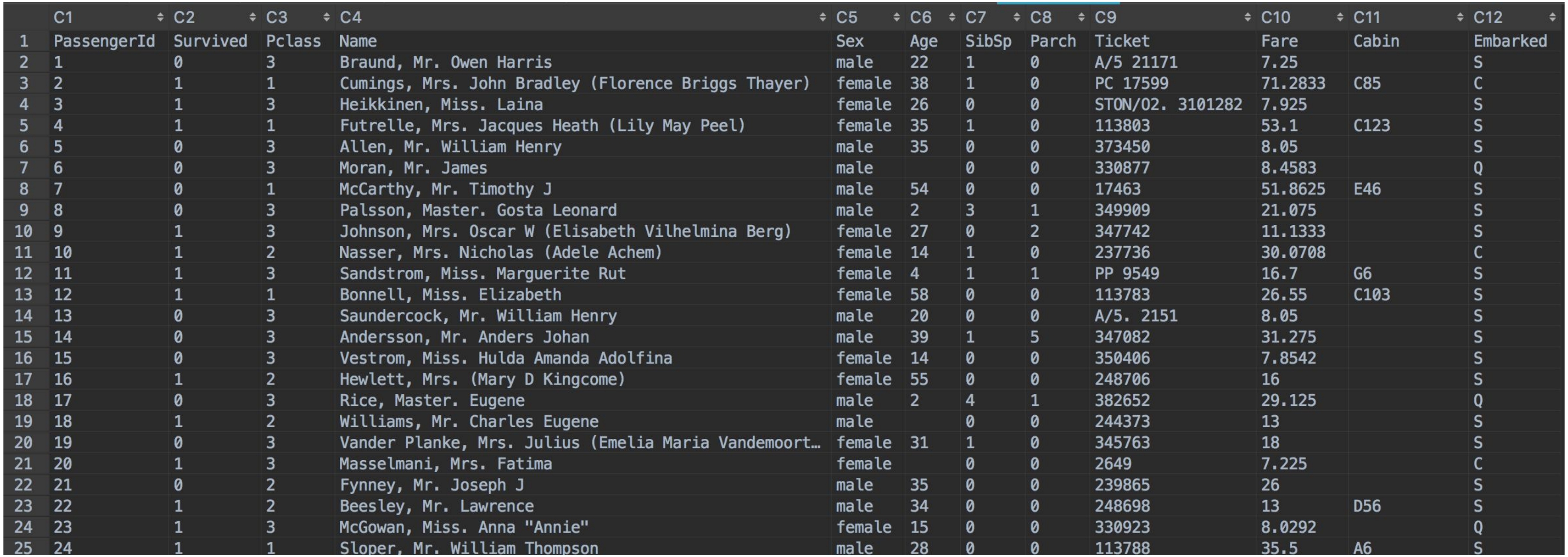
6) TensorFlow

TensorFlow™ is an open source software library for numerical computation using data flow graphs.

Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them. The flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device

Data

1) Data example



	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
1	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
2	1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25		S
3	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833	C85	C
4	3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.925		S
5	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1	C123	S
6	5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.05		S
7	6	0	3	Moran, Mr. James	male		0	0	330877	8.4583		Q
8	7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.8625	E46	S
9	8	0	3	Palsson, Master. Gosta Leonard	male	2	3	1	349909	21.075		S
10	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27	0	2	347742	11.1333		S
11	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14	1	0	237736	30.0708		C
12	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4	1	1	PP 9549	16.7	G6	S
13	12	1	1	Bonnell, Miss. Elizabeth	female	58	0	0	113783	26.55	C103	S
14	13	0	3	Saundercock, Mr. William Henry	male	20	0	0	A/5. 2151	8.05		S
15	14	0	3	Andersson, Mr. Anders Johan	male	39	1	5	347082	31.275		S
16	15	0	3	Vestrom, Miss. Hulda Amanda Adolfina	female	14	0	0	350406	7.8542		S
17	16	1	2	Hewlett, Mrs. (Mary D Kingcome)	female	55	0	0	248706	16		S
18	17	0	3	Rice, Master. Eugene	male	2	4	1	382652	29.125		Q
19	18	1	2	Williams, Mr. Charles Eugene	male		0	0	244373	13		S
20	19	0	3	Vander Planke, Mrs. Julius (Emelia Maria Vandemoort…	female	31	1	0	345763	18		S
21	20	1	3	Masselmani, Mrs. Fatima	female		0	0	2649	7.225		C
22	21	0	2	Fynney, Mr. Joseph J	male	35	0	0	239865	26		S
23	22	1	2	Beesley, Mr. Lawrence	male	34	0	0	248698	13	D56	S
24	23	1	3	McGowan, Miss. Anna "Annie"	female	15	0	0	330923	8.0292		Q
25	24	1	1	Sloper, Mr. William Thompson	male	28	0	0	113788	35.5	A6	S

Data

Classifying. We may want to classify or categorize our samples. We may also want to understand the implications or correlation of different classes with our solution goal.

Correlating. One can approach the problem based on available features within the training dataset. Which features within the dataset contribute significantly to our solution goal? Statistically speaking is there a correlation among a feature and solution goal? As the feature values change does the solution state change as well, and visa-versa? This can be tested both for numerical and categorical features in the given dataset. We may also want to determine correlation among features other than survival for subsequent goals and workflow stages. Correlating certain features may help in creating, completing, or correcting features.

Converting. For modeling stage, one needs to prepare the data. Depending on the choice of model algorithm one may require all features to be converted to numerical equivalent values. So for instance converting text categorical values to numeric values.

Data

Completing. Data preparation may also require us to estimate any missing values within a feature. Model algorithms may work best when there are no missing values.

Correcting. We may also analyze the given training dataset for errors or possibly innacurate values within features and try to correct these values or exclude the samples containing the errors. One way to do this is to detect any outliers among our samples or features. We may also completely discard a feature if it is not contribting to the analysis or may significantly skew the results.

Creating. Can we create new features based on an existing feature or a set of features, such that the new feature follows the correlation, conversion, completeness goals.

Charting. How to select the right visualization plots and charts depending on nature of the data and the solution goals.

Data

2) Analyze by describing data

Which features are available in the dataset?

Which features are categorical?

Which features are numerical?

Which features are mixed data types?

Which features may contain errors or typos?

Which features contain blank, null or empty values?

What are the data types for various features?

What is the distribution of numerical feature values across the samples?

What is the distribution of categorical features?

Data

3) Assumption based on data analysis

Correlating

Completing

Correcting

Creating

Classifying

Data

4) Analyze by visualization data (correlation data)

Correlating numerical features

Correlating numerical and ordinal features

Correlating categorical features

Correlating categorical and numerical features

Data

5) Wrangle data

Correcting by dropping features

Creating new feature extracting from existing

Converting a categorical feature

Create new feature combining existing features

Converting categorical feature to numeric

Quick completing and converting a numeric feature

Machine learning

1) Model

There are 60+ predictive modelling algorithms to choose from. We must understand the type of problem and solution requirement to narrow down to a select few models which we can evaluate. Titanic problem is a classification and regression problem. We want to identify relationship between output (Survived or not) with other variables or features (Gender, Age, Port...). We are also performing a category of machine learning which is called supervised learning as we are training our model with a given dataset.

Machine learning

2) Model examples

Decision tree learning
Association rule learning
Artificial neural networks
Deep learning
Inductive logic programming
Support vector machines
Clustering
Bayesian networks
Reinforcement learning
Representation learning
Similarity and metric learning
Sparse dictionary learning
Genetic algorithms
Rule-based machine learning
Learning classifier systems
...

Models which can solve Titanic problem:

- Logistic Regression
- KNN or k-Nearest Neighbors
- Support Vector Machines
- Naive Bayes classifier
- Decision Tree
- Random Forest
- Perceptron
- Artificial neural network
- RVM or Relevance Vector Machine

Machine learning

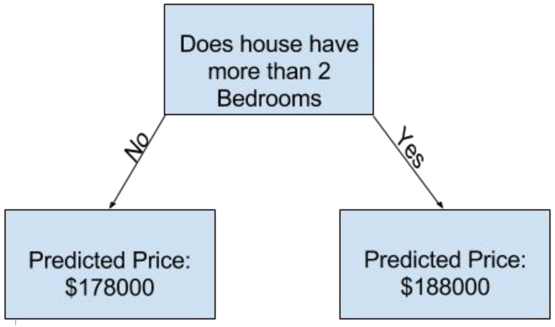
3) Model training. Decision Tree example

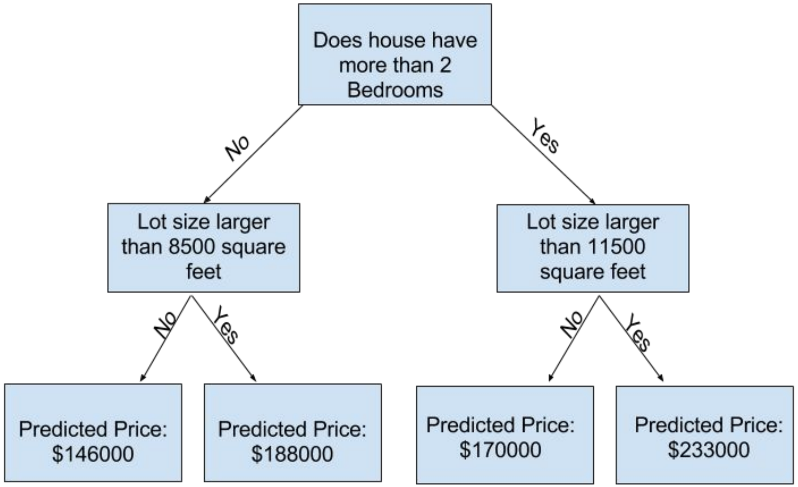
Data: **Price, Bedrooms, Square**

| target | **| predictors |**

1. Splitting data for training and testing parts
2. Train model with bigger part of data and known target
3. Test model with smaller part of data and known target

Don't test model with data which was used for training!





Does house have more than 2 Bedrooms
No
Yes
Lot size larger than 8500 square feet
Lot size larger than 11500 square feet
No
Yes
No
Yes
Predicted Price: $146000
Predicted Price: $188000
Predicted Price: $170000
Predicted Price: $233000

Machine learning

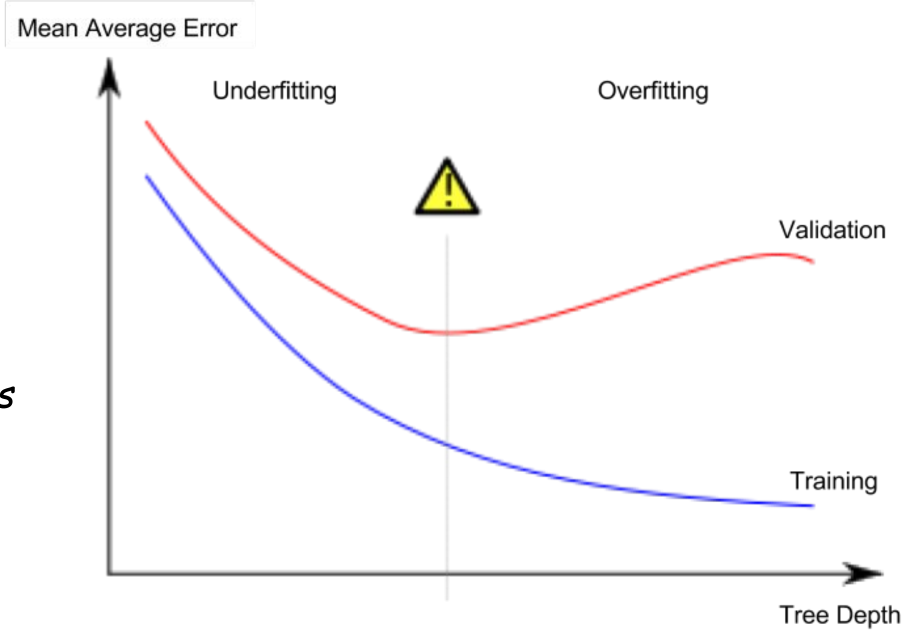
4) Model evaluation. Decision Tree example

- **Error calculation:**

  error = actual – predicted

- **Underfitting & Overfitting problem**

  Selection of better network parameters



TensorFlow

1) Short overview

`tf.Graph` - TansorFlow graph class. Consists of nodes and operations

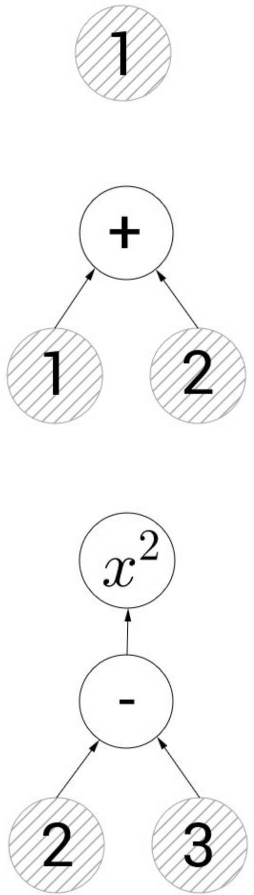
`tf.placeholder` - TensorFlow node class. Use it for inputs values

`tf.constant` - TensorFlow node class. Use it for constants values

`tf.Variable` - TensorFlow node class. Use it for variables values

`tf.add, tf.mul,` … - TensorFlow operation classes. Alternaiv of +,-,/,* etc.

`tf.Session` - TensorFlow session class. TensorFlow does calculation here



TensorFlow

2) Custom graph

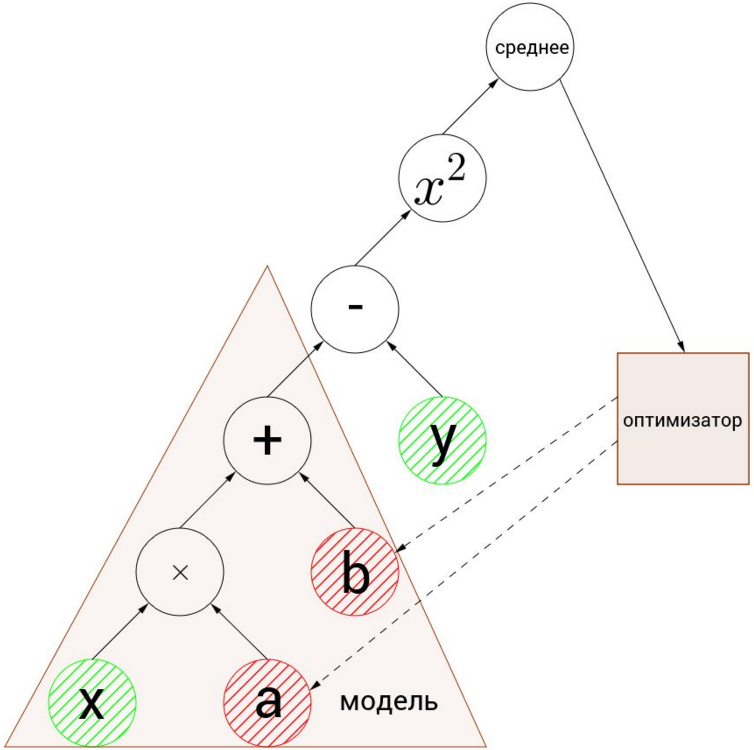
- Loss function

  cost = (calculated - etalon) ²

- Optimizer

  Function that will change graph parameters by some algorithm

  `tf.train.GradientDescentOptimizer`



Conclusions

We have reviewed:

- the most important concepts in data science
- popular technologies and tools to handle Data Science problems
- how to preprocess data
- how to do basics in Data Science with Python

References

https://en.wikipedia.org/wiki/Data_science
https://en.wikipedia.org/wiki/Data_mining
https://en.wikipedia.org/wiki/Machine_learning

Python - https://www.python.org/
NumPy - http://www.numpy.org/
Pandas - http://pandas.pydata.org/
Matplotlib - https://matplotlib.org/
Seaborn - https://seaborn.pydata.org/
Sklearn - http://scikit-learn.org/stable/
TensorFlow - https://www.tensorflow.org/

Data preprocessing overview - https://www.kaggle.com/startupsci/titanic-data-science-solutions
Data science "Get started" - https://www.kaggle.com/dansbecker/welcome-to-data-science-1
TensorFlow "Get started" - https://habrahabr.ru/post/326650/

GitHub Repo with materials for this presentation - https://github.com/FindYourGrind/kaggle_competition

Q & A

Thanks