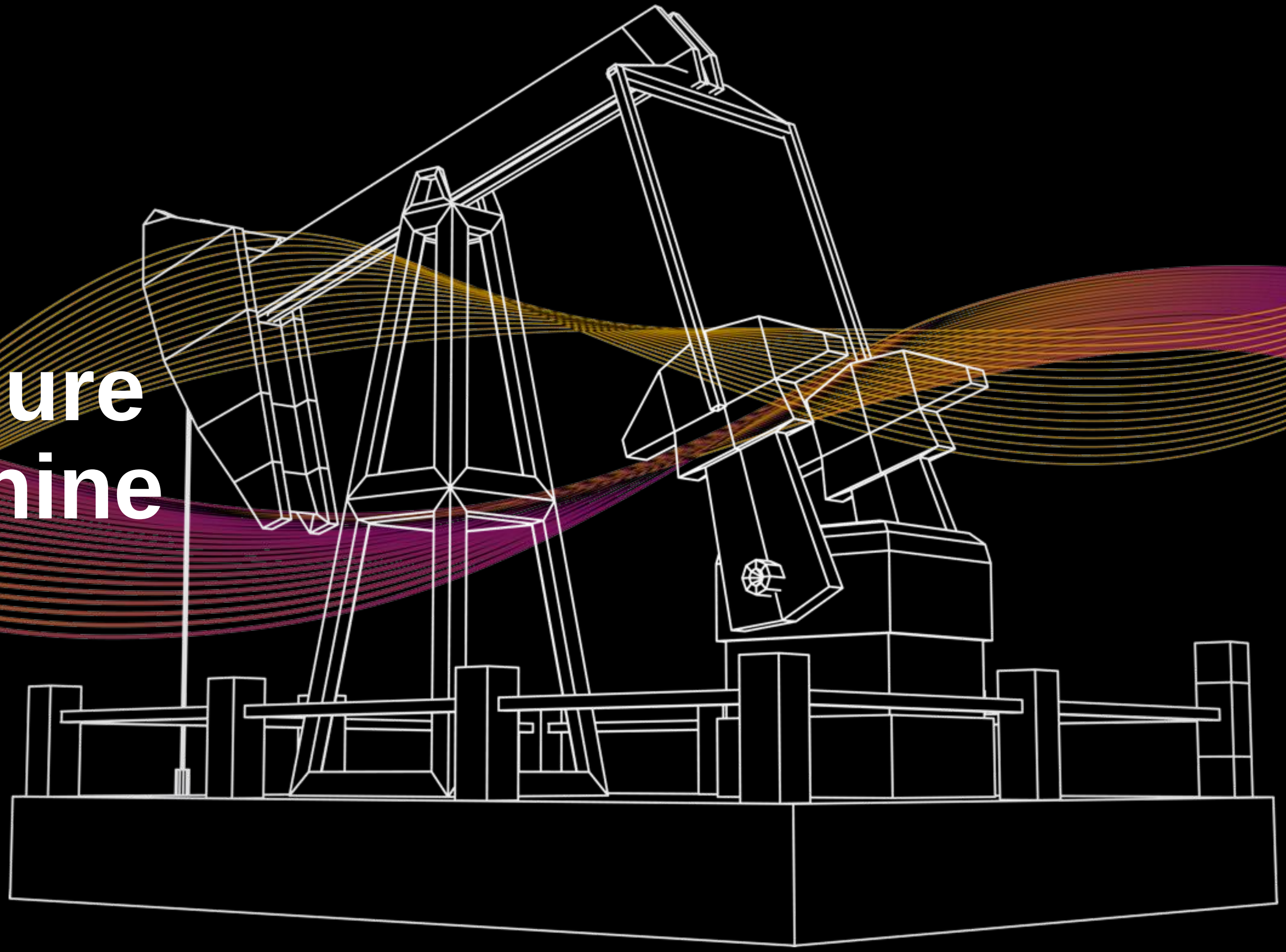
The IV International
Oil&Gas SAP Summit ONLINE

Virtual ESP intake pressure sensor: employing machine learning technology



Andrey Rzhaksinsky
Data Scientist, SAP

СУРГУТНЕФТЕГАЗ

THE BEST RUN SAP

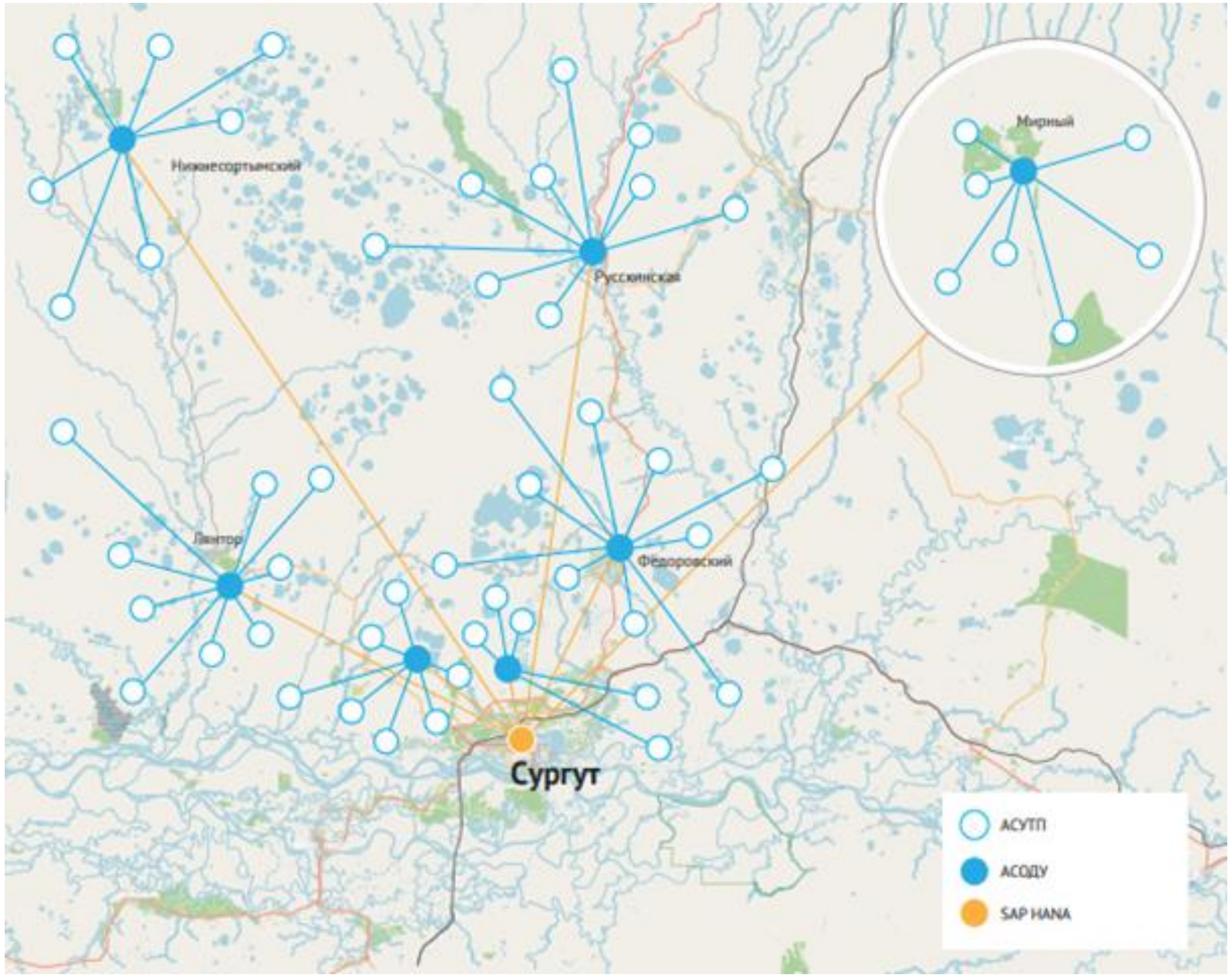
Current situation

2020 IV МЕЖДУНАРОДНЫЙ НЕФТЕГАЗОВЫЙ САММИТ SAP ONLINE

20,000+

Total number of wells

- The data is collected via a **unified communications protocol** in a uniform format and stored in the lowest level **Automated Process Control Systems** (method developed by PJSC Surgutneftegas)
- The information is consolidated in the high level **Automated Process Control Systems** (method developed by PJSC Surgutneftegas)
- The process engineer’s monitor shows **up to 1,000 wells with 6 operating parameters** for ESP based on telemetry



https://glavteh.ru/погружная-телеметрия-сургутнефтегаз

Virtual sensor for use in wells

The goal is to furnish ESPs with virtual sensors to identify abnormal well operation. The number of physical sensors is to be reduced.

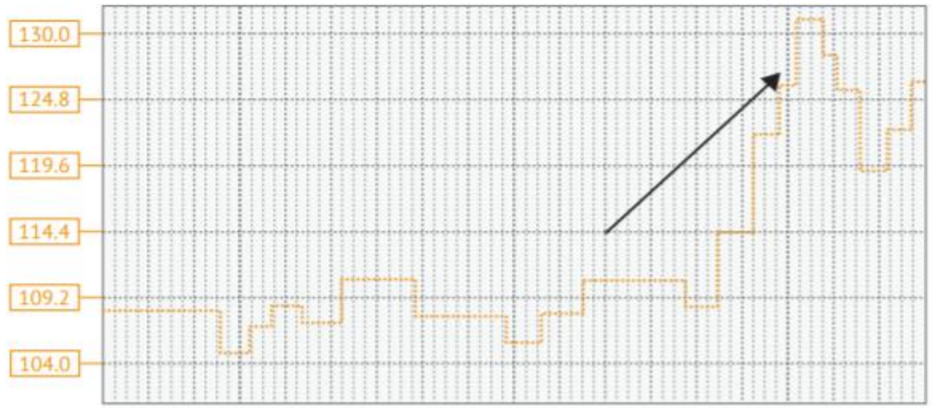
Signals: “**simultaneous pressure rise and temperature rise,**” “**simultaneous pressure rise and current strength fall,**” “**simultaneous temperature rise and current strength fall.**” Possible meaning: missing ESP, or supply flow failure;

Signal: “**lasting slow or sharp rapid pressure rise.**” Possible meaning: ESP supply flow fall, flow string not leak proof, or check valve frozen (in winter);

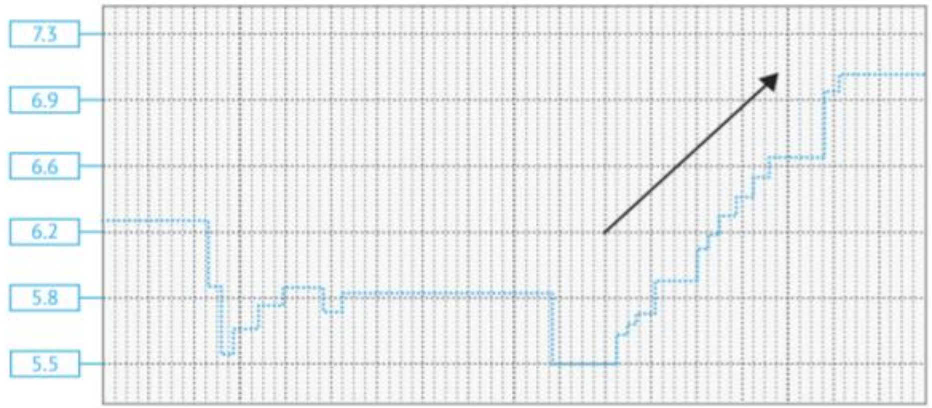
Signal: “**pressure rise after bringing a periodic well into operation.**” Possible meaning: ESP supply flow missing, or well’s flow line frozen (in winter);

Signals: “**simultaneous current rise and temperature rise,**” “**simultaneous current rise and pressure rise.**” Possible meaning: salting or blockage in ESP.

Изменение температуры, °C



Изменение давления, МПа



Data consolidation for the analysis task

IV МЕЖДУНАРОДНЫЙ НЕФТЕГАЗОВЫЙ САММИТ SAP ONLINE 2020

18,000
Wells whose data is uploaded into the analysis systems

1 year
Depth of telemetric history for analysis

5 minutes
Frequency of saving data for every parameter of every well

100
Figures collected for analysis at every moment in time, including telemetric data, operating mode and manufacturer’s data, reference data

2
Additional databases uploaded to forecast the status and probable failures: Activity database and Failure database

2.2 bln
Records collected in the database for analysis

2.1 TB
Database size when not compressed

300 GB
Data size in SAP HANA – compression ratio 1:7

SAP HANA ML – Automated Predictive Library (APL)

Machine learning capabilities built into a database

Automated modeling – a key to rapid implementation

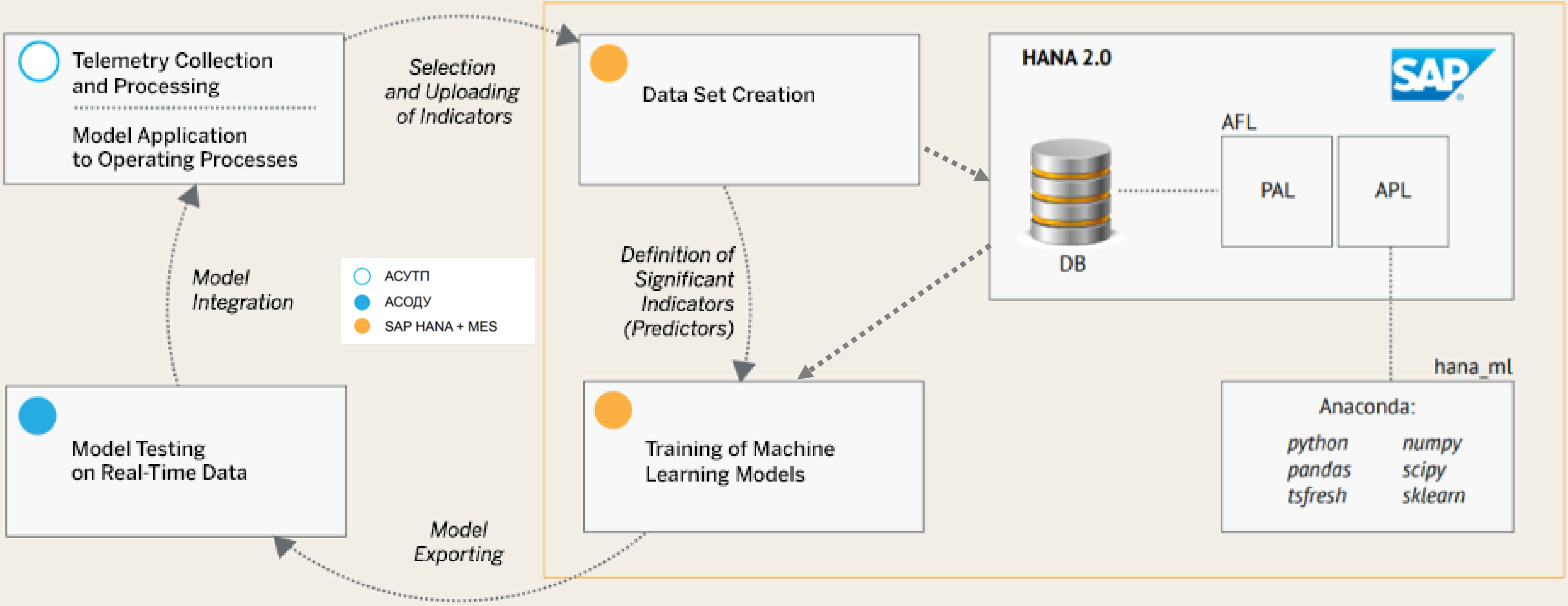
- Quick and easy to use by staff without data analysis skills; can be used in applications powered by HANA
- APL provides developers with simple procedures to create, train, use, deploy and launch predictive models

SAP HANA Platform

Automated Predictive Library (APL)

- Classification
- Regression
- Time series predictions
- Cluster analysis
- Association analysis
- Guidelines
- Nodal analysis

Data flows used in the solution



Options to transfer the outcome to Automated Process Control System

Automated Process Control System is a real time system designed to monitor well operation online and not to train the models of machine learning. Also, there is a limited capability of communications with external services as the system should not wait for a response from external systems.

To integrate a model to an Automated Process Control System, you need to upload the model as calculated factors and launch the models in the preset environment within the Automated Process Control System.

Rules generation within SAP APL

Rules generated when creating the decision tree for the intake pressure model

```
450=<T1138P6000315<1050) && (T1138P3700028=$Unknown$) => T
1050=<T1138P6000315<4350) && (T1138P4000019>=4) => T
4350=<T1138P6000315<7650) && (T1013P2500132=$Unknown$) => T
(T1138P6000315>=7650) && (T1138P600050>=0.6685) => F
(T1138P6000315<450) && (T1138P4200539<0.5) && 7.175=<T1138P2300058<8.125) => F
(T1138P6000315<450) && (T1138P4200539<0.5) && 8.125=<T1138P2300058<8.705) => F
(T1138P6000315<450) && (T1138P4200539<0.5) && 8.705=<T1138P2300058<9.315) => F
(T1138P6000315<450) && (T1138P4200539<0.5) && 9.315=<T1138P2300058<10.265) => F
(T1138P6000315<450) && (T1138P4200539<0.5) && 10.265=<T1138P2300058<11.645) => F
(T1138P6000315<450) && (T1138P4200539<0.5) && 12.395=<T1138P2300058<13.595) => F
(T1138P6000315<450) && (T1138P4200539<0.5) && 15.005=<T1138P2300058<16.9949) => T
450=<T1138P6000315<1050) && (T1138P3700028>=12.5) && (T1138P5200021<21) => T
450=<T1138P6000315<1050) && (T1138P3700028>=12.5) && (T1138P5200021=$Unknown$) => F
1050=<T1138P6000315<4350) && (T1138P4000019<4) && (T1013P2300347<8.9) => T
4350=<T1138P6000315<7650) && (T1013P2500132<981) && (T1013P300149>=1762.5) => T
4350=<T1138P6000315<7650) && (T1013P2500132>=981) && 4=<T1138P4000064<92.5) => F
(T1138P6000315>=7650) && (T1138P600050<0.034) && 12.2202=<T1013P800250<12.2944) => T
(T1138P6000315>=7650) && (T1138P600050<0.034) && 12.3664=<T1013P800250<14.3264) => F
(T1138P6000315>=7650) && (T1138P600050<0.034) && 14.3269=<T1013P800250<16.6268) => F
(T1138P6000315>=7650) && (T1138P600050<0.034) && 29.2054=<T1013P800250<29.2125) => T
(T1138P6000315>=7650) && (T1138P600050<0.034) && 39.7106=<T1013P800250<40.168) => T
(T1138P6000315<450) && (T1138P4200539<0.5) && (T1138P2300058<6.784) && (T1138P4000018>=
(T1138P6000315<450) && (T1138P4200539<0.5) && 6.784=<T1138P2300058<7.175) && (T1138P27
(T1138P6000315<450) && (T1138P4200539<0.5) && 6.784=<T1138P2300058<7.175) && 227.5=<T1
(T1138P6000315<450) && (T1138P4200539>=0.5) && (T1138P3700028>=12.5) && 27455.5=<T1010
(T1138P6000315<450) && (T1138P4200539>=0.5) && (T1138P3700028>=12.5) && 33075.5=<T1010
(T1138P6000315<450) && (T1138P4200539>=0.5) && (T1138P3700028>=12.5) && 41826.1=<T1010
```

Java/C++ code generation within SAP PAL

Source code generated when creating the decision tree for the model of the ESP underuse setpoint

```
//T1205P2300000
private static void initializeInputVariable18() {
    byte[] lInput = {84, 49, 50, 48, 53, 80, 50, 51, 48, 48, 48, 48, 48};
    mInputVariables[18] = new String(lInput, sCharset);
    byte[] lInputStorage = {110, 117, 109, 98, 101, 114};
    mInputStorageVariables[18] = new String(lInputStorage, sCharset);
}
private static void initializeOutputVariable0() {
    byte[] lOutput = {114, 114, 95, 84, 49, 49, 51, 56, 80, 52, 48, 48, 48, 48, 49, 54};
    mOutputVariables[0] = new String(lOutput, sCharset);
    byte[] lOutputStorage = {110, 117, 109, 98, 101, 114};
    mOutputStorageVariables[0] = new String(lOutputStorage, sCharset);
}

private double Kxen_RobustRegression_0_KxVar5( IKxJModelInput iInput ) {
double lValue5 = iInput.doubleValue(0);
if (iInput.isEmpty(0, mMissingStrings[0]))
{
    return (double)1.635789664455e0;
}
if ( lValue5 > 6001 ) {
    lValue5 = (double)6001;
}
else if ( lValue5 < 0 ) {
    lValue5 = (double)0;
}
if( 0 == doublesegcmp( lValue5, 0.0e0, 1, 0.0e0, 1) ) {
    return (double)1.628545779144e0;
}
if( 0 == doublesegcmp( lValue5, 0.0e0, 0, 1.0e0, 1) ) {
    return (double)(1.621301893832e0);
}
if( 0 == doublesegcmp( lValue5, 1.0e0, 0, 5.0e0, 1) ) {
    return (double)(1.614058008521e0);
}
if( 0 == doublesegcmp( lValue5, 5.0e0, 0, 1.0e1, 1) ) {
    return (double)(1.606814123209e0);
}
if( 0 == doublesegcmp( lValue5, 1.0e1, 0, 1.2e1, 1) ) {
    return (double)(1.599570237897e0);
}
if( 0 == doublesegcmp( lValue5, 1.2e1, 0, 2.0e1, 1) ) {
    return (double)(1.592326352586e0);
}
if( 0 == doublesegcmp( lValue5, 2.0e1, 0, 2.5e1, 1) ) {
    return (double)(1.585082467274e0);
}
if( 0 == doublesegcmp( lValue5, 2.5e1, 0, 3.0e1, 0) ) {
    return (double)(1.577838581963e0);
}
```

Modeling outcome and major parameters

- **Models have been built that use the dynamics of:**
  - the power factor cos phi;
  - the engine load, %
  - the current in the engine, A
  - the downtime and running time for periodic wells
  - the pressure in the measuring equipment collector
  - other indicators

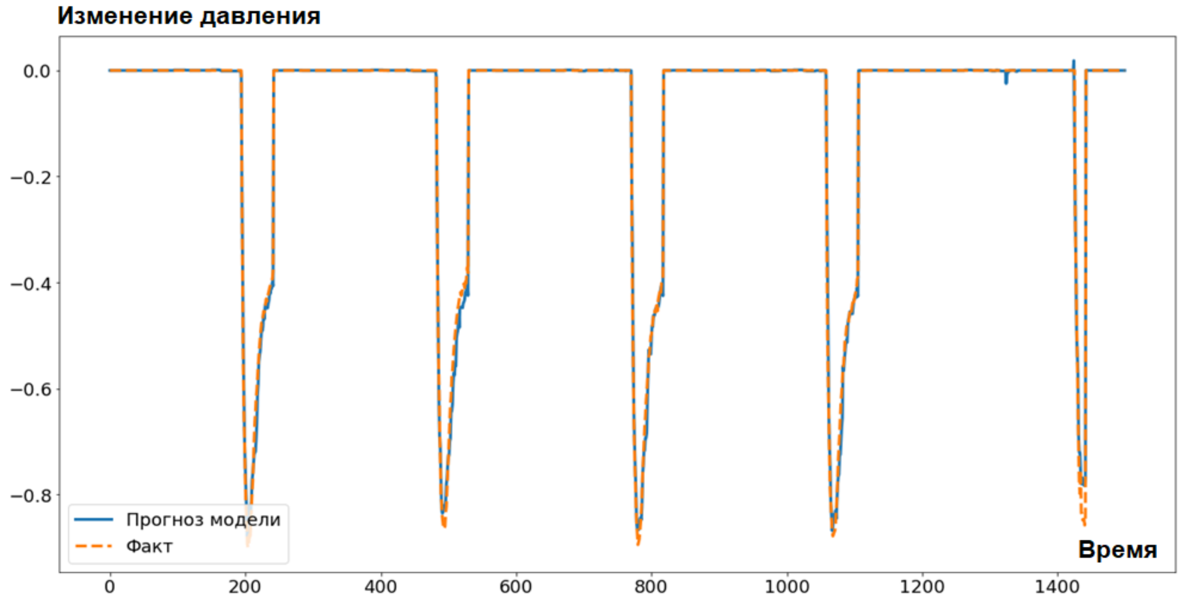
  to predict a rise in pressure and temperature in the wells without thermomanometers.

- Preparations are under way for model **testing**

- The built **telemetric data storage** и tested approaches lead to the development of new initiatives

- The predicted efficiency from implementing only these models may come to **dozens of millions in Russian rubles per year**

Example of a modeled prediction and shown dependence of intake ESP pressure change on time



Next steps

- More models for different application areas: equipment telemetry, failure prediction, finance, IT
- Using SAP Data Intelligence to manage models
- A program for developing competencies and expert support
- The training program “Managing Data Science projects” for project managers and project customers