ADS508 Data Science With Cloud Computing
Design Document Template

Authors: Ben Earnest, Andrew Kim, Gabi Rivera
Company Name: Blood Cancer Research (BCR)
Company Industry: Drug Research
Company Size: 200 Employees

Abstract

The U.S. Department of Health & Human Services has tasked our research company (BCR) to discover target proteins and develop novel drugs that disrupt pathways connected to progression of Acute Myeloid Leukemia. The information from the research will be used to support the discovery and development of new drug candidates for clinical trials of cancer therapies.

Problem Statement

Blood Cancer Research (BCR) is a non-profit institute that aims to discover novel drug targets against various hematologic cancers. Our goal is to support the effort to advance blood cancer exploratory work, with the intent to increase blood cancer survival rate through enhanced drug target repertoire. Just to give depth to the need in developing new approaches and therapies against hematological malignancies, back in 2021 it was estimated that 397,501 people in the US have leukemia or in remission (LLS, 2021). New cases of leukemia, lymphoma, and myeloma were expected to account for about 10% of an estimated 1,898,160 overall new cancer cases (LLS, 2021). With this, we propose building a machine learning algorithm that identifies Acute Myeloid Leukemia (AML) key proteins from datasets derived from OpenCell database and Beat AML program (National Cancer Institute).

Goals

The goal is to build a predictive model that can determine AML protein of interest through identification of protein-protein interactions. The Beat AML program supply the datasets related to patient attributes, AML status, gene/protein ID, inhibitory drug ID, IC50, and cell response to drug treatment. Once the AML gene/protein profile is established, then protein interaction, localization, and abundance information from OpenCell will be employed to determine strength of AML protein binding interactions. Discovered proteins that bind strongly to known AML proteins will be verified against scientific literature for historical relevance and confirmation.

Non-Goals

Blood Cancer Research is not intentionally classifying characteristics that potentially create biased representation against groups of people. Any patient information is anonymized. The creation of the ML model is only intended to identify potential proteins of interest. Identified proteins need further rigorous scientific and clinical research peer review.

Data Sources

Source data will come from the Chan Zuckerberg Biohub OpenCell database and Beat AML program derived from the Cancer Target Discovery and Development Data Portal. The tables range from 11kB to 4.6MB of datasets. The data as CSV files will be uploaded and stored in a Github repository. The data will also be stored in Amazon S3, which will involve creating and specifying a bucket within Amazon Web Services (AWS) Region before the data is uploaded to the specified bucket as objects in S3.

> *"OpenCell – proteome-scale effort to measure the localization and interactions of human proteins using high-throughput genome engineering to endogenously tag thousands of proteins in the human proteome. This dataset consists of the raw confocal fluorescence microscopy images for all tagged cell lines in the OpenCell library. These images can be interpreted both individually, to determine the localization of particular proteins of interest, and in aggregate, by training machine learning models to classify or quantify subcellular localization patterns." - Open Data on AWS*

> *"Beat AML 1.0 is a collaborative research program involving 11 academic medical centers who worked collectively to better understand drugs and drug combinations that should be prioritized for further development within clinical and/or molecular subsets of acute myeloid leukemia (AML) patients. Beat AML 1.0 provides the largest-to-date dataset on primary acute myeloid leukemia samples offering genomic, clinical, and drug response." - Open Data on AWS*

Source Link:
https://ocg.cancer.gov/programs/ctd2/data-portal
https://opencell.czbiohub.org/download

Data Exploration

Datasets are stored in S3 by manually uploading the CSV files. From there, AWS Glue Crawler was utilized to define the S3 tables into the Data Catalog. The datasets were then ingested in Athena as tables under the database specified. Partition and indexes will be created as needed to save on computational capacity. Athena together with SageMaker Studio notebook

will be used to explore initial descriptions and information about the datasets. Tools such as seaborn, matplotlib, klib, etc. will be used. Visualizations will be created to present the distribution of key attributes and discover initial trends in the data as part of preliminary data quality check for outliers and skewness. Further data quality will be imposed with the help of SageMaker Clarify to detect potential bias during data preparation and balance data appropriately.

***Github Repository Information:**
Main repository link (including final individual notebook .ipynb files):
https://github.com/ApkimCA/Data-Science-Cloud-Computing/tree/main

Data Preparation

For our first data set, BeatAML Clinical Summary, the first cleansing technique applied as part of the data scrubbing process was selecting and retaining the features that were most relevant towards our initial goal of identifying AML by creating a new dataframe with the selected features and labeling the modified data set under a new name "clsm_cut". This technique was then followed with feature transformation, where we analyzed the numerical attributes within the *%.Blasts.in.BM* and *%.Blasts.in.PB* features for unique values. In the analysis, both features contained few text values and records that were converted to whole numbers or NAN (i.e.: >50 to 51 and """"rare""""" to NAN). Through the conversion process, both features were transformed from object to float category. The remainder of the categorical attributes were converted to numerical values through integer encoding for the final model.

Another key cleansing technique applied to our first data set involved detecting any missing or null values towards each feature. This was where, through distributing a missing value plot, we found that the data set contained 52,400 missing data entries between six features (*inferred ethnicity, overall Survival, %.Blasts in BM, %.Blasts in PB, FLT3 and NPM1)*. To resolve this issue, a distribution plot using klib was generated towards each of the features containing missing data to assess the distribution of the sample data. Through each plot, key measurements (i.e.: mean, standard deviation, skewness, Kurtosis) were made, and the median measurement was selected as the best representation of the missing data values.

Compared to the first dataset, our second data set, OpenCell Protein Interaction data set, will not undergo any transformation nor cleansing techniques.

The BeatAML Clinical Summary data set features 159 attributes comprising patient information from 672 tested tumor specimens to address analysis pertaining to clinical, genomic, transcriptomic, and functional biology of AML. For the purpose of our initial goal to identify

AML, 18 attributes were picked to be bucketed and will be used for the duration of our project. The features that were picked include *lab ID, patient ID, consensus sex, inferred ethnicity, is relapse, is transformed, prior Malignancy Non Myeloid, prior MDS, prior MDSMPN, priorMPN, ELN2017, dxAtSpecimenAcquisition, vital Status, overall Survival, %.Blasts in BM, %.Blasts in PB, FLT3,* and *NPM1*. Of the fields that will be used, the feature *dxAtSpecimenAcquisition* will be labeled as our target variable because it identifies patients with AML. As a result, the remaining 141 attributes from the BeatAML Clinical Summary data set will be excluded, with a majority containing many missing records.

The OpenCell Protein Interaction data set features a total of nine attributes, all of which will be retained and used for the project. These attributes include *target gene name, interactor gene name, target ensg id, interactor ensg id, interactor uniprot ids, pval, enrichment, interaction stoichiometry,* and *abundance stoichiometry*.

As mentioned above, 18 of 159 attributes from the BeatAML Clinical Summary dataset will be bucketed. Having at least 20% blast population in the bone marrow and peripheral blood is a usual sign of having AML. On average, a healthy individual has 5% or less blast in the bone marrow while the blood does not elicit any. Another key diagnosis is mutation in *NPM1* and *FLT3-itd* genes. *NPM1* is the most common gene mutation in adult AML patients with ~30% of cases (Falini et al., 2020). While FLT3 mutation is the most common genetic alteration that is tied to poor prognosis (Kiyoi et al., 2020). Attributes such as relapse status, transformed status, non-myeloid prior malignancy status, prior MDS status, prior MDS&MPN status, and prior MPN status were included to help create a patient history background. MDS (Myelodysplastic syndrome) as well as MPN (Myeloproliferative neoplasm) are known to develop into AML later on. Patients who have MDS/MPN combined have a higher risk of progressing into AML at 23% to 54% of the time (Zhao et al., 2021). Malignancy type diagnosis, at the time of sample acquisition, is also included to help guide the model, determine the success rate and calculate important metrics. For the OpenCell Protein Interaction dataset, all nin attributes were kept. The intention is to identify proteins that show high binding and concentration against NPM1.

During Feature transformation, both *%.Blasts.in.BM* and *%.Blasts.in.PB* contain few text records that were first converted to whole numbers such as >5 to 4 percent. *%.Blasts.in.PB* had unique instances of ““rare”” and ‘rare’ which was converted to NAN as no other AML value determinants were provided. In the end, *%.Blasts.in.BM* had two values converted to whole numbers and *%.Blasts.in.PB* had three values replaced with whole numbers as well as NAN. From here, *%.Blasts.in.BM* and *%.Blasts.in.PB* data types were transformed into float from object category. All other categorical attributes will be converted to numerical values through integer encoding as needed once the final model is determined.

Data imbalance exists when the levels or instances of the target feature are not equal. This can lead to bias in a dataset, or a model that only predicts one outcome. Particularly for classification models, better performance will be seen from a balanced dataset, where there are equal proportions for the outcome (target) variable. For our effort, the initial target variable is *dxAtSpecimenAcquisition,* which identifies the presence of AML in a patient. There is significant imbalance for this variable, or instances where a patient has AML vice some other condition, or not at all. To ensure good model performance, oversampling is required to balance the outcomes prior to modeling.

For our data splitting process, we will be using a 70/30, 80/20, and 90/10 train-test splits of our data set. However, we will further break down the test portion into a 50%-50% test and validation grouping. We will then use the test portion to tune and select the best performing model before validating it on the remaining holdout values from the data set. This approach ensures we have data held out for validation that the model hasn't seen, ensuring we eliminate the possibility of validating the model on data it was trained with. Final model performance will be evaluated on the validation data set.

Data/Model Training

Taking advantage of SageMaker Jumpstart packages for quick model creation and assessment is ideal. For our purposes, we are utilizing built-in algorithms and bring-your-own containers to mainly have a comparative analysis of our model's regression performance. Ready to use pre-trained models allow us to solve AML predictive requirements without spending a lot of time on producing codes. Through this route, XGBoost algorithm will be employed to determine the relationship between selected attributes against AML diagnosis. On the other hand, bring-your-own containers allow us to fully customize all the processes throughout the regression model creation. We will also utilize SageMaker Autopilot to further expand our model selection. At the end, we will compare all the models and choose one that has the highest performance based on specified metrics.

The algorithms we will be using are the recommendation from Autopilot, logistic regression, and XGBoost. Logistic regression was chosen because of its ability to describe and predict a binary outcome from the data while assessing relationships between a dependent variable with one or more independent variables. The objective is to leverage regression to predict the likelihood of patients who are or are not diagnosed with AML. Similarly, XGBoost is ideal for its regression predictive ability using given structured relational data. Sagemaker Autopilot will also be used to establish baseline model performance at low cost and effort.

Autopilot processes the data through different algorithms utilizing automated machine learning (autoML). The user can specify the number of algorithms to consider, and Autopilot will compare regression, classification, or deep learning models to find the best one for the data and problem. In this case, we will have a selection of models that are in the range of fully customized, pre-made, and some that are selected entirely through automation.

AWS SageMaker Estimator will be implemented to manage the configuration and running of our SageMaker training job, which will comprise the XGBoost container, the role used when retrieving the IAM execution role created for the notebook instance and passed to the tuning job, instance count of 1, instance type of *ml.t3.medium* that serves as our optimal instance in the event of increasing our server towards bigger models, and output path in the form of s3 formatted to Bucket. All will then be passed to an XGBoost algorithm. The impact of having these parameters passed will enable us to develop and train an XGB fitted model.

AWS provides virtual server instances in the form of Amazon Elastic Compute Cloud (EC2). EC2 allows us to scale both the size and number of server instances based on the demand and workload (Fregly & Barth, 2021). To support our effort, we have chosen a single EC2 instance, noting that this can be scaled up or down depending on the workload, or if more data is collected and used for model training, and more predictions are required. The type of the single EC2 instance is *ml.t3.medium*, which consists of 2 vCPU's and 4 GiB of memory. This comes at a cost of 24 CPU Credits/hour. We also recognize this instance is optimized for small or medium sized databases, which may drive us to larger server instances, should our database grow.

The evaluation process of our models will be determined through a conjunction of the accuracy score and various performance metrics (i.e.: precision, recall, f1-score, PR curve, etc.) from our models. Due to the imbalances that were noted within the data, the accuracy score alone cannot be dependent to both evaluate the performance of our models or provide an accurate prediction of whether patients are diagnosed or not diagnosed with AML. The use of various performance metrics enables the model evaluation process to diagnose and assess the accuracy performances of the models through an in-depth perspective while noting the results that were predicted accurately or not accurately. These metrics will enable us to evaluate the performance of our models, the correct or incorrect predictions that were generated by each model, and any assessments to both improve the performance of the models and relegate towards the SageMaker's XGBoost model.

Measuring Impact

We will measure success on this project by developing a model that can identify the AML protein of interest with an accuracy of at least .70. We will also target identifying the strongest AML protein interactions with an accuracy of .70 (min). It is our estimate that meeting these

model performance thresholds will increase the AML survival rate by at least 10%. This comes from better identification of AML, and identifying the best treatment options based on the predicted strongest AML protein interactions.

Security Checklist, Privacy and Other Risks

- The data that will be used will not store or process PHI, PII, user behavior tracking, and credit card data since we remain to keep the information involving patients anonymized.

- This application will write to a local, private S3 bucket when the code is run. There will not be a public S3 bucket for the data.

- Confirmation bias by way of an overload of data is a top bias considered and addressed to avoid the risk of too much data wrangling, which will distort the data to a point it says what we want it to. This in turn would affect our decision-making process, expectations, predictions, and approach towards the outcome. A key approach to avoid this form of bias is to assess the data through various viewpoints and analyses. False causality bias is another form of bias considered with the risk of associating correlations as causal factors for treatment responses. The best approach to avoid this form of bias is to analyze and evaluate every variable if they are dependent or independent from each other and what factors within variables make a correlation possible.

- An ethical concern with the data that should be addressed is that there is an imbalance for race and age in the data. This could lead to a bias in the model, and predictions that could negatively impact people based on race or age. This will need to be addressed in the data cleaning phase, before we move to splitting the data and building the model.

Future Enhancements:

Predictive regression models identifying AML diagnosis were successfully created. This is the first part of the goal to support efforts in advancing blood cancer exploratory work from available clinical data. A follow-up extension involving classification algorithms needs to be implemented to fulfill discovery of key protein-protein interactions related to AML diagnosis while determining key AML protein of interest. Future projects will involve further selection and leveraging the best models to determine the strength of AML protein binding interactions. Other potential features that support relevant protein interaction can also be explored by looking into protein localization at specific cell cycle as well as abundance or concentration at specific time.

Enhancement #1: Modify Feature Selection and Feature Engineering

Changes to both feature selection and feature engineering could improve model performance. Much of our research effort has involved developing enough domain knowledge, but it is understood by the data analytics team that there are opportunities to improve model performance through better feature selection. Through work with domain experts, we will be able to engineer new features, and potentially reduce the feature space to prevent model overfit. This could also reduce training time and save server cost.

Enhancement #2: Verify Data Leakages

In addition, better domain knowledge and feature selection will help verify data leakage is not occurring, which is when some of our features occur after the target variable. This is a common pitfall in model training. Ensuring these features are removed from the training data will support improved performance. An example of this was creating the 'AML_detected" variable, then removing "dxAtAquisition". The "AML_detected" variable was a target variable created from the "dxAtAquisition" variable. Had we not removed it from the training data, we would have had a problem with data leakage, and our trained model performance would have been much better than expected, but would not have performed well on validation data or new data. We could also look at scaling and normalizing variables where appropriate.

Enhancement #3: Standardizing/Normalizing data

Standardizing and normalizing the data, especially towards features that generate skewed distributions, would provide an effective structure for creating and maintaining the overall quality of the data while ensuring an equal balance of the data being analyzed and implemented towards a modeling process. The use of both techniques would also ensure an improvement of model performance. Among our features that generated a skewed distribution through visualizations was our target variable "dxatspecimenacquisition", which was skewed to the right. This indicates the presence of many outliers within the feature, which we chose not to remove to ensure the effectiveness and quality of the data.

Enhancement #4: Additional Time

Data preparation was one of the most time-consuming feats that took a lot of effort to accomplish. About 50% of the clinical summary dataset was missing. Only 20 features were selected to be relevant out of 158 total. There were two phases of data transformation and feature creation, once before data exploration and another before integrating the data into the models. Leveraging data wrangler for data preparation would have been a great resource if given additional time and access. Freeing up time to focus more on model creation and the later part of the pipeline would have been beneficial in ensuring there is enough time dedicated to fulfill other projected goals in identifying AML as well as determining highly associated proteins.

References

Falini, B., Brunetti, L., Sportoletti, P., & Martelli, M. P. (2020, October 8). *NPM1-mutated acute myeloid leukemia: from bench to bedside*. Blood. Retrieved March 27, 2023, from https://ashpublications.org/blood/article/136/15/1707/461241/NPM1-mutated-acute-myeloid-leukemia-from-bench-to

Fregly, C., & Barth, A. (2021). *Data science on AWS*. O'Reilly.

Kiyoi, H., Kawashima, N., & Ishikawa, Y. (2020). *FLT3 mutations in acute myeloid leukemia: Therapeutic paradigm beyond inhibitor development*. Cancer science, 111(2), 312–322. https://doi.org/10.1111/cas.14274

The Leukemia & Lymphoma Society. (2021, April). *Facts and statistics overview*. LLS. Retrieved March 13, 2023, from https://www.lls.org/facts-and-statistics/facts-and-statistics-overview

Zhao, M., Sun, J., Liu, S., Fan, H., Fu, Y., Tan, Y., & Gao, S. (2021). *Development of a myelodysplastic/myeloproliferative neoplasm-unclassifiable in a patient with acute myeloid leukemia: a case report and literature review. The Journal of international medical research, 49(5), 3000605211018426. https://doi.org/10.1177/03000605211018426*

Appendix: Identification of Acute Myeloid Leukemia through ML XGBoost Model (SageMaker Jumpstart) approach

In [3]:
```
import warnings
warnings.filterwarnings('ignore')
```

Dataset Sources

Beat Acute Myeloid Leukemia (AML) 1.0 was accessed on 13Mar2023 from https://registry.opendata.aws/beataml. OHSU BeatAML Datasets Link: https://ctd2-data.nci.nih.gov/Public/OHSU-1/BeatAML_Waves1_2/

OpenCell Datasets Link: https://opencell.czbiohub.org/download

Check Pre-Requisites from the `01_setup/` Folder

In [4]:
```
%store -r setup_instance_check_passed
```

In [5]:
```
try:
    setup_instance_check_passed
except NameError:
    print("++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Instance Check.")
    print("++++++++++++++++++++++++++++++++")
```

In [6]:
```
print(setup_instance_check_passed)
```

```
True
```

In [7]:
```
%store -r setup_dependencies_passed
```

In [8]:
```
try:
    setup_dependencies_passed
except NameError:
    print("++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Setup Dependencies.")
    print("++++++++++++++++++++++++++++++++")
```

In [9]:
```
print(setup_dependencies_passed)
```

```
True
```

In [10]:
```
%store -r setup_s3_bucket_passed
```

In [11]:
```
try:
    setup_s3_bucket_passed
except NameError:
    print("++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Setup S3 Bucket.")
    print("++++++++++++++++++++++++++++++++")
```

In [12]:
```
print(setup_s3_bucket_passed)
```

```
True
```

In [13]:
```
%store -r setup_iam_roles_passed
```

In [14]:
```
try:
    setup_iam_roles_passed
except NameError:
    print("++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Setup IAM Roles.")
    print("++++++++++++++++++++++++++++++++")
```

In [15]:
```
print(setup_iam_roles_passed)
```

```
True
```

In [16]:
```
if not setup_instance_check_passed:
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Instance Check.")
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
if not setup_dependencies_passed:
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Setup Dependencies.")
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
if not setup_s3_bucket_passed:
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Setup S3 Bucket.")
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
if not setup_iam_roles_passed:
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Setup IAM Roles.")
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
```

In [17]:
```
import boto3
import sagemaker
import pandas as pd

sess = sagemaker.Session()
bucket = sess.default_bucket()
role = sagemaker.get_execution_role()
region = boto3.Session().region_name
account_id = boto3.client("sts").get_caller_identity().get("Account")

sm = boto3.Session().client(service_name="sagemaker", region_name=region)
```

S3 Original Dataset Location

Importing Raw Datasets from AWS S3. Use the AWS Command Line Interface (CLI) to list the S3 bucket content using the following CLI commands:

In [18]:
```
!aws s3 ls s3://team4rawdatasets/CSV/Input/OHSU_BeatAML_ClinicalSummary/
```

```
2023-03-21 02:37:21          0
2023-03-27 02:40:07     714614 OHSU_BeatAMLWaves1_2_Tyner_ClinicalSummary.txt
```

In [19]:
```
!aws s3 ls s3://team4rawdatasets/CSV/Input/OpenCell_ProteinInteraction/
```

```
2023-03-21 02:37:38          0
2023-03-21 02:38:40    4568928 opencell-protein-interactions.csv
```

Set S3 Source Location

In [20]:
```
#BeatAML Clinical Summary
s3_public_path_clsm = "s3://team4rawdatasets/CSV/Input/OHSU_BeatAML_ClinicalSummary/"
```

In [21]:
```
%store s3_public_path_clsm
```

```
Stored 's3_public_path_clsm' (str)
```

In [22]:
```
print(s3_public_path_clsm)
```

```
s3://team4rawdatasets/CSV/Input/OHSU_BeatAML_ClinicalSummary/
```

In [23]:
```
!aws s3 ls $s3_public_path_clsm
```

```
2023-03-21 02:37:21          0
2023-03-27 02:40:07     714614 OHSU_BeatAMLWaves1_2_Tyner_ClinicalSummary.txt
```

In [24]:
```
#BeatAML OpenCell Protein Interaction
s3_public_path_pi = "s3://team4rawdatasets/CSV/Input/OpenCell_ProteinInteraction/"
```

In [25]:
```
%store s3_public_path_pi
```

```
Stored 's3_public_path_pi' (str)
```

In [26]:
```
print(s3_public_path_pi)
```

```
s3://team4rawdatasets/CSV/Input/OpenCell_ProteinInteraction/
```

In [27]:
```
!aws s3 ls $s3_public_path_pi
```

```
2023-03-21 02:37:38          0
2023-03-21 02:38:40    4568928 opencell-protein-interactions.csv
```

In [28]:
```
from IPython.core.display import display, HTML

display(
    HTML(
        '<b>Review <a target="blank" href="https://s3.console.aws.amazon.com/s3/buckets/team4rawdatasets?prefix=CSV/I
            region, account_id, region
        )
    )
)
```

Review S3 Bucket

Athena

Athena Database

PyAthena is a Python DB API 2.0 (PEP 249) compliant client for Amazon Athena.

In [29]:
```
!pip install --disable-pip-version-check -q PyAthena==2.1.0
from pyathena import connect
```

```
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system p
ackage manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
```

In [30]:
```
ingest_create_athena_db_passed = False
```

In [31]:
```
database_name = "bcr"
```

In [32]:
```
# Set S3 staging directory -- this is a temporary directory used for Athena queries
s3_staging_dir = "s3://{0}/athena/staging".format(bucket)
```

In [33]:
```
conn = connect(region_name=region, s3_staging_dir=s3_staging_dir)
```

In [34]:
```
statement0 = "CREATE DATABASE IF NOT EXISTS {}".format(database_name)
print(statement0)
```

```
CREATE DATABASE IF NOT EXISTS bcr
```

In [35]:
```
pd.read_sql(statement0, conn)
```

Out[35]:

Verify The Database Has Been Created Succesfully

In [36]:
```
statement00 = "SHOW DATABASES"

df_show = pd.read_sql(statement00, conn)
df_show.head(5)
```

Out[36]:

	database_name
0	bcr
1	default
2	dsoaws
3	sagemaker_featurestore

In [37]:
```
if database_name in df_show.values:
    ingest_create_athena_db_passed = True
```

In [38]:
```
%store ingest_create_athena_db_passed
```

```
Stored 'ingest_create_athena_db_passed' (bool)
```

Athena Table Created Through AWS Glue Crawler

In [39]:
```
from IPython.core.display import display, HTML

display(
    HTML(
        '<b>Review <a target="top" href="https://us-east-1.console.aws.amazon.com/glue/home?region=us-east-1#/v2/data
            region
        )
    )
)
```

Review AWS Glue Catalog

Athena Sample Query

In [40]:
```
# Set Athena database & table
table_clsm = "ohsu_beataml_clinicalsummary"
table_pi = "opencell_proteininteraction"
```

In [41]:
```
#Athena SQL Code
statement1 = """
SELECT *
FROM {}.{}
""".format(
    database_name, table_pi
)

print(statement1)
```

```
SELECT *
FROM bcr.opencell_proteininteraction
```

In [42]:
```
pi = pd.read_sql(statement1, conn)
pi.head(5)
```

Out[42]:

	target_gene_name	interactor_gene_name	target_ensg_id	interactor_ensg_id	interactor_uniprot_ids	
0	CAPZB	LIN7C	ENSG00000077549	ENSG00000148943	Q9NUP9;G3V1D4	4
1	CAPZB	LMO7	ENSG00000077549	ENSG00000136153	Q8WWI1-3;Q8WWI1;Q8WWI1-2;Q8WWI1-4;J3KP06;F8WD2...	7
2	CAPZB	LONP1	ENSG00000077549	ENSG00000196365	K7EJE8;K7EKE6;P36776-3;P36776-2;P36776;K7ER27	4
3	CAPZB	LRCH2	ENSG00000077549	ENSG00000130224	Q5VUJ6-2;Q5VUJ6	8
4	CAPZB	LRPPRC	ENSG00000077549	ENSG00000138095	P42704;C9JCA9;B8ZZ38;A0A0C4DG06;H7C3W8	3

In [43]:
```
if not pi.empty:
    print("[OK]")
else:
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
    print("[ERROR] YOUR DATA HAS NOT BEEN CONVERTED TO PARQUET. LOOK IN PREVIOUS CELLS TO FIND THE ISSUE.")
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
```

```
[OK]
```

Data cleaning

Import Tools:

In [44]:
```
!pip install klib
```

```
Requirement already satisfied: klib in /opt/conda/lib/python3.7/site-packages (1.0.1)
Requirement already satisfied: pandas<2.0.0,>=1.1.2 in /opt/conda/lib/python3.7/site-packages (from klib) (1.3.5)
Requirement already satisfied: scipy<2.0.0,>=1.1.0 in /opt/conda/lib/python3.7/site-packages (from klib) (1.4.1)
Requirement already satisfied: numpy<2.0.0,>=1.16.3 in /opt/conda/lib/python3.7/site-packages (from klib) (1.21.6)
Requirement already satisfied: seaborn<0.12.0,>=0.11.1 in /opt/conda/lib/python3.7/site-packages (from klib) (0.11.
2)
Requirement already satisfied: Jinja2<4.0.0,>=3.0.3 in /opt/conda/lib/python3.7/site-packages (from klib) (3.1.2)
Requirement already satisfied: matplotlib<4.0.0,>=3.0.3 in /opt/conda/lib/python3.7/site-packages (from klib) (3.1.
3)
Requirement already satisfied: MarkupSafe>=2.0 in /opt/conda/lib/python3.7/site-packages (from Jinja2<4.0.0,>=3.0.3-
>klib) (2.1.2)
Requirement already satisfied: python-dateutil>=2.1 in /opt/conda/lib/python3.7/site-packages (from matplotlib<4.0.
0,>=3.0.3->klib) (2.8.2)
Requirement already satisfied: kiwisolver>=1.0.1 in /opt/conda/lib/python3.7/site-packages (from matplotlib<4.0.0,>=
3.0.3->klib) (1.1.0)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.7/site-packages (from matplotlib<4.0.0,>=3.0.3
->klib) (0.10.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /opt/conda/lib/python3.7/site-packages (f
rom matplotlib<4.0.0,>=3.0.3->klib) (2.4.6)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/lib/python3.7/site-packages (from pandas<2.0.0,>=1.1.2->kl
ib) (2019.3)
Requirement already satisfied: six in /opt/conda/lib/python3.7/site-packages (from cycler>=0.10->matplotlib<4.0.0,>=
3.0.3->klib) (1.14.0)
Requirement already satisfied: setuptools in /opt/conda/lib/python3.7/site-packages (from kiwisolver>=1.0.1->matplot
lib<4.0.0,>=3.0.3->klib) (59.3.0)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system p
ackage manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv

[notice] A new release of pip is available: 23.0.1 -> 23.1
[notice] To update, run: pip install --upgrade pip
```

In [45]:
```
import numpy as np
import seaborn as sns
import klib
import matplotlib.pyplot as plt

%matplotlib inline
%config InlineBackend.figure_format='retina'
```

BeatAML Clinical Summary

OHSU BeatAML Clinical Summary Table

In [46]:
```
# SQL statement
statement2 = """
SELECT *
FROM {}.{}
""".format(
    database_name, table_clsm
)

print(statement2)
```

```
SELECT *
FROM bcr.ohsu_beataml_clinicalsummary
```

In [47]:
```
clsm = pd.read_sql(statement2, conn)
clsm.head(5)
```

Out[47]:

	labid	patientid	consensus_sex	inferred_sex	inferred_ethnicity	centerid	cebpa_biallelic	ageatdiagnosis	isrelapse	isdenovo	...	st
0	09-00705	163	Male	Male	White	1	n	73.0	False	True	...	
1	10-00136	174	Male	Male	White	1	n	69.0	False	True	...	
2	10-00172	175	Female	Male	White	1	n	59.0	False	True	...	
3	10-00507	45	Female	Female	White	1	n	70.0	False	True	...	
4	10-00542	174	Male	Male	White	1	n	69.0	True	False	...	

5 rows × 159 columns

In [48]:
```
clsm = clsm.replace('', np.NAN)
clsm.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 672 entries, 0 to 671
Columns: 159 entries, labid to zrsr2
dtypes: bool(9), float64(22), int64(7), object(121)
memory usage: 793.5+ KB
```

In [49]:
```
clsm.info(2)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 672 entries, 0 to 671
Data columns (total 159 columns):
 #    Column                                          Dtype
---   ------                                          -----
 0    labid                                           object
 1    patientid                                       int64
 2    consensus_sex                                   object
 3    inferred_sex                                    object
 4    inferred_ethnicity                              object
 5    centerid                                        int64
 6    cebpa_biallelic                                 object
 7    ageatdiagnosis                                  float64
 8    isrelapse                                       bool
 9    isdenovo                                        bool
 10   istransformed                                   bool
 11   finalfusion                                     object
 12   specificdxatacquisition_mdsmpn                  bool
 13   nonaml_mdsmpn_specificdxatacquisition           bool
 14   priormalignancynonmyeloid                       object
 15   priormalignancytype                             object
 16   cumulativechemo                                 object
 17   priormalignancyradiationtx                      object
 18   priormds                                        object
 19   priormdsmorethantwomths                         object
 20   priormdsmpn                                     object
 21   priormdsmpnmorethantwomths                      object
 22   priormpn                                        object
 23   priormpnmorethantwomths                         object
 24   dxatinclusion                                   object
 25   specificdxatinclusion                           object
 26   eln2017                                         object
 27   eln2008                                         object
 28   dxatspecimenacquisition                         object
 29   specificdxatacquisition                         object
 30   ageatspecimenacquisition                        float64
 31   timeofsamplecollectionrelativetoinclusion       int64
 32   specimengroups                                  object
 33   specimentype                                    object
 34   rnaseq                                          object
 35   exomeseq                                        object
 36   totaldrug                                       object
 37   rnaseqanalysis                                  object
 38   analysisexomeseq                                object
```

39	analysisdrug	object
40	cumulativetreatmenttypecount	int64
41	cumulativetreatmenttypes	object
42	cumulativetreatmentregimencount	int64
43	cumulativetreatmentregimens	object
44	cumulativetreatmentstagecount	int64
45	cumulativetreatmentstages	object
46	responsetoinductiontx	object
47	typeinductiontx	object
48	responsedurationtoinductiontx	float64
49	mostrecenttreatmenttype	object
50	currentregimen	object
51	currentstage	object
52	mostrecenttreatmentduration	float64
53	vitalstatus	object
54	overallsurvival	float64
55	causeofdeath	object
56	any_different_labs	bool
57	any_different_labs_also_beataml	bool
58	different_lab_ids	object
59	different_id_karyotype_interval	int64
60	%.basophils.in.pb	float64
61	%.blasts.in.bm	object
62	%.blasts.in.pb	object
63	%.eosinophils.in.pb	float64
64	%.immature.granulocytes.in.pb	float64
65	%.lymphocytes.in.pb	float64
66	%.monocytes.in.pb	float64
67	%.neutrophils.in.pb	float64
68	%.nucleated.rbcs.in.pb	float64
69	alt	object
70	ast	float64
71	albumin	float64
72	creatinine	float64
73	fab/blast.morphology	object
74	hematocrit	float64
75	hemoglobin	float64
76	karyotype	object
77	ldh	float64
78	mcv	float64
79	other.cytogenetics	object
80	platelet.count	float64
81	surface.antigens.(immunohistochemical.stains)	object
82	total.protein	float64

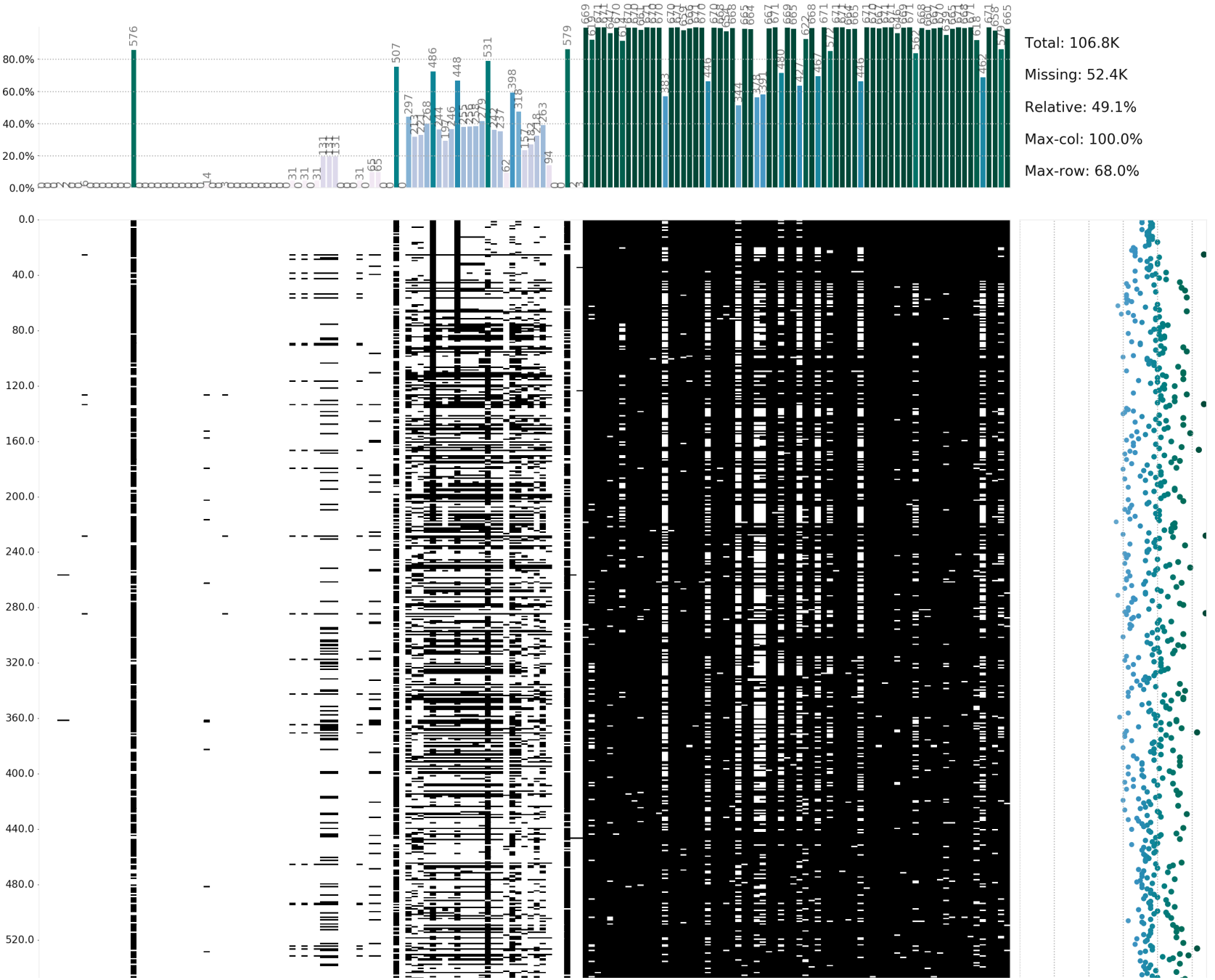
83	wbc.count	float64
84	any_different_cgs	bool
85	any_different_cgs_also_beataml	bool
86	different_cgs_lab_ids	object
87	flt3-itd	object
88	npm1	object
89	abl1	object
90	asxl1	object
91	asxl2	object
92	atm	object
93	bcor	object
94	bcorl1	object
95	braf	object
96	brca2	object
97	calr	object
98	cbl	object
99	ccnd2	object
100	ccnd3	object
101	cd36	object
102	cebpa	object
103	chek2	object
104	ciita	object
105	crebbp	object
106	csf3r	object
107	ctcf	object
108	cux1	object
109	dnmt3a	object
110	ep300	object
111	etv6	object
112	ezh2	object
113	fbxw7	object
114	flt3	object
115	gata1	object
116	gata2	object
117	idh1	object
118	idh2	object
119	ikzf1	object
120	jak1	object
121	jak2	object
122	jak3	object
123	kdm6a	object
124	kit	object
125	kmt2a	object
126	kmt2d	object

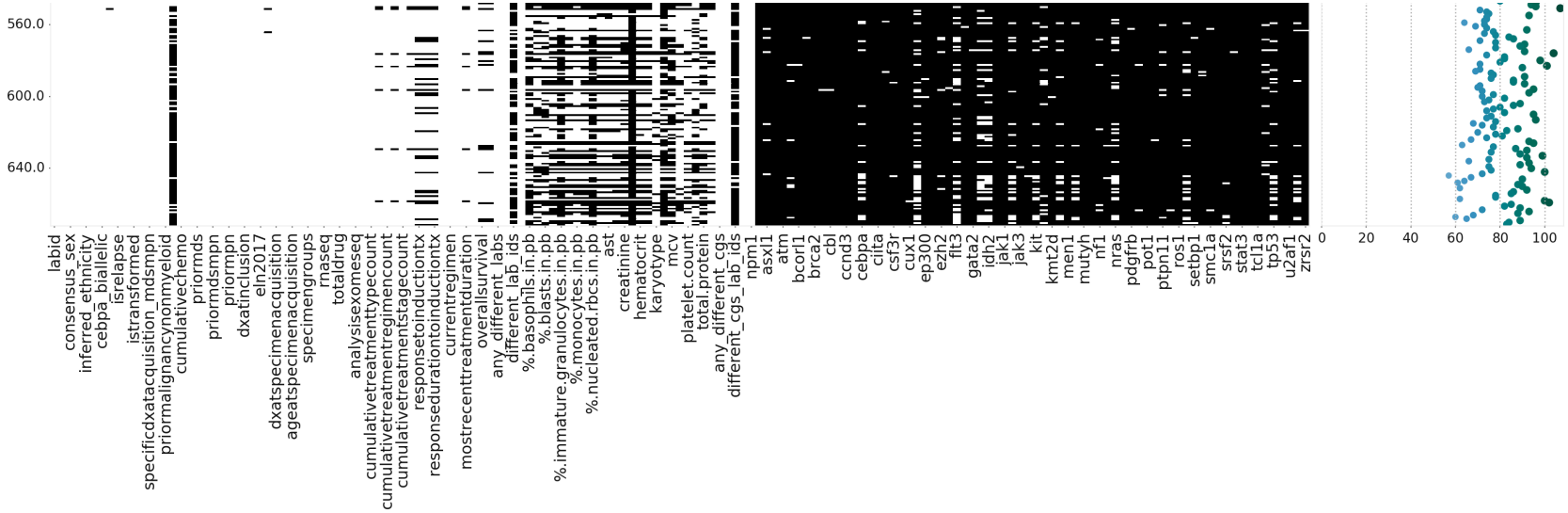
```
 127  kras                                            object
 128  men1                                            object
 129  mpl                                             object
 130  mutyh                                           object
 131  myd88                                           object
 132  nf1                                             object
 133  notch1                                          object
 134  nras                                            object
 135  pax5                                            object
 136  pdgfrb                                          object
 137  phf6                                            object
 138  pot1                                            object
 139  prdm1                                           object
 140  ptpn11                                          object
 141  rad21                                           object
 142  ros1                                            object
 143  runx1                                           object
 144  setbp1                                          object
 145  sf3b1                                           object
 146  smc1a                                           object
 147  socs1                                           object
 148  srsf2                                           object
 149  stag2                                           object
 150  stat3                                           object
 151  suz12                                           object
 152  tcl1a                                           object
 153  tet2                                            object
 154  tp53                                            object
 155  tyk2                                            object
 156  u2af1                                           object
 157  wt1                                             object
 158  zrsr2                                           object
dtypes: bool(9), float64(22), int64(7), object(121)
memory usage: 793.5+ KB
```

In [50]:
```
klib.missingval_plot(clsm)
```

Out[50]:
```
GridSpec(6, 6)
```

Missing value plot





Select Relevant Features

In [51]:
```
clsm_cut = pd.DataFrame(clsm[['labid', 'patientid', 'consensus_sex', 'inferred_ethnicity', 'isrelapse',
                              'istransformed', 'priormalignancynonmyeloid', 'priormds', 'priormdsmpn', 'priormpn',
                              'eln2017', 'dxatspecimenacquisition', 'vitalstatus', 'overallsurvival', '%.blasts.in.bm
                               '%.blasts.in.pb','flt3-itd', 'npm1']])

clsm_cut
```

Out[51]:

	labid	patientid	consensus_sex	inferred_ethnicity	isrelapse	istransformed	priormalignancynonmyeloid	priormds	priormdsmpn
0	09-00705	163	Male	White	False	False	n	n	n
1	10-00136	174	Male	White	False	False	n	n	n
2	10-00172	175	Female	White	False	False	n	n	n
3	10-00507	45	Female	White	False	False	n	n	n
4	10-00542	174	Male	White	True	False	n	n	n
...	...	...	...	...	...	...	...	...	...
667	17-00072	4366	Male	White	False	True	n	n	n
668	17-00077	4317	Female	White	False	False	n	n	n
669	17-00093	4379	Female	Black	False	True	n	n	n
670	17-00094	4380	Male	White	False	True	n	n	n
671	17-00096	2747	Male	White	False	True	n	n	y

672 rows × 18 columns

In [52]:
```
clsm_cut.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 672 entries, 0 to 671
Data columns (total 18 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   labid                       672 non-null    object
 1   patientid                   672 non-null    int64
 2   consensus_sex               672 non-null    object
 3   inferred_ethnicity          670 non-null    object
 4   isrelapse                   672 non-null    bool
 5   istransformed               672 non-null    bool
 6   priormalignancynonmyeloid   672 non-null    object
 7   priormds                    672 non-null    object
 8   priormdsmpn                 672 non-null    object
 9   priormpn                    672 non-null    object
 10  eln2017                     672 non-null    object
 11  dxatspecimenacquisition     672 non-null    object
 12  vitalstatus                 672 non-null    object
 13  overallsurvival             607 non-null    float64
 14  %.blasts.in.bm              459 non-null    object
 15  %.blasts.in.pb              451 non-null    object
 16  flt3-itd                    670 non-null    object
 17  npm1                        669 non-null    object
dtypes: bool(2), float64(1), int64(1), object(14)
memory usage: 85.4+ KB
```

In [53]:
```
clsm_cut.describe()
```

Out[53]:

	patientid	overallsurvival
count	672.000000	607.000000
mean	2088.020833	441.881384
std	973.372734	479.180429
min	17.000000	-1.000000
25%	1450.750000	167.000000
50%	2016.000000	323.000000
75%	2501.500000	555.000000
max	4380.000000	5305.000000

Attribute Tranformation

% Blasts Attributes Numerical Prep

%.blasts.in.bm Attribute:

In [54]:
```
clsm_cut['%.blasts.in.bm'].unique()
```

Out[54]:
```
array(['94', '80', '91', '97', '87', nan, '40', '75', '83', '95', '85',
       '90', '70', '92', '72', '68', '88', '36', '81', '93', '34', '77.5',
       '46', '65', '50', '76', '71', '60', '73', '55', '0.5', '30', '62',
       '18', '82', '28', '41', '64', '84', '21', '51', '17', '49.4', '32',
       '29', '25', '59.3', '66', '20', '52', '54', '22', '10', '12',
       '46.0', '13', '67', '39', '25.9', '45', '37', '78', '8', '3',
       '54.8', '74', '96', '4', '86.1', '42', '56', '69', '79', '33', '9',
       '.4', '51.5', '15', '5', '24', '7', '2', '6', '1', '58', '>50',
       '35', '86', '32.0', '93.2', '0', '27', '89.6', '23', '98', '19',
       '91.8', '>95', '57', '71.5', '78.3', '63', '1.5', '53.74', '59.5',
       '44', '42.5', '26', '3.5', '48', '26.3', '47', '88.5'],
      dtype=object)
```

In [55]:
```
# > and < will be changed to whole numbers less than or greater than.
clsm_cut['%.blasts.in.bm'] = clsm_cut['%.blasts.in.bm'].replace(['>50'], 51)
clsm_cut['%.blasts.in.bm'] = clsm_cut['%.blasts.in.bm'].replace(['>95'], 96)

clsm_cut['%.blasts.in.bm'].unique()
```

Out[55]:
```
array(['94', '80', '91', '97', '87', nan, '40', '75', '83', '95', '85',
       '90', '70', '92', '72', '68', '88', '36', '81', '93', '34', '77.5',
       '46', '65', '50', '76', '71', '60', '73', '55', '0.5', '30', '62',
       '18', '82', '28', '41', '64', '84', '21', '51', '17', '49.4', '32',
       '29', '25', '59.3', '66', '20', '52', '54', '22', '10', '12',
       '46.0', '13', '67', '39', '25.9', '45', '37', '78', '8', '3',
       '54.8', '74', '96', '4', '86.1', '42', '56', '69', '79', '33', '9',
       '.4', '51.5', '15', '5', '24', '7', '2', '6', '1', '58', 51, '35',
       '86', '32.0', '93.2', '0', '27', '89.6', '23', '98', '19', '91.8',
       96, '57', '71.5', '78.3', '63', '1.5', '53.74', '59.5', '44',
       '42.5', '26', '3.5', '48', '26.3', '47', '88.5'], dtype=object)
```

%.blasts.in.pb Attribute:

In [56]:
```
clsm_cut['%.blasts.in.pb'].unique()
```

Out[56]:
```
array(['97', '19', '99', '80', nan, '51', '30', '41', '84', '77', '75',
       '63', '60', '96', '66', '45', '93', '9', '82', '15', '33', '0',
       '13', '94', '89', '83', '>90', '78', '72', '59', '32', '6', '29',
       '24', '64', '57', '52', '2.1', '<5', '17', '22', '5', '47', '56',
       '25', '23', '42', '65', '71', '8', '3.5', '66.3', '95', '44', '10',
       '28.6', '18', '58', '67', '40', '92', '54', '1.0', '2', '20', '28',
       '35', '85', '1', '42.4', '16', '49.1', '14', '88', '46', '7',
       '0.5', '79', '26', '87', '20.4', '68', '48', '5.3', '61', '90',
       '17.4', '57.4', '43.8', '50', '37', '4', '3', '12', '81', '11',
       '90.5', '"""rare"""', '90.2', '55', '12.0', 'rare', '39', '31.0',
       '86', '47.4', '27.4', '39.6', '83.0', '12.9', '5.0', '15.4', '9.5',
       '62', '64.6', '27.8', '69.14', '52.2', '91', '67.25', '49', '23.7',
       '48.6', '98', '74.8', '2.6', '43', '29.6', '47.5', '38', '2.5',
       '25.2', '3.56', '70', '99.2', '73', '26.7', '38.5', '7.7', '74',
       '93.3', '12.1', '11.2', '92.9', '98.4', '6.8', '10.5', '53', '3.1',
       '28.9', '72.9', '40.2', '31', '3.3', '42.1', '11.5', '77.8', '3.8',
       '59.5', '21.7', '53.2'], dtype=object)
```

In [57]:
```
#%.Blasts.in.PB attribute has 1 "rare" record with no flt3 nor npm1 input. This will be changed to NAN
clsm_cut['%.blasts.in.pb'] = clsm_cut['%.blasts.in.pb'].replace(['"""rare"""'], np.nan)
clsm_cut['%.blasts.in.pb'] = clsm_cut['%.blasts.in.pb'].replace(['rare'], np.nan)
# > and < will be changed to whole numbers less than or greater than.
clsm_cut['%.blasts.in.pb'] = clsm_cut['%.blasts.in.pb'].replace(['<5'], 4)
clsm_cut['%.blasts.in.pb'] = clsm_cut['%.blasts.in.pb'].replace(['>90'], 91)

clsm_cut['%.blasts.in.pb'].unique()
```

Out[57]:
```
array(['97', '19', '99', '80', nan, '51', '30', '41', '84', '77', '75',
       '63', '60', '96', '66', '45', '93', '9', '82', '15', '33', '0',
       '13', '94', '89', '83', 91, '78', '72', '59', '32', '6', '29',
       '24', '64', '57', '52', '2.1', 4, '17', '22', '5', '47', '56',
       '25', '23', '42', '65', '71', '8', '3.5', '66.3', '95', '44', '10',
       '28.6', '18', '58', '67', '40', '92', '54', '1.0', '2', '20', '28',
       '35', '85', '1', '42.4', '16', '49.1', '14', '88', '46', '7',
       '0.5', '79', '26', '87', '20.4', '68', '48', '5.3', '61', '90',
       '17.4', '57.4', '43.8', '50', '37', '4', '3', '12', '81', '11',
       '90.5', '90.2', '55', '12.0', '39', '31.0', '86', '47.4', '27.4',
       '39.6', '83.0', '12.9', '5.0', '15.4', '9.5', '62', '64.6', '27.8',
       '69.14', '52.2', '91', '67.25', '49', '23.7', '48.6', '98', '74.8',
       '2.6', '43', '29.6', '47.5', '38', '2.5', '25.2', '3.56', '70',
       '99.2', '73', '26.7', '38.5', '7.7', '74', '93.3', '12.1', '11.2',
       '92.9', '98.4', '6.8', '10.5', '53', '3.1', '28.9', '72.9', '40.2',
       '31', '3.3', '42.1', '11.5', '77.8', '3.8', '59.5', '21.7', '53.2'],
      dtype=object)
```

From Categorical to Numerical

Transform %.blasts.in.bm and %.blasts.in.pb from object to float:

In [58]:
```
clsm_cut['%.blasts.in.bm'] = clsm_cut['%.blasts.in.bm'].astype(float)
clsm_cut['%.blasts.in.pb'] = clsm_cut['%.blasts.in.pb'].astype(float)


clsm_cut.info()
```

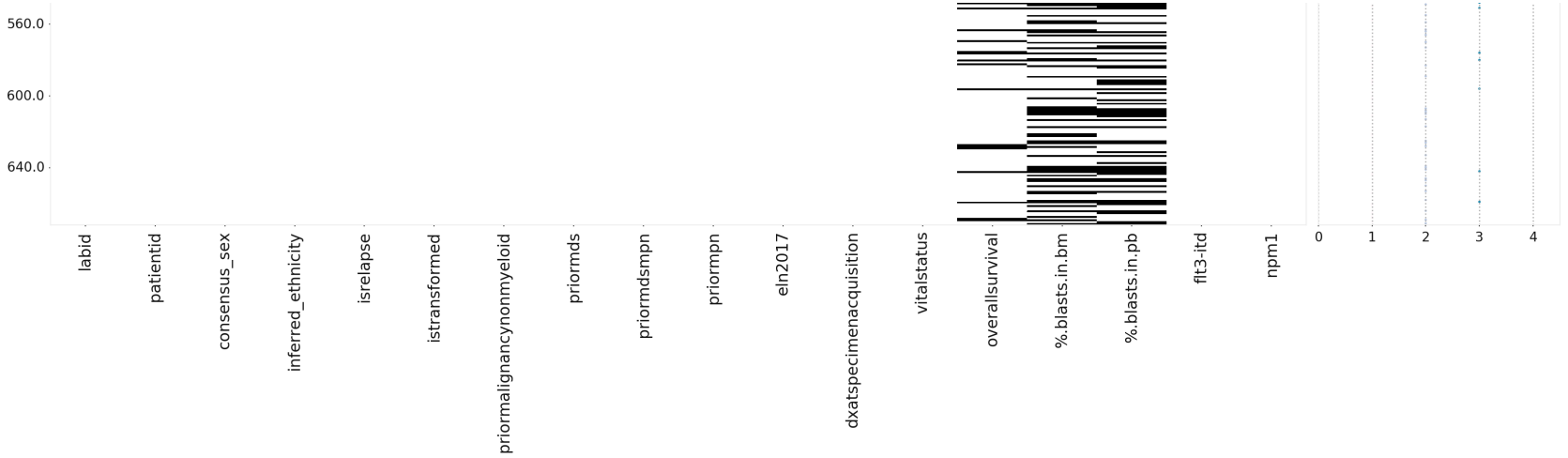
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 672 entries, 0 to 671
Data columns (total 18 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   labid                       672 non-null    object
 1   patientid                   672 non-null    int64
 2   consensus_sex               672 non-null    object
 3   inferred_ethnicity          670 non-null    object
 4   isrelapse                   672 non-null    bool
 5   istransformed               672 non-null    bool
 6   priormalignancynonmyeloid   672 non-null    object
 7   priormds                    672 non-null    object
 8   priormdsmpn                 672 non-null    object
 9   priormpn                    672 non-null    object
 10  eln2017                     672 non-null    object
 11  dxatspecimenacquisition     672 non-null    object
 12  vitalstatus                 672 non-null    object
 13  overallsurvival             607 non-null    float64
 14  %.blasts.in.bm              459 non-null    float64
 15  %.blasts.in.pb              448 non-null    float64
 16  flt3-itd                    670 non-null    object
 17  npm1                        669 non-null    object
dtypes: bool(2), float64(3), int64(1), object(12)
memory usage: 85.4+ KB
```

clsm_cut Identify Missing Values

In [59]:
```
klib.missingval_plot(clsm_cut)
```

Out[59]:
```
GridSpec(6, 6)
```

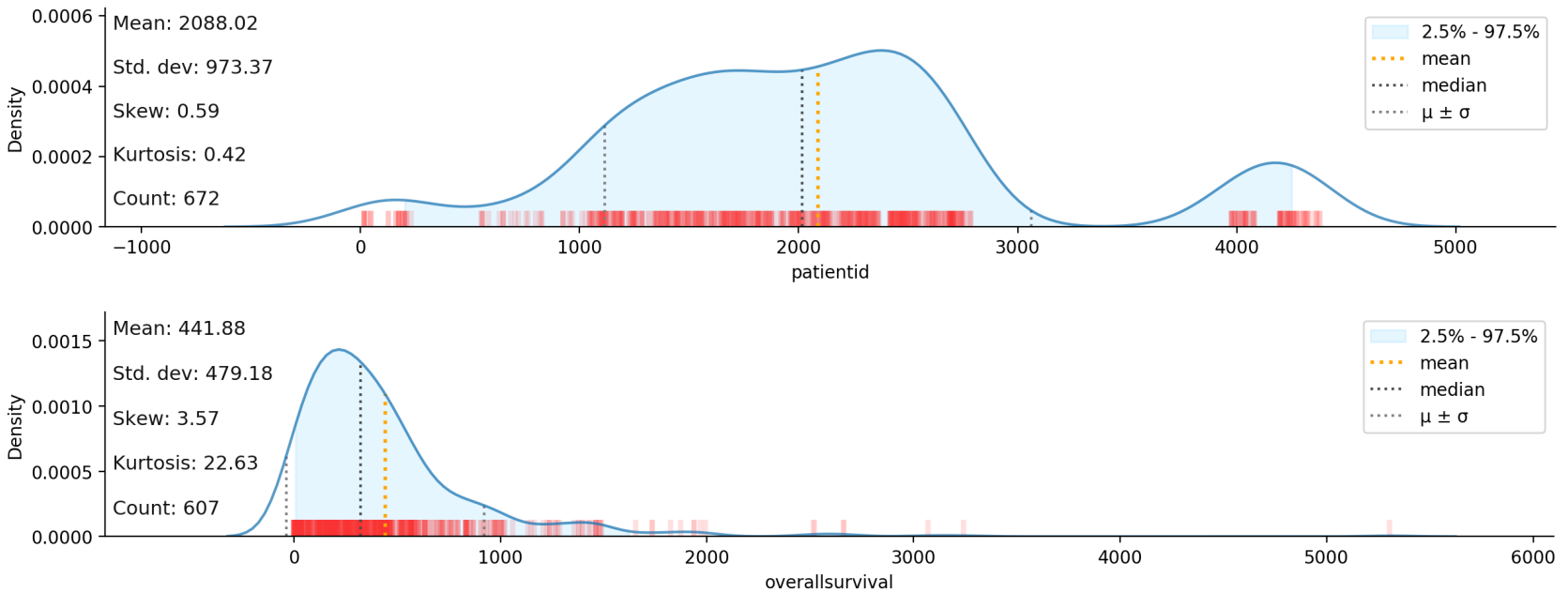
Missing value plot

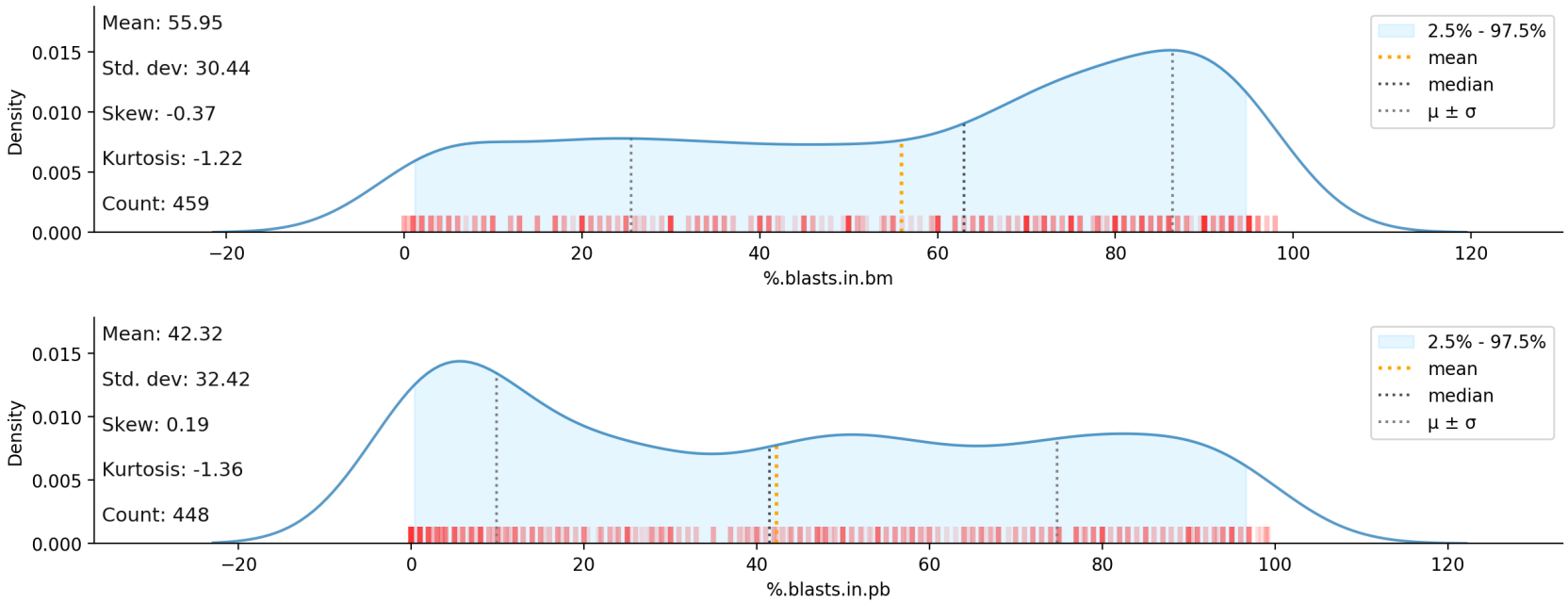


Replace Missing Values

In [60]:
```
klib.dist_plot(clsm_cut)
```

Out[60]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fce464e0750>
```





In [61]:
```
clsm_cut.describe()
```

Out[61]:

	patientid	overallsurvival	%.blasts.in.bm	%.blasts.in.pb
count	672.000000	607.000000	459.000000	448.000000
mean	2088.020833	441.881384	55.949325	42.316629
std	973.372734	479.180429	30.440925	32.418249
min	17.000000	-1.000000	0.000000	0.000000
25%	1450.750000	167.000000	30.000000	10.000000
50%	2016.000000	323.000000	63.000000	41.500000
75%	2501.500000	555.000000	83.000000	72.000000
max	4380.000000	5305.000000	98.000000	99.200000

In [62]:
```
#From distibution, skewness suggest median is the best representation.
clsm_cut['overallsurvival'] = clsm_cut['overallsurvival'].fillna(clsm_cut['overallsurvival'].median())
clsm_cut['%.blasts.in.bm'] = clsm_cut['%.blasts.in.bm'].fillna(clsm_cut['%.blasts.in.bm'].median())
clsm_cut['%.blasts.in.pb'] = clsm_cut['%.blasts.in.pb'].fillna(clsm_cut['%.blasts.in.pb'].median())
```

In [63]:
```
#Replace categorical NaN with unknown
clsm_cut = clsm_cut.replace(np.nan, 'unknown', regex=True)
```

In [64]:
```
#Determine mode of inferred_ethnicity:
clsm_cut['inferred_ethnicity'].mode()
```

Out[64]:
```
0    White
dtype: object
```

In [65]:
```
#In inferred_ethnicity, replace mode of unknown to white:
clsm_cut['inferred_ethnicity'] = clsm_cut['inferred_ethnicity'].replace(['unknown'], 'white')

clsm_cut['inferred_ethnicity'].unique()
```

Out[65]:
```
array(['White', 'HispNative', 'AdmixedBlack', 'Asian', 'Black',
       'AdmixedAsian', 'white', 'AdmixedWhite', 'AdmixedHispNative'],
      dtype=object)
```

In [66]:
```
#Determine mode of flt3-itd:
clsm_cut['flt3-itd'].mode()
```

Out[66]:
```
0    negative
dtype: object
```

In [67]:
```
#In flt3-itd, replace mode of unknown to negative:
clsm_cut['flt3-itd'] = clsm_cut['flt3-itd'].replace(['unknown'], 'negative')

clsm_cut['flt3-itd'].unique()
```

Out[67]:
```
array(['positive', 'negative'], dtype=object)
```

In [68]:
```
#Determine mode of npm1:
clsm_cut['npm1'].mode()
```

Out[68]:
```
0    negative
dtype: object
```

In [69]:
```
#In npm1, replace mode of unknown to negative:
clsm_cut['npm1'] = clsm_cut['npm1'].replace(['unknown'], 'negative')

clsm_cut['npm1'].unique()
```

Out[69]:
```
array(['positive', 'negative'], dtype=object)
```

In [70]:
```
klib.missingval_plot(clsm_cut)
```

```
No missing values found in the dataset.
```

In [71]:
```
clsm_cut.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 672 entries, 0 to 671
Data columns (total 18 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   labid                      672 non-null    object
 1   patientid                  672 non-null    int64
 2   consensus_sex              672 non-null    object
 3   inferred_ethnicity         672 non-null    object
 4   isrelapse                  672 non-null    bool
 5   istransformed              672 non-null    bool
 6   priormalignancynonmyeloid  672 non-null    object
 7   priormds                   672 non-null    object
 8   priormdsmpn                672 non-null    object
 9   priormpn                   672 non-null    object
 10  eln2017                    672 non-null    object
 11  dxatspecimenacquisition    672 non-null    object
 12  vitalstatus                672 non-null    object
 13  overallsurvival            672 non-null    float64
 14  %.blasts.in.bm             672 non-null    float64
 15  %.blasts.in.pb             672 non-null    float64
 16  flt3-itd                   672 non-null    object
 17  npm1                       672 non-null    object
dtypes: bool(2), float64(3), int64(1), object(12)
memory usage: 85.4+ KB
```

Check for Duplicates

In [72]:
```
clsm_cut = clsm_cut.drop_duplicates(ignore_index=True)
clsm_cut.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 672 entries, 0 to 671
Data columns (total 18 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   labid                       672 non-null    object
 1   patientid                   672 non-null    int64
 2   consensus_sex               672 non-null    object
 3   inferred_ethnicity          672 non-null    object
 4   isrelapse                   672 non-null    bool
 5   istransformed               672 non-null    bool
 6   priormalignancynonmyeloid   672 non-null    object
 7   priormds                    672 non-null    object
 8   priormdsmpn                 672 non-null    object
 9   priormpn                    672 non-null    object
 10  eln2017                     672 non-null    object
 11  dxatspecimenacquisition     672 non-null    object
 12  vitalstatus                 672 non-null    object
 13  overallsurvival             672 non-null    float64
 14  %.blasts.in.bm              672 non-null    float64
 15  %.blasts.in.pb              672 non-null    float64
 16  flt3-itd                    672 non-null    object
 17  npm1                        672 non-null    object
dtypes: bool(2), float64(3), int64(1), object(12)
memory usage: 85.4+ KB
```

Create Target Variable

In [73]:
```
clsm_cut['dxatspecimenacquisition'].value_counts()
```

Out[73]:
```
ACUTE MYELOID LEUKAEMIA (AML) AND RELATED PRECURSOR NEOPLASMS    646
MYELODYSPLASTIC SYNDROMES                                         15
MYELODYSPLASTIC/MYELOPROLIFERATIVE NEOPLASMS                       4
ACUTE LEUKAEMIAS OF AMBIGUOUS LINEAGE                              3
MYELOPROLIFERATIVE NEOPLASMS                                       3
MATURE B-CELL NEOPLASMS                                            1
Name: dxatspecimenacquisition, dtype: int64
```

In [74]:
```
#create column for AML detected
clsm_cut['AML_detected'] = ['yes' if x == 'ACUTE MYELOID LEUKAEMIA (AML) AND RELATED PRECURSOR NEOPLASMS'
                            else 'no' for x in clsm_cut['dxatspecimenacquisition']]
```

In [75]:
```
clsm_cut.head()
```

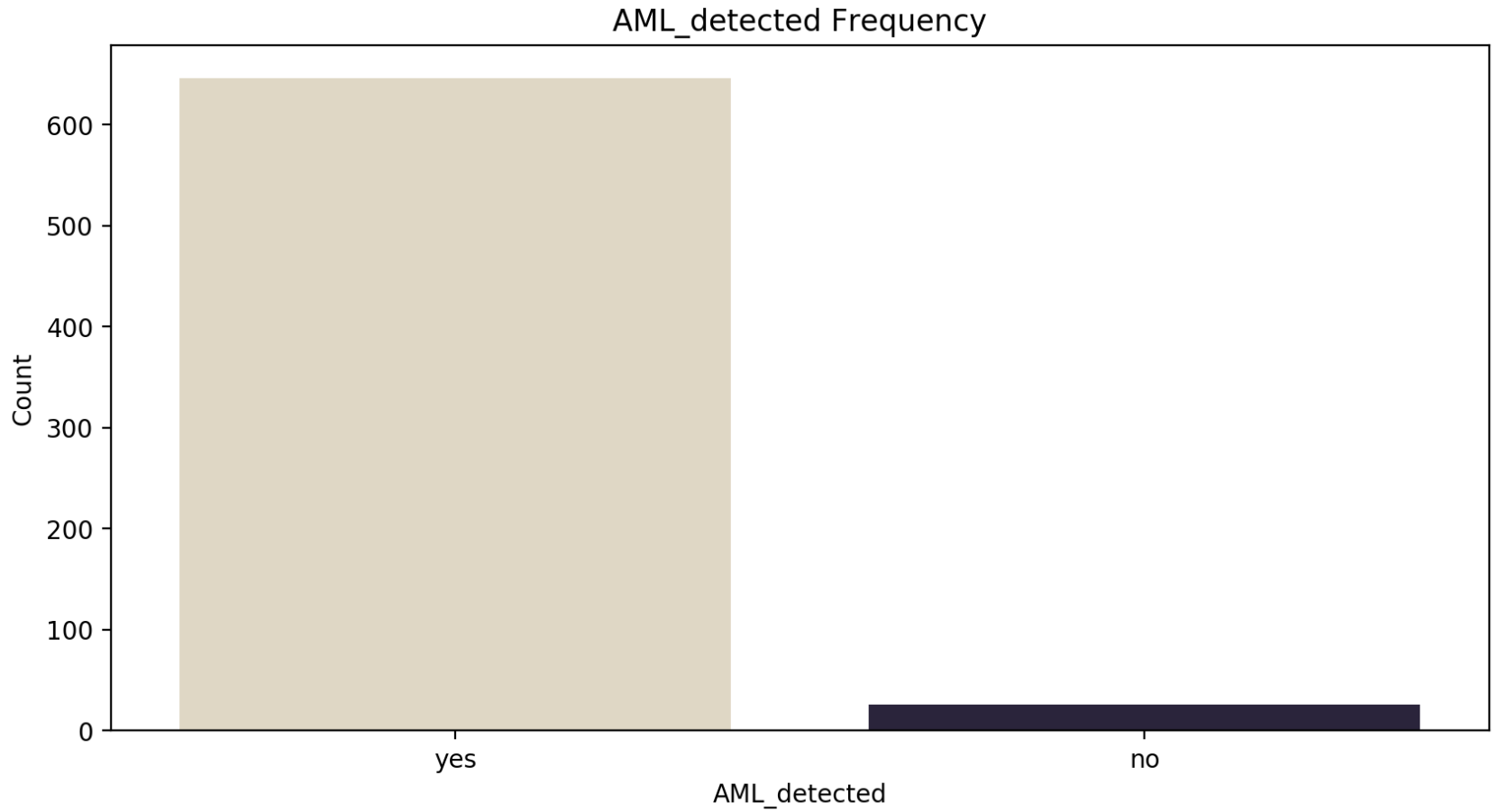
Out[75]:

	labid	patientid	consensus_sex	inferred_ethnicity	isrelapse	istransformed	priormalignancynonmyeloid	priormds	priormdsmpn	p
0	09-00705	163	Male	White	False	False	n	n	n	
1	10-00136	174	Male	White	False	False	n	n	n	
2	10-00172	175	Female	White	False	False	n	n	n	
3	10-00507	45	Female	White	False	False	n	n	n	
4	10-00542	174	Male	White	True	False	n	n	n	

Data Exploration

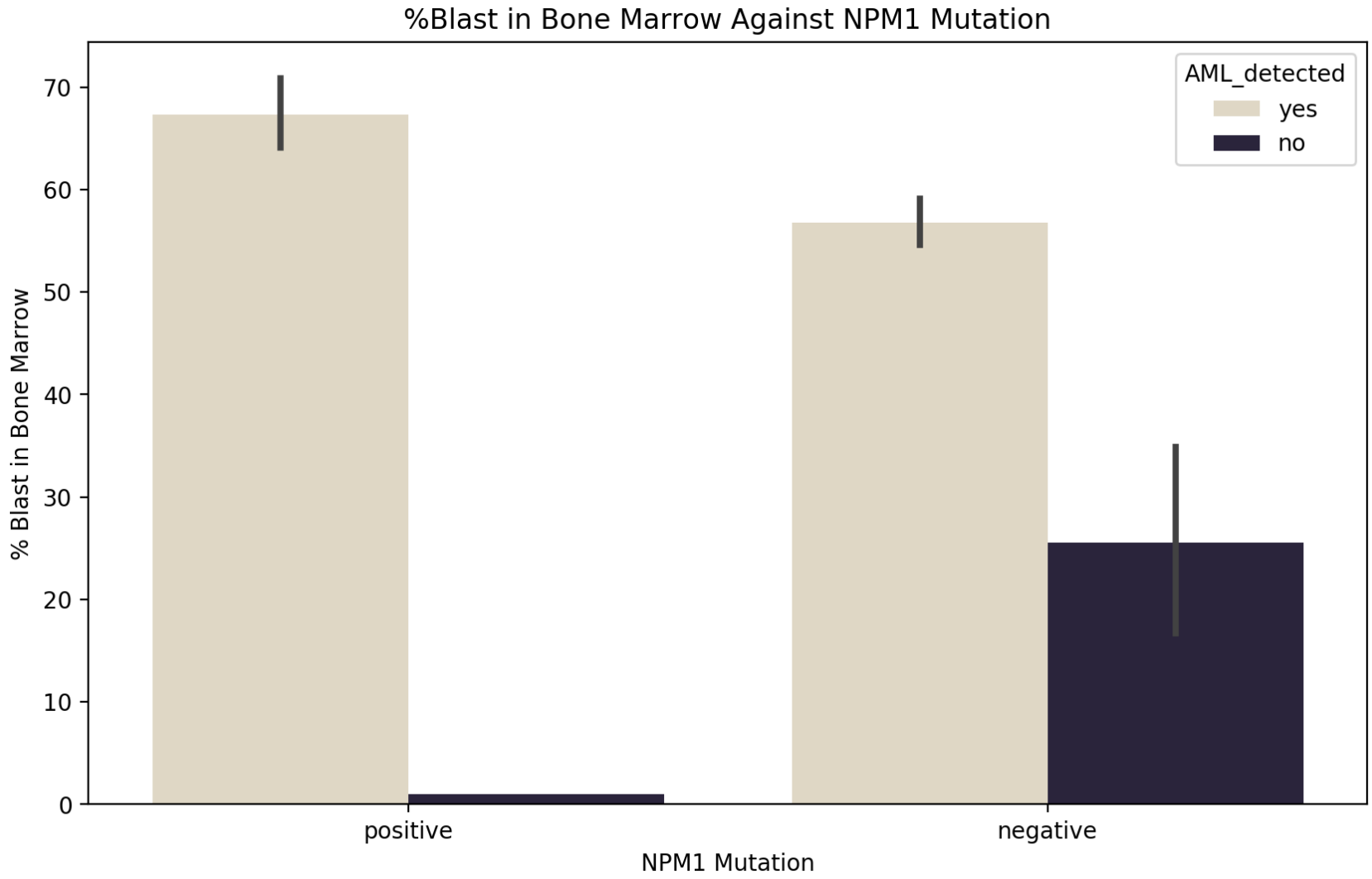
In [77]:
```
sns.countplot(x=clsm_cut["AML_detected"], palette = "ch:s=-.2,r=.6")
plt.gcf().set_size_inches(10, 5)
plt.xlabel('AML_detected')
plt.ylabel('Count')
plt.title("AML_detected Frequency")
```

Out[77]:
```
Text(0.5, 1.0, 'AML_detected Frequency')
```



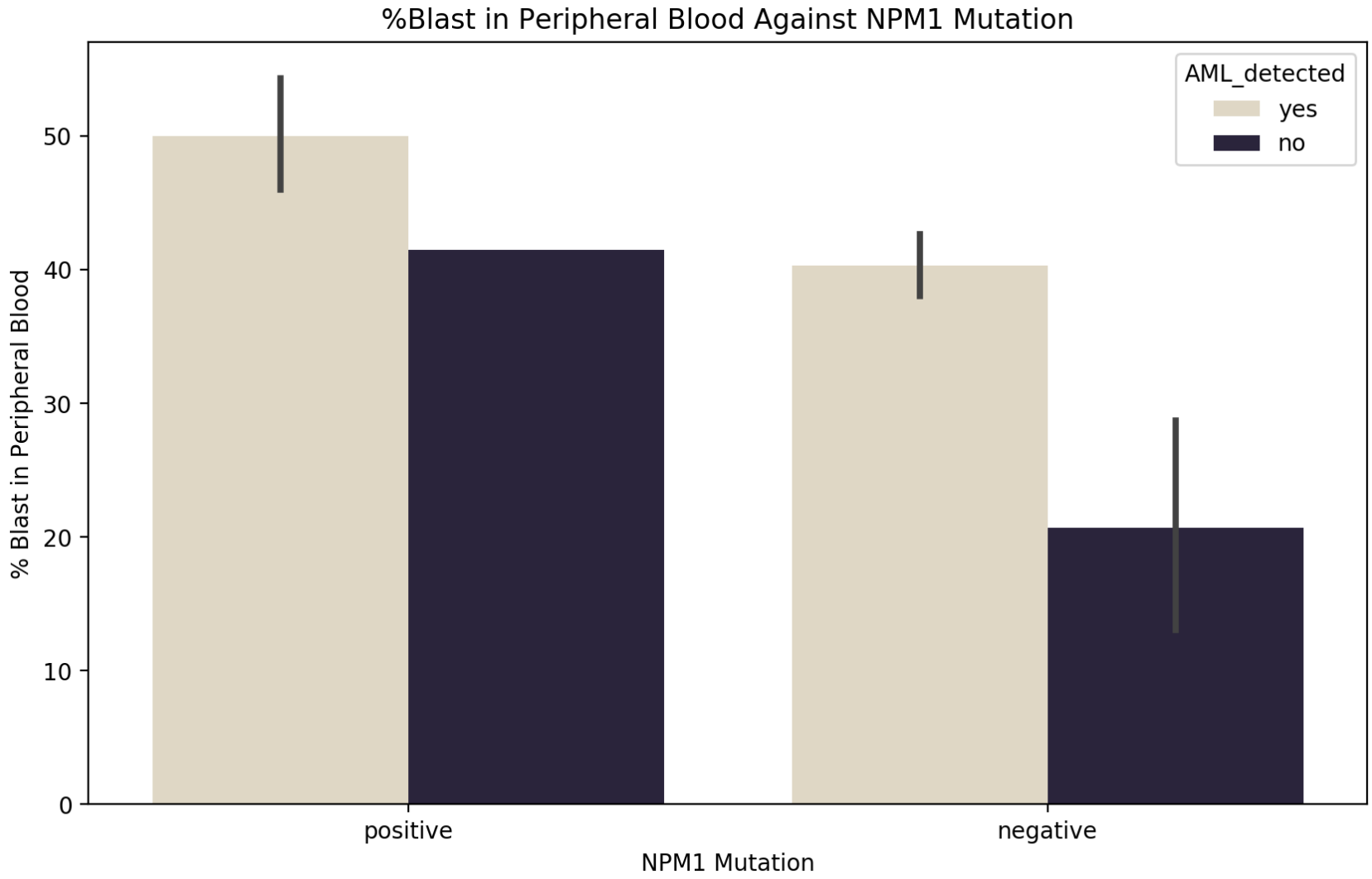
In [78]:
```
sns.barplot(data= clsm_cut,x = 'npm1', y = '%.blasts.in.bm',
            hue = 'AML_detected', palette = "ch:s=-.2,r=.6")
plt.gcf().set_size_inches(10, 6)
plt.xlabel('NPM1 Mutation')
plt.ylabel('% Blast in Bone Marrow')
plt.legend(loc='upper right', title = 'AML_detected')
plt.title("%Blast in Bone Marrow Against NPM1 Mutation")
```

Out[78]:
```
Text(0.5, 1.0, '%Blast in Bone Marrow Against NPM1 Mutation')
```



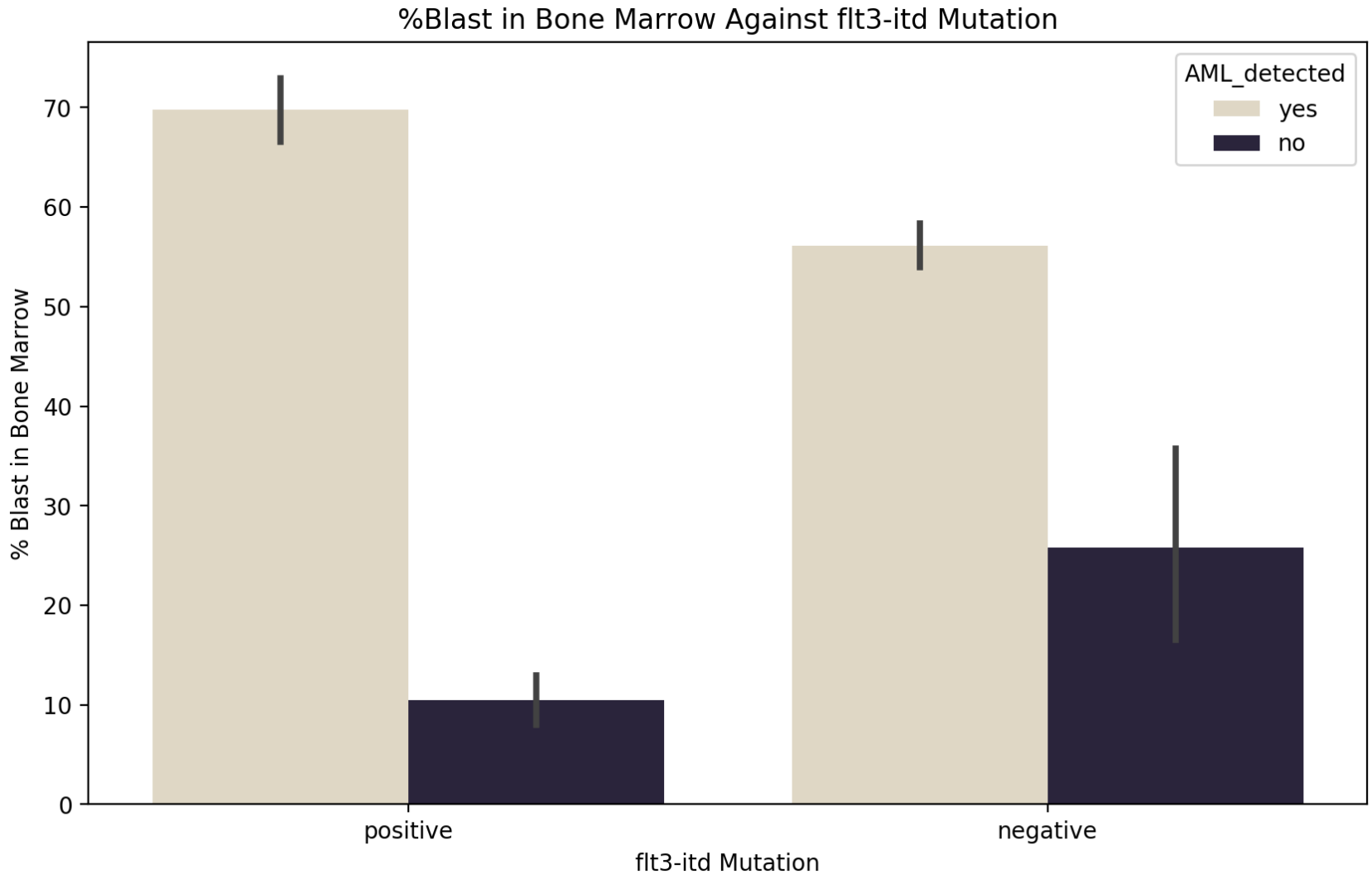
In [79]:
```
sns.barplot(data= clsm_cut,x = 'npm1', y = '%.blasts.in.pb',
            hue = 'AML_detected', palette = "ch:s=-.2,r=.6")
plt.gcf().set_size_inches(10, 6)
plt.xlabel('NPM1 Mutation')
plt.ylabel('% Blast in Peripheral Blood')
plt.legend(loc='upper right', title = 'AML_detected')
plt.title("%Blast in Peripheral Blood Against NPM1 Mutation")
```

Out[79]:
```
Text(0.5, 1.0, '%Blast in Peripheral Blood Against NPM1 Mutation')
```



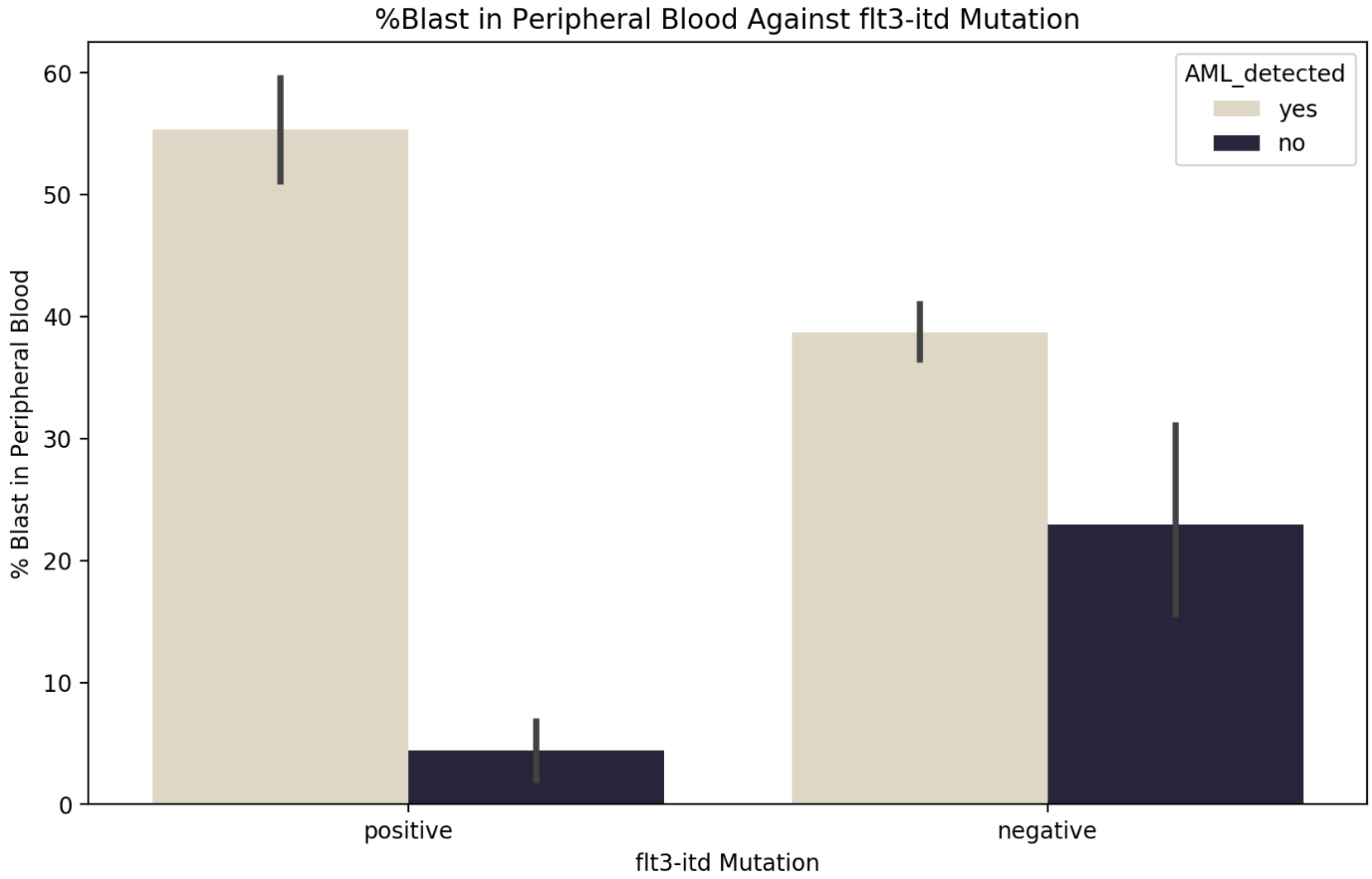
In [80]:
```
sns.barplot(data= clsm_cut,x = 'flt3-itd', y = '%.blasts.in.bm',
            hue = 'AML_detected', palette = "ch:s=-.2,r=.6")
plt.gcf().set_size_inches(10, 6)
plt.xlabel('flt3-itd Mutation')
plt.ylabel('% Blast in Bone Marrow')
plt.legend(loc='upper right', title = 'AML_detected')
plt.title("%Blast in Bone Marrow Against flt3-itd Mutation")
```

Out[80]:
```
Text(0.5, 1.0, '%Blast in Bone Marrow Against flt3-itd Mutation')
```



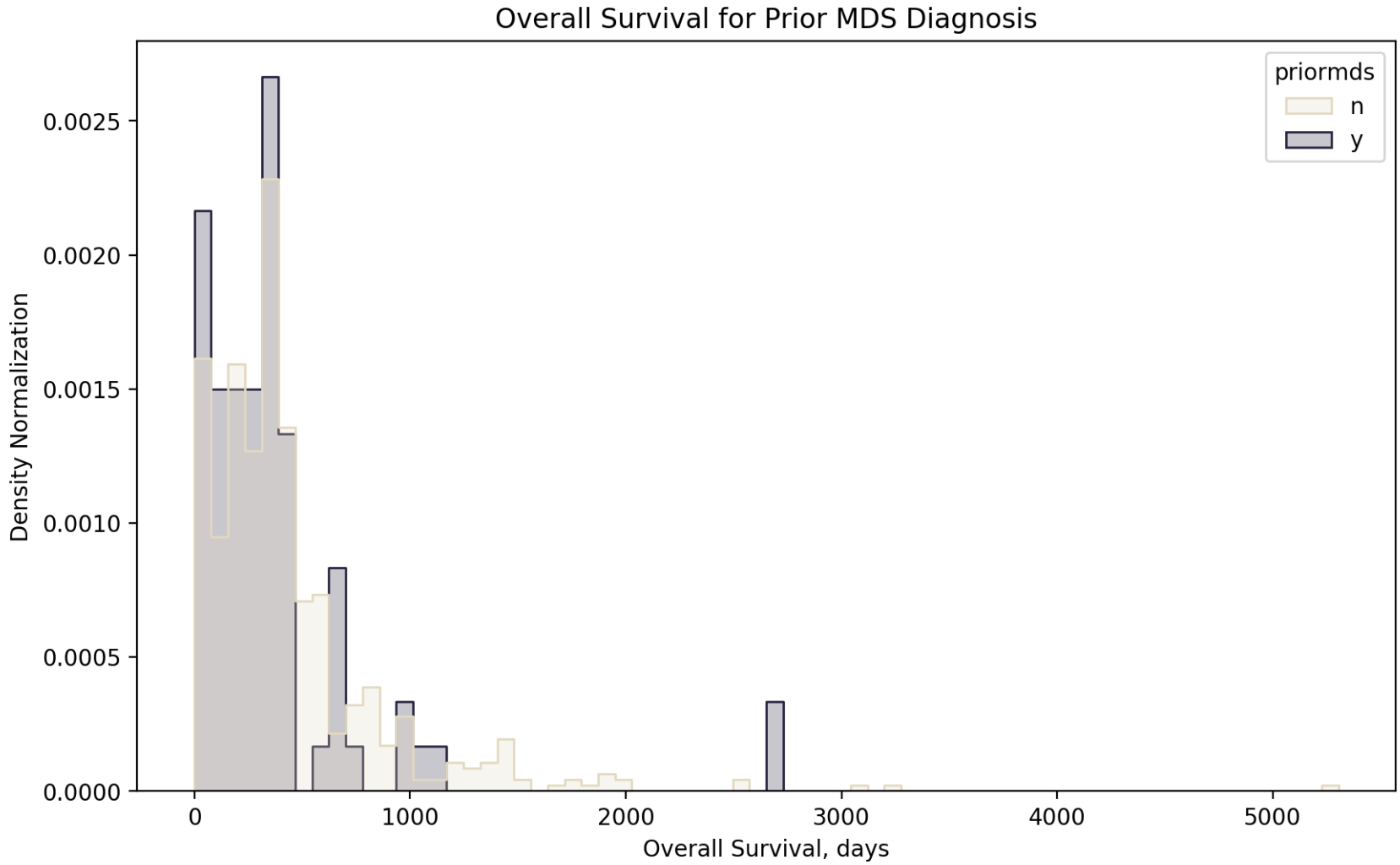
In [81]:
```
sns.barplot(data= clsm_cut,x = 'flt3-itd', y = '%.blasts.in.pb',
            hue = 'AML_detected', palette = "ch:s=-.2,r=.6")
plt.gcf().set_size_inches(10, 6)
plt.xlabel('flt3-itd Mutation')
plt.ylabel('% Blast in Peripheral Blood')
plt.legend(loc='upper right', title = 'AML_detected')
plt.title("%Blast in Peripheral Blood Against flt3-itd Mutation")
```

Out[81]:
```
Text(0.5, 1.0, '%Blast in Peripheral Blood Against flt3-itd Mutation')
```



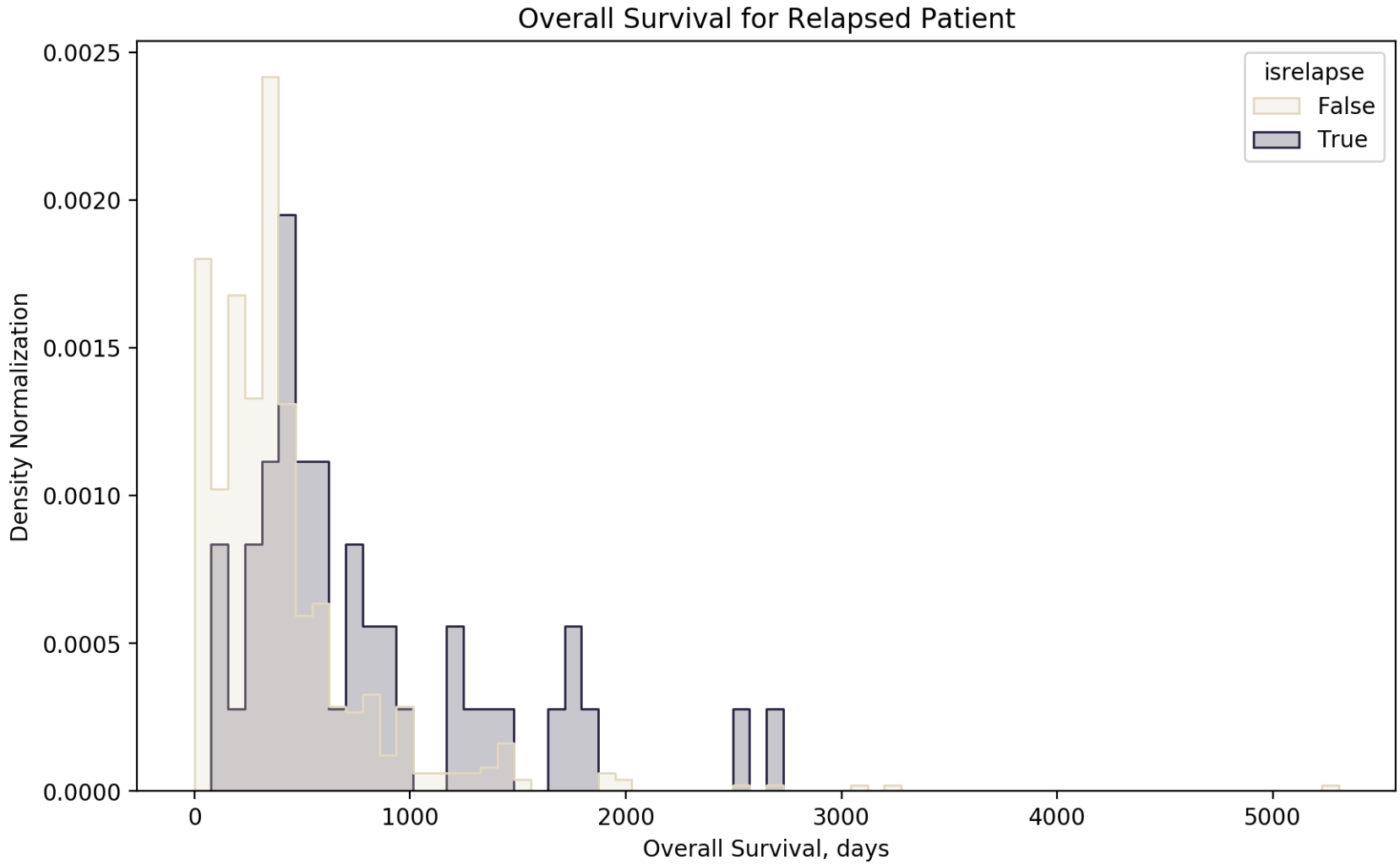
In [82]:
```
sns.histplot(data=clsm_cut, x="overallsurvival", hue="priormds", element="step", palette = "ch:s=-.2,r=.6",
             stat="density", common_norm=False)
plt.gcf().set_size_inches(10, 6)
plt.xlabel('Overall Survival, days')
plt.ylabel('Density Normalization')
#plt.legend(loc='upper right', title = 'Prior MDS Diagnosis')
plt.title("Overall Survival for Prior MDS Diagnosis")
```

Out[82]:
```
Text(0.5, 1.0, 'Overall Survival for Prior MDS Diagnosis')
```



In [83]:
```
sns.histplot(data=clsm_cut, x="overallsurvival", hue="isrelapse", element="step", palette = "ch:s=-.2,r=.6",
             stat="density", common_norm=False)
plt.gcf().set_size_inches(10, 6)
plt.xlabel('Overall Survival, days')
plt.ylabel('Density Normalization')
#plt.legend(loc='upper right', title = 'Prior MDS Diagnosis')
plt.title("Overall Survival for Relapsed Patient")
```

Out[83]:
```
Text(0.5, 1.0, 'Overall Survival for Relapsed Patient')
```



New Dataframe For SageMaker JumpStart Regression Model

Transform select categorical attributes to numerical:

In [84]:
```
#AML_detected
clsm_cut['AML_detected'].replace(['no', 'yes'],
                         [0, 1], inplace=True)

#npm1
clsm_cut['npm1'].replace(['negative', 'positive'],
                         [0, 1], inplace=True)

#flt3-itd
clsm_cut['flt3-itd'].replace(['negative', 'positive'],
                         [0, 1], inplace=True)

#priormalignancynonmyeloid
clsm_cut['priormalignancynonmyeloid'].replace(['n', 'y'],
                         [0, 1], inplace=True)

#priormds
clsm_cut['priormds'].replace(['y', 'n'],
                         [1, 0], inplace=True)

#priormdsmpn
clsm_cut['priormdsmpn'].replace(['n', 'y'],
                         [0, 1], inplace=True)

#priormpn
clsm_cut['priormpn'].replace(['n', 'y'],
                         [0, 1], inplace=True)

#isrelapse
clsm_cut['isrelapse'].replace(['False', 'True'],
                         [0, 1], inplace=True)

#istransformed
clsm_cut['istransformed'].replace(['True', 'False'],
                         [1, 0], inplace=True)
```

In [85]:
```
clsm_t = pd.DataFrame(clsm_cut[['AML_detected', 'npm1', 'flt3-itd', 'isrelapse', 'istransformed',
                                'priormalignancynonmyeloid', 'priormds', 'priormdsmpn', 'priormpn',
                                '%.blasts.in.pb', '%.blasts.in.bm', 'overallsurvival']])
```

In [86]:
```
#Transform data type:
clsm_t['npm1'] = clsm_cut['npm1'].astype(int)
clsm_t['flt3-itd'] = clsm_cut['flt3-itd'].astype(int)

clsm_t['isrelapse'] = clsm_cut['isrelapse'].astype(int)
clsm_t['istransformed'] = clsm_cut['istransformed'].astype(int)
```

New clsm Dataframe Correlation Matrix

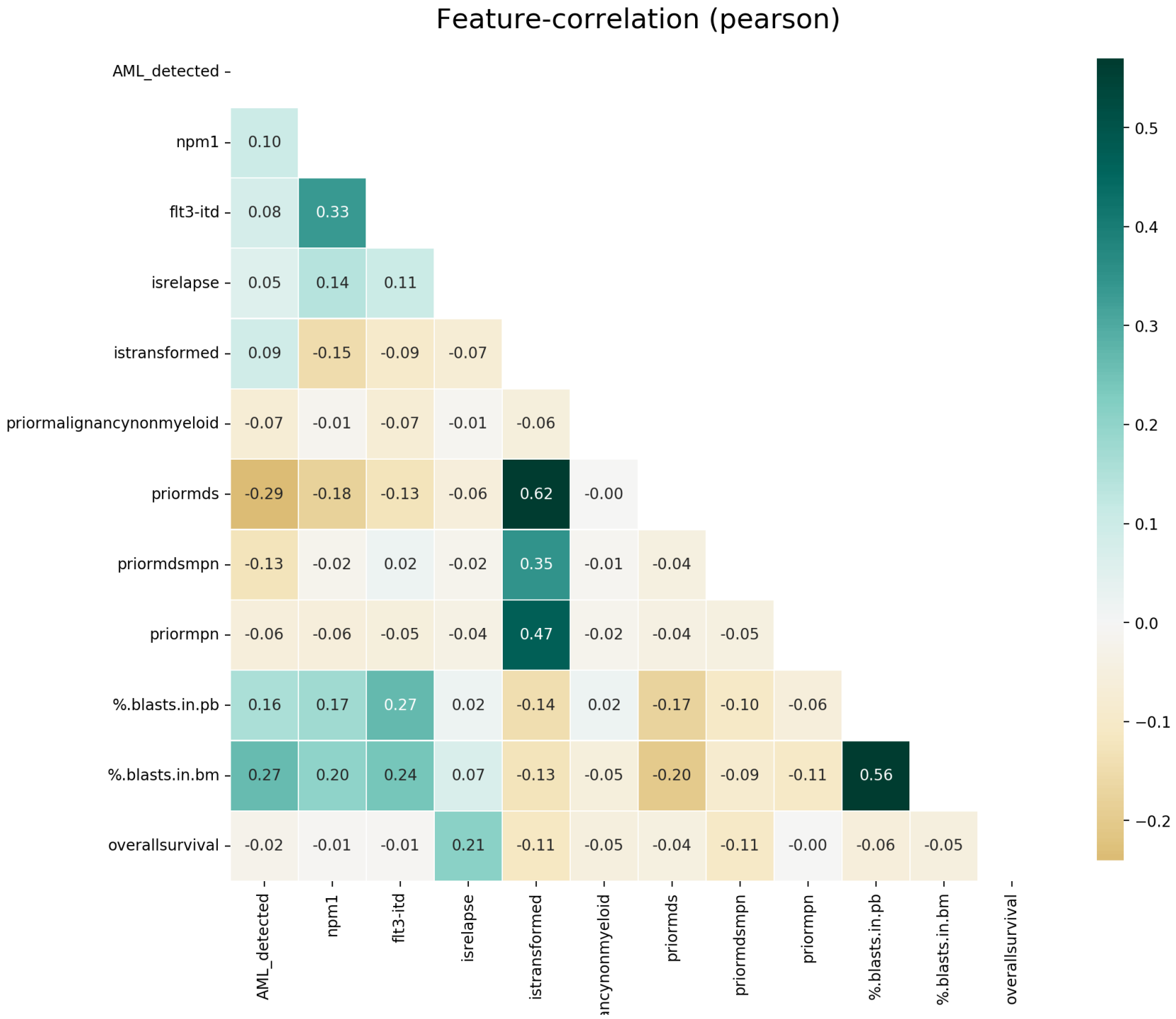
In [87]:
```
clsm_t.corr()
```

Out[87]:

	AML_detected	npm1	flt3-itd	isrelapse	istransformed	priormalignancynonmyeloid	priormds	prio
AML_detected	1.000000	0.098997	0.077525	0.054383	0.089238	-7.245182e-02	-2.912038e-01	
npm1	0.098997	1.000000	0.333543	0.140481	-0.148233	-1.257739e-02	-1.771024e-01	
flt3-itd	0.077525	0.333543	1.000000	0.107818	-0.092782	-7.228395e-02	-1.272762e-01	
isrelapse	0.054383	0.140481	0.107818	1.000000	-0.072971	-9.623173e-03	-6.051426e-02	
istransformed	0.089238	-0.148233	-0.092782	-0.072971	1.000000	-5.562376e-02	6.200179e-01	
priormalignancynonmyeloid	-0.072452	-0.012577	-0.072284	-0.009623	-0.055624	1.000000e+00	-1.056121e-17	
priormds	-0.291204	-0.177102	-0.127276	-0.060514	0.620018	-1.056121e-17	1.000000e+00	
priormdsmpn	-0.127707	-0.019763	0.022049	-0.020414	0.346275	-9.820928e-03	-4.405654e-02	
priormpn	-0.057154	-0.059377	-0.054524	-0.037020	0.472862	-1.913898e-02	-4.227151e-02	
%.blasts.in.pb	0.155752	0.174675	0.271851	0.020293	-0.141930	2.215108e-02	-1.739113e-01	
%.blasts.in.bm	0.265724	0.201114	0.242420	0.069284	-0.128224	-5.402601e-02	-2.015171e-01	
overallsurvival	-0.022216	-0.006728	-0.008150	0.210147	-0.113017	-4.893419e-02	-4.466673e-02	

In [88]:
```
klib.corr_plot(clsm_t)
```

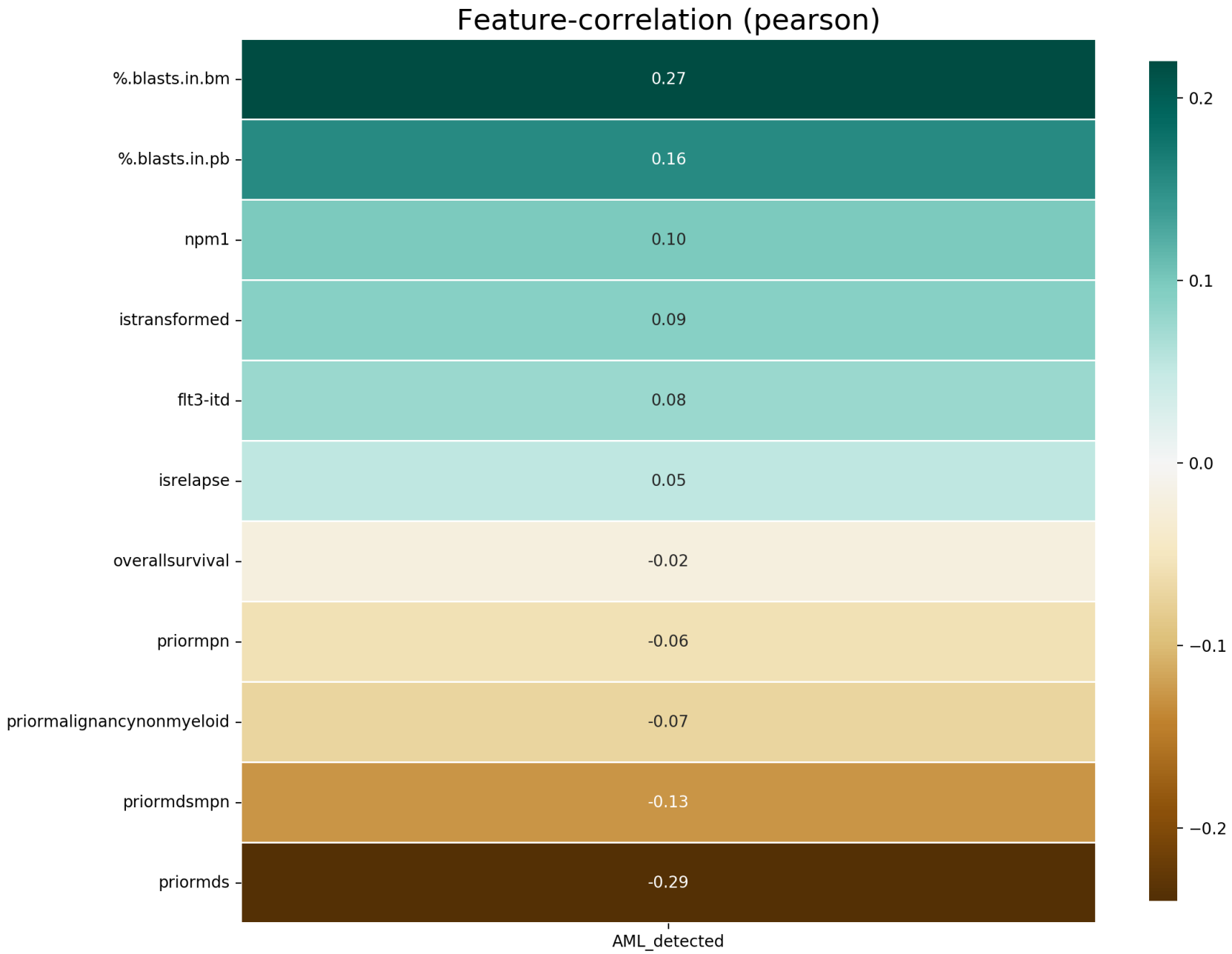
Out[88]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fce4bac4950>
```


Feature-correlation (pearson)
AML_detected
npm1 0.10
flt3-itd 0.08 0.33
isrelapse 0.05 0.14 0.11
istransformed 0.09 -0.15 -0.09 -0.07
priormalignancynonmyeloid -0.07 -0.01 -0.07 -0.01 -0.06
priormds -0.29 -0.18 -0.13 -0.06 0.62 -0.00
priormdsmpn -0.13 -0.02 0.02 -0.02 0.35 -0.01 -0.04
priormpn -0.06 -0.06 -0.05 -0.04 0.47 -0.02 -0.04 -0.05
%.blasts.in.pb 0.16 0.17 0.27 0.02 -0.14 0.02 -0.17 -0.10 -0.06
%.blasts.in.bm 0.27 0.20 0.24 0.07 -0.13 -0.05 -0.20 -0.09 -0.11 0.56
overallsurvival -0.02 -0.01 -0.01 0.21 -0.11 -0.05 -0.04 -0.11 -0.00 -0.06 -0.05
AML_detected
npm1
flt3-itd
isrelapse
istransformed
ancynonmyeloid
priormds
priormdsmpn
priormpn
%.blasts.in.pb
%.blasts.in.bm
overallsurvival
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2

priormalign

In [89]:
```
klib.corr_plot(clsm_t, target='AML_detected')
```

Out[89]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fce46246bd0>
```



One-Hot encoding

In [90]:
```
clsm_t = pd.get_dummies(clsm_t, columns= ['npm1', 'flt3-itd', 'priormalignancynonmyeloid',
                                          'priormds', 'priormdsmpn', 'priormpn', 'isrelapse', 'istransformed'
```

In [91]:
```
clsm_t.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 672 entries, 0 to 671
Data columns (total 20 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   AML_detected                 672 non-null    int64  
 1   %.blasts.in.pb               672 non-null    float64
 2   %.blasts.in.bm               672 non-null    float64
 3   overallsurvival              672 non-null    float64
 4   npm1_0                       672 non-null    uint8  
 5   npm1_1                       672 non-null    uint8  
 6   flt3-itd_0                   672 non-null    uint8  
 7   flt3-itd_1                   672 non-null    uint8  
 8   priormalignancynonmyeloid_0  672 non-null    uint8  
 9   priormalignancynonmyeloid_1  672 non-null    uint8  
 10  priormds_0                   672 non-null    uint8  
 11  priormds_1                   672 non-null    uint8  
 12  priormdsmpn_0                672 non-null    uint8  
 13  priormdsmpn_1                672 non-null    uint8  
 14  priormpn_0                   672 non-null    uint8  
 15  priormpn_1                   672 non-null    uint8  
 16  isrelapse_0                  672 non-null    uint8  
 17  isrelapse_1                  672 non-null    uint8  
 18  istransformed_0              672 non-null    uint8  
 19  istransformed_1              672 non-null    uint8  
dtypes: float64(3), int64(1), uint8(16)
memory usage: 31.6 KB
```

In [92]:
```
clsm_t.head()
```

Out[92]:

	AML_detected	%.blasts.in.pb	%.blasts.in.bm	overallsurvival	npm1_0	npm1_1	flt3-itd_0	flt3-itd_1	priormalignancynonmyeloid_0	priormalign
0	1	97.0	94.0	425.0	0	1	0	1	1	
1	1	19.0	80.0	419.0	1	0	0	1	1	
2	1	99.0	91.0	541.0	1	0	0	1	1	
3	1	97.0	97.0	511.0	0	1	0	1	1	
4	1	80.0	87.0	419.0	1	0	0	1	1	

Transform Headers

In [93]:
```
clsm_t = clsm_t.rename(columns={ '%.blasts.in.pb': 'Feature_1', '%.blasts.in.bm': 'Feature_2',
                                 'overallsurvival': 'Feature_3',
                                  'npm1_0': 'Feature_4', 'npm1_1': 'Feature_5',
                                 'flt3-itd_0': 'Feature_6', 'flt3-itd_1': 'Feature_7',
                                  'priormalignancynonmyeloid_0': 'Feature_8', 'priormalignancynonmyeloid_1'
                                  'priormds_0': 'Feature_10', 'priormds_1': 'Feature_11',
                                 'priormdsmpn_0': 'Feature_12', 'priormdsmpn_1': 'Feature_13',
                                  'priormpn_0': 'Feature_14', 'priormpn_1': 'Feature_15',
                                  'isrelapse_0': 'Feature_16', 'isrelapse_1': 'Feature_17',
                                 'istransformed_0': 'Feature_18', 'istransformed_1': 'Feature_19' })
```

In [94]:
```
clsm_t.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 672 entries, 0 to 671
Data columns (total 20 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   AML_detected  672 non-null    int64
 1   Feature_1     672 non-null    float64
 2   Feature_2     672 non-null    float64
 3   Feature_3     672 non-null    float64
 4   Feature_4     672 non-null    uint8
 5   Feature_5     672 non-null    uint8
 6   Feature_6     672 non-null    uint8
 7   Feature_7     672 non-null    uint8
 8   Feature_8     672 non-null    uint8
 9   Feature_9     672 non-null    uint8
 10  Feature_10    672 non-null    uint8
 11  Feature_11    672 non-null    uint8
 12  Feature_12    672 non-null    uint8
 13  Feature_13    672 non-null    uint8
 14  Feature_14    672 non-null    uint8
 15  Feature_15    672 non-null    uint8
 16  Feature_16    672 non-null    uint8
 17  Feature_17    672 non-null    uint8
 18  Feature_18    672 non-null    uint8
 19  Feature_19    672 non-null    uint8
dtypes: float64(3), int64(1), uint8(16)
memory usage: 31.6 KB
```

Save New Pre-Processed clsm Dataframe to S3

In [95]:
```
clsm_t.to_csv('clsm_t.csv')
```

In [96]:
```
#Manually upload into S3
!aws s3 ls s3://team4rawdatasets/CSV/Input/
```

```
                           PRE OHSU_BeatAML_ClinicalSummary/
                           PRE OpenCell_ProteinInteraction/
2023-03-21 01:19:41          0
2023-04-12 05:44:22      30850 clsm_t.csv
```

In [97]:
```
from IPython.core.display import display, HTML

display(
    HTML(
        '<b>Review <a target="blank" href="https://s3.console.aws.amazon.com/s3/buckets/team4rawdatasets?region=us-ea
            region, account_id, region
        )
    )
)
```

Review S3 Output Bucket

Split the Data into Train, Test, and Validation sets

In [98]:
```
clsm_t.info()
```

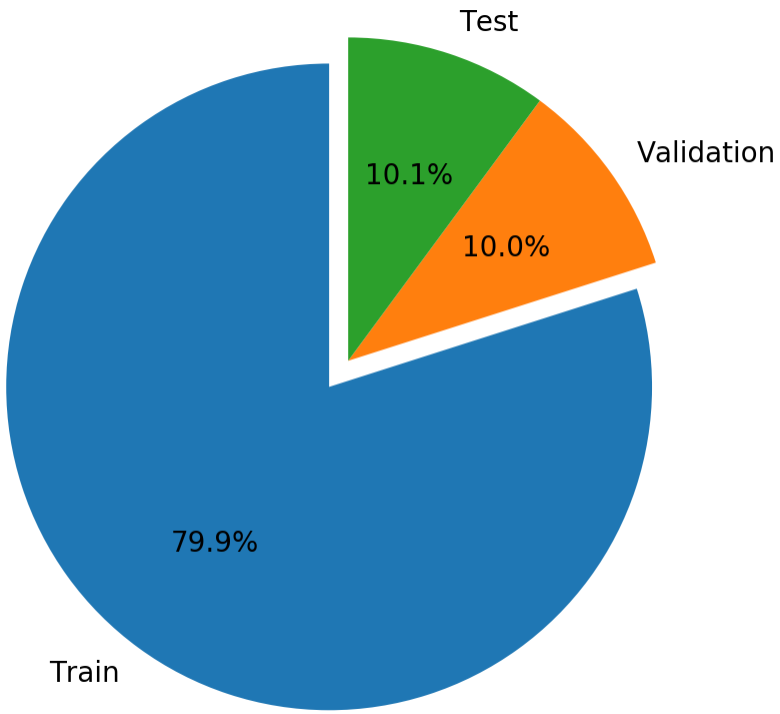
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 672 entries, 0 to 671
Data columns (total 20 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   AML_detected  672 non-null    int64  
 1   Feature_1     672 non-null    float64
 2   Feature_2     672 non-null    float64
 3   Feature_3     672 non-null    float64
 4   Feature_4     672 non-null    uint8  
 5   Feature_5     672 non-null    uint8  
 6   Feature_6     672 non-null    uint8  
 7   Feature_7     672 non-null    uint8  
 8   Feature_8     672 non-null    uint8  
 9   Feature_9     672 non-null    uint8  
 10  Feature_10    672 non-null    uint8  
 11  Feature_11    672 non-null    uint8  
 12  Feature_12    672 non-null    uint8  
 13  Feature_13    672 non-null    uint8  
 14  Feature_14    672 non-null    uint8  
 15  Feature_15    672 non-null    uint8  
 16  Feature_16    672 non-null    uint8  
 17  Feature_17    672 non-null    uint8  
 18  Feature_18    672 non-null    uint8  
 19  Feature_19    672 non-null    uint8  
dtypes: float64(3), int64(1), uint8(16)
memory usage: 31.6 KB
```

In [99]:

```python
from sklearn.model_selection import train_test_split

# Split all data into 80% train and 20% holdout
clsm_train, clsm_holdout = train_test_split(clsm_t, test_size=0.20, random_state=42)

# Split holdout data into 50% validation and 50% test
clsm_validation, clsm_test = train_test_split(clsm_holdout, test_size=0.50, random_state=42)
```

In [100…

```
# Pie chart, where the slices will be ordered and plotted counter-clockwise:
labels = ["Train", "Validation", "Test"]
sizes = [len(clsm_train.index), len(clsm_validation.index), len(clsm_test.index)]
explode = (0.1, 0, 0)

fig1, ax1 = plt.subplots()

ax1.pie(sizes, explode=explode, labels=labels, autopct="%1.1f%%", startangle=90)

# Equal aspect ratio ensures that pie is drawn as a circle.
ax1.axis("equal")

plt.show()
```



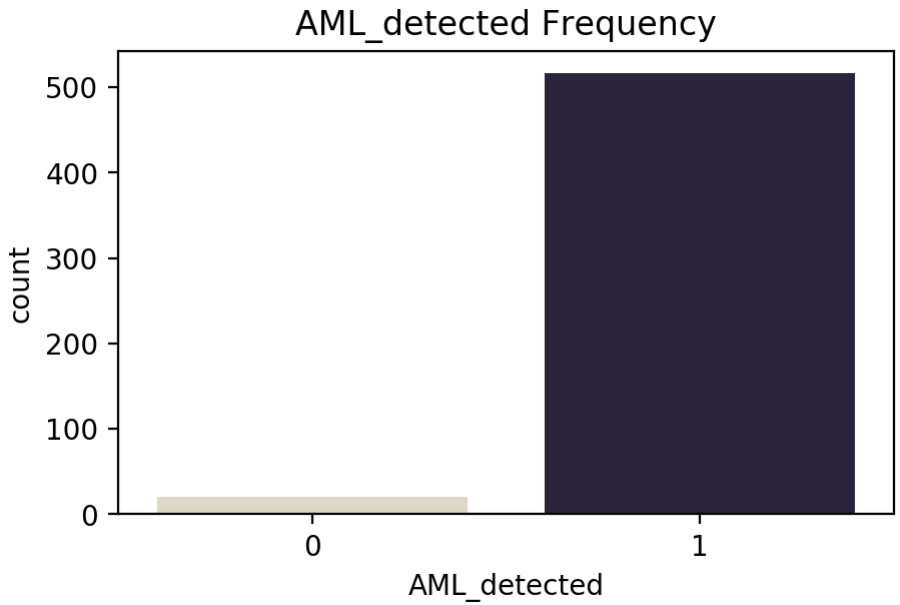
Show 80% Train Data Split

In [101…

```
clsm_train.shape
```

Out[101]: (537, 20)

In [102…
```
sns.countplot(x=clsm_train["AML_detected"], palette = "ch:s=-.2,r=.6")
plt.xlabel('AML_detected')
plt.title('AML_detected Frequency')
plt.gcf().set_size_inches(5, 3)
```

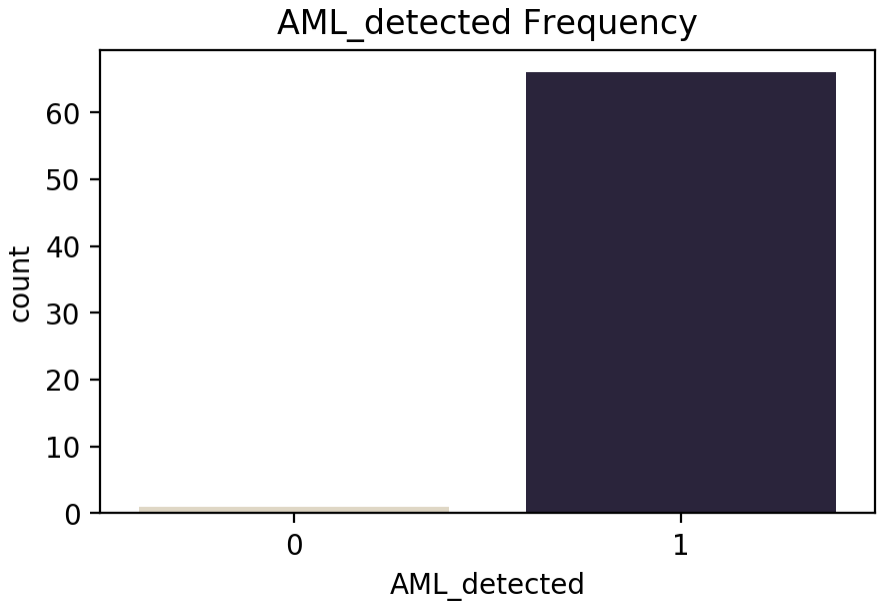


Show 10% Validation Data Split

In [103…
```
clsm_validation.shape
```

Out[103]: (67, 20)

In [104…
```
sns.countplot(x=clsm_validation["AML_detected"], palette = "ch:s=-.2,r=.6")
plt.xlabel('AML_detected')
plt.title('AML_detected Frequency')
plt.gcf().set_size_inches(5, 3)
```



In [105…
```
clsm_validation.to_csv('clsm_validation_.csv')
```

In [106…
```
#Manually upload into S3
!aws s3 ls s3://clsm/tabular_regressonehot/
```

```
                           PRE output/
                           PRE test/
                           PRE train/
                           PRE validation/
2023-04-11 01:26:47          0
```

In [107…
```
display(
    HTML(
        '<b>Review <a target="blank" href="https://s3.console.aws.amazon.com/s3/buckets/clsm?prefix=tabular_regressor
            region, account_id, region
        )
    )
)
```

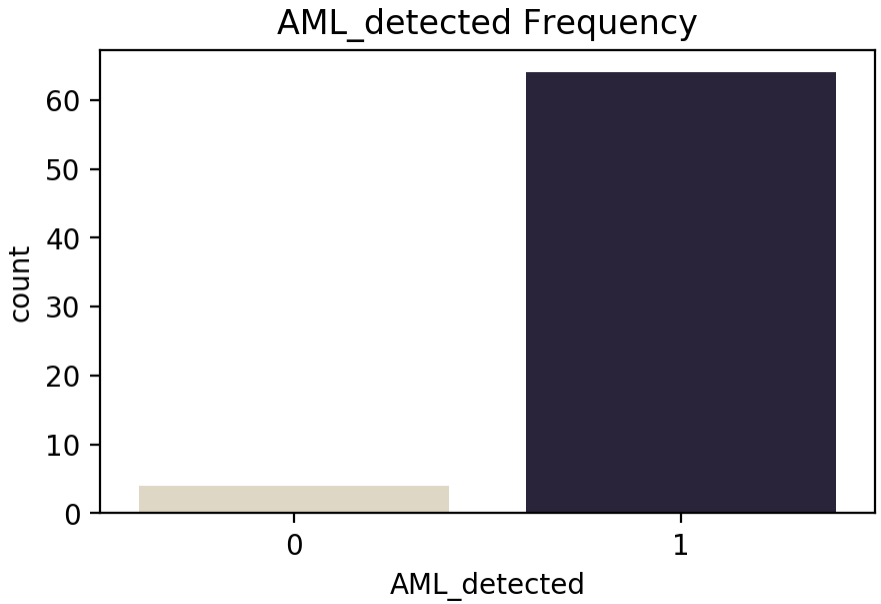
Review S3 Output Bucket

Show 10% Test Data Split

In [108…
```
clsm_test.shape
```

Out[108]:
```
(68, 20)
```

In [109…
```
sns.countplot(x=clsm_test["AML_detected"], palette = "ch:s=-.2,r=.6")
plt.xlabel('AML_detected')
plt.title('AML_detected Frequency')
plt.gcf().set_size_inches(5, 3)
```



In [110…
```
clsm_test.to_csv('clsm_test_.csv')
```

In [111…
```
#Manually upload into S3
!aws s3 ls s3://clsm/tabular_regressonehot/
```

```
                           PRE output/
                           PRE test/
                           PRE train/
                           PRE validation/
2023-04-11 01:26:47          0
```

In [112…]
```
display(
    HTML(
        '<b>Review <a target="blank" href="https://s3.console.aws.amazon.com/s3/buckets/clsm?prefix=tabular_regressor
            region, account_id, region
        )
    )
)
```

Review S3 Output Bucket

Balance Training Dataset

Training Dataset:

In [113…]
```
clsm_train["AML_detected"].value_counts()
```

Out[113]:
```
1    516
0     21
Name: AML_detected, dtype: int64
```

Balancing Equation: n = [p(records)-rare]/(1-p), where p=0.50, records=537, rare=21.

In [114…]
```
#resampling of training data set
to_resample= clsm_train.loc[clsm_train["AML_detected"] == 0] #isolate all records of AML_detected
our_resample=to_resample.sample(n=495, replace=True) #sample w/ replacement
clsm_t_rebal=pd.concat([clsm_train, our_resample]) #combine original training set w/ resampled records
clsm_t_rebal["AML_detected"].value_counts()
```

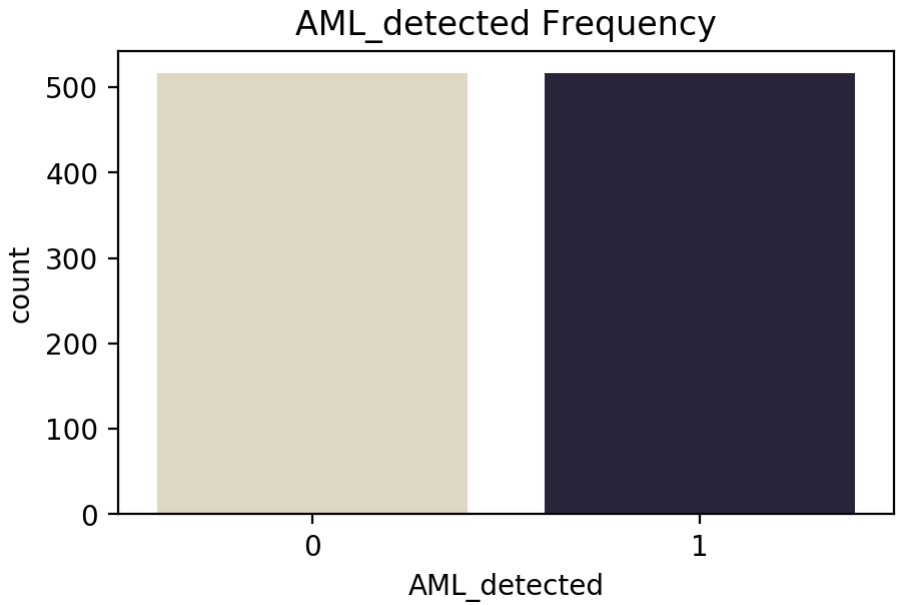
Out[114]:
```
1    516
0    516
Name: AML_detected, dtype: int64
```

In [115…]
```
clsm_t_rebal.shape
```

Out[115]:
```
(1032, 20)
```

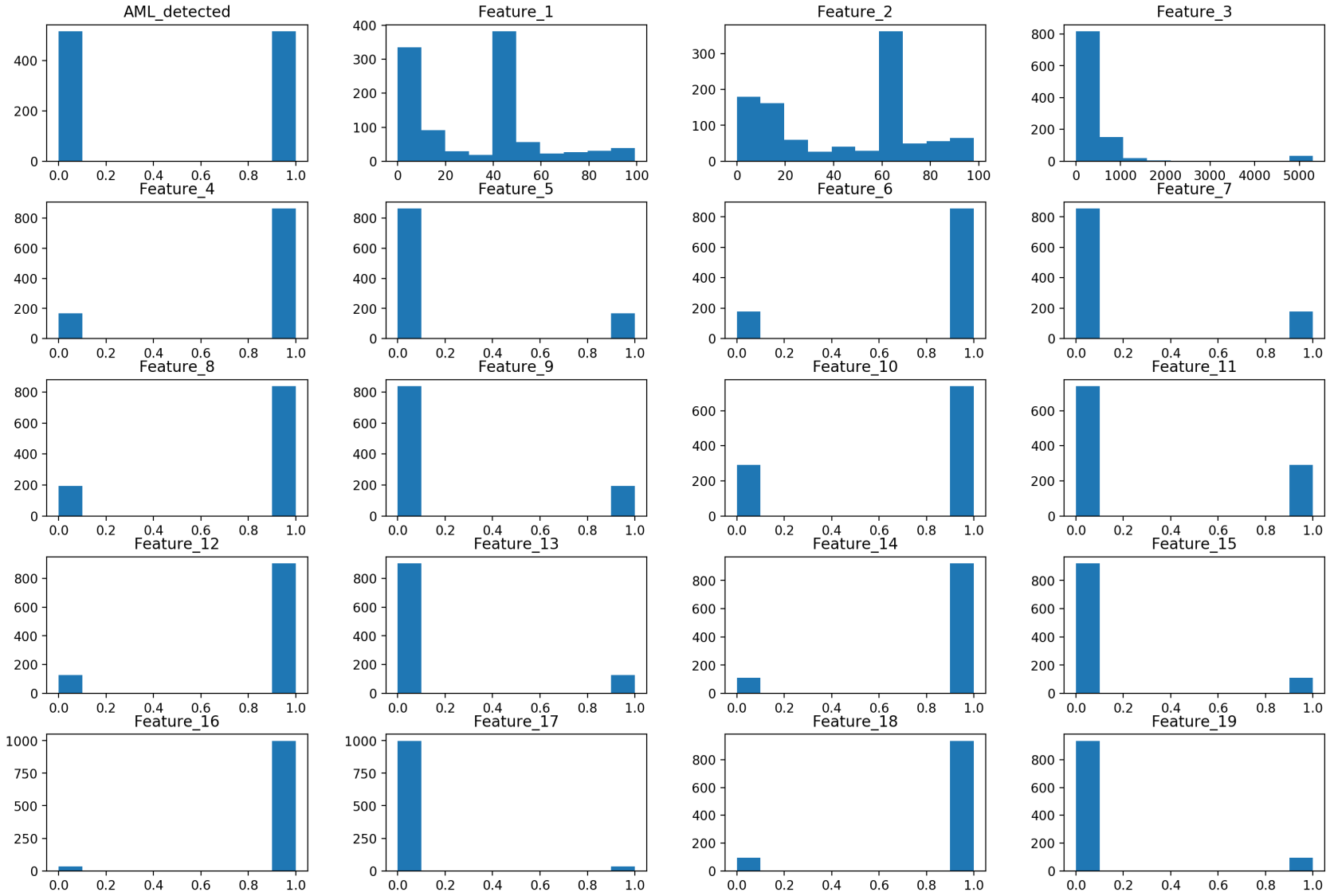
In [116…

```
sns.countplot(x=clsm_t_rebal["AML_detected"], palette = "ch:s=-.2,r=.6")
plt.xlabel('AML_detected')
plt.title('AML_detected Frequency')
plt.gcf().set_size_inches(5, 3)
```



In [117…

```
#clsm_t_rebal Distribution
clsm_t_rebal.hist(grid=False, figsize=(18,12))
plt.show()
```



In [118…

```
clsm_t_rebal.head()
```

Out[118]:

	AML_detected	Feature_1	Feature_2	Feature_3	Feature_4	Feature_5	Feature_6	Feature_7	Feature_8	Feature_9	Feature_10	Featur
480	1	95.0	95.0	201.0	0	1	1	0	1	0	1	
605	1	57.0	40.0	179.0	0	1	1	0	0	1	1	
61	1	41.5	63.0	1993.0	1	0	1	0	1	0	1	
145	1	16.0	63.0	323.0	1	0	1	0	1	0	0	
353	1	0.0	30.0	323.0	1	0	1	0	1	0	1	

In [119…
```
clsm_t_rebal.to_csv('clsm_t_rebal_.csv')
```

In [120…
```
#Manually upload into S3
!aws s3 ls s3://clsm/tabular_regressonehot/
```

```
                           PRE output/
                           PRE test/
                           PRE train/
                           PRE validation/
2023-04-11 01:26:47          0
```

In [121…
```
display(
    HTML(
        '<b>Review <a target="blank" href="https://s3.console.aws.amazon.com/s3/buckets/clsm?prefix=tabular_regressor
            region, account_id, region
        )
    )
)
```

Review S3 Output Bucket

SageMaker JumpStart: XGBoost Model

Set-Up

In [122…
```
!pip install ipywidgets==7.0.0 --quiet
```

```
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system p
ackage manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv

[notice] A new release of pip is available: 23.0.1 -> 23.1
[notice] To update, run: pip install --upgrade pip
```

In [123…

```python
import json
from sagemaker.session import Session

sagemaker_session = Session()
aws_role = sagemaker_session.get_caller_identity_arn()
aws_region = boto3.Session().region_name
sess = sagemaker.Session()
```

Retrieve Training Artifacts

In [124…

```python
model_id, model_version = "xgboost-regression-model", "*"
```

In [125…

```python
from ipywidgets import Dropdown
from sagemaker.jumpstart.notebook_utils import list_jumpstart_models
from sagemaker.jumpstart.filters import And

# Retrieves all xgboost and sklearn regression models available by SageMaker Built-In Algorithms.
filter_value = And(f"framework in ['xgboost', 'sklearn']", "task == regression")
text_embedding_models = list_jumpstart_models(filter=filter_value)

# display the model-ids in a dropdown to select a model for inference.
model_dropdown = Dropdown(
    options=text_embedding_models,
    value=model_id,
    description="Select a model",
    style={"description_width": "initial"},
    layout={"width": "max-content"},
)
```

Chose a model for training

In [126…

```python
display(model_dropdown)
```

```
A Jupyter Widget
```

In [127…

```
from sagemaker import image_uris, model_uris, script_uris

train_model_id, train_model_version, train_scope = model_dropdown.value, "*", "training"
training_instance_type = "ml.m5.xlarge"

# Retrieve the docker image
train_image_uri = image_uris.retrieve(
    region=None,
    framework=None,
    model_id=train_model_id,
    model_version=train_model_version,
    image_scope=train_scope,
    instance_type=training_instance_type,
)
# Retrieve the training script
train_source_uri = script_uris.retrieve(
    model_id=train_model_id, model_version=train_model_version, script_scope=train_scope
)
# Retrieve the pre-trained model tarball to further fine-tune
train_model_uri = model_uris.retrieve(
    model_id=train_model_id, model_version=train_model_version, model_scope=train_scope
)
```

Set Training Parameters

In [128…

```
# Sample training data is available in this bucket
training_dataset_s3_path = "s3://clsm/tabular_regressonehot/"

s3_output_location = "s3://clsm/tabular_regressonehot/output/"
```

In [129…

```
from sagemaker import hyperparameters

# Retrieve the default hyper-parameters for fine-tuning the model
hyperparameters = hyperparameters.retrieve_default(
    model_id=train_model_id, model_version=train_model_version
)

# [Optional] Override default hyperparameters with custom values
hyperparameters["num_boost_round"] = "500"
hyperparameters["reg_lambda"] = "3"
print(hyperparameters)
```

```
{'num_boost_round': '500', 'early_stopping_rounds': '30', 'learning_rate': '0.3', 'gamma': '0', 'min_child_weight':
'1', 'max_depth': '6', 'subsample': '1', 'colsample_bytree': '1', 'reg_lambda': '3', 'reg_alpha': '0'}
```

Start Training

In [130…

```python
from sagemaker.estimator import Estimator
from sagemaker.utils import name_from_base

training_job_name = name_from_base(f"clsm-rebal-{train_model_id}-training")

# Create SageMaker Estimator instance
tabular_estimator = Estimator(
    role=aws_role,
    image_uri=train_image_uri,
    source_dir=train_source_uri,
    model_uri=train_model_uri,
    entry_point="transfer_learning.py",
    instance_count=1,
    instance_type=training_instance_type,
    max_run=360000,
    hyperparameters=hyperparameters,
    output_path=s3_output_location,
)

# Launch a SageMaker Training job by passing s3 path of the training data
tabular_estimator.fit({"training": "s3://clsm/tabular_regressonehot/train/clsm_t_rebal_.csv",
                       "validation": "s3://clsm/tabular_regressonehot/validation/clsm_validation_.csv" }, logs=True,
```

```
INFO:sagemaker:Creating training-job with name: clsm-rebal-xgboost-regression-model-tra-2023-04-17-02-35-14-214
```

```
2023-04-17 02:35:18 Starting - Starting the training job...
2023-04-17 02:35:34 Starting - Preparing the instances for training...
2023-04-17 02:36:21 Downloading - Downloading input data...
2023-04-17 02:36:43 Training - Downloading the training image...
2023-04-17 02:37:24 Uploading - Uploading generated training model[2023-04-17 02:37:15.789 ip-10-0-175-156.ec2.inter
nal:7 INFO utils.py:28] RULE_JOB_STOP_SIGNAL_FILENAME: None
[2023-04-17 02:37:15.814 ip-10-0-175-156.ec2.internal:7 INFO profiler_config_parser.py:111] User has disabled profil
er.
[2023-04-17:02:37:15:INFO] Imported framework sagemaker_xgboost_container.training
[2023-04-17:02:37:15:INFO] No GPUs detected (normal if no gpus installed)
[2023-04-17:02:37:15:INFO] Invoking user training script.
[2023-04-17:02:37:16:INFO] Module transfer_learning does not provide a setup.py.
Generating setup.py
[2023-04-17:02:37:16:INFO] Generating setup.cfg
[2023-04-17:02:37:16:INFO] Generating MANIFEST.in
[2023-04-17:02:37:16:INFO] Installing module with the following command:
/miniconda3/bin/python3 -m pip install . -r requirements.txt
Processing /opt/ml/code
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Processing ./lib/sagemaker_jumpstart_script_utilities/sagemaker_jumpstart_script_utilities-1.0.1-py2.py3-none-any.wh
l
Building wheels for collected packages: transfer-learning
  Building wheel for transfer-learning (setup.py): started
  Building wheel for transfer-learning (setup.py): finished with status 'done'
  Created wheel for transfer-learning: filename=transfer_learning-1.0.0-py2.py3-none-any.whl size=12553 sha256=ac01a
8e54c7bc5c42d108aaad4296f09a2916e14628573cb9e59a834d6469772
  Stored in directory: /home/model-server/tmp/pip-ephem-wheel-cache-tksdojds/wheels/3e/0f/51/2f1df833dd0412c1bc2f5ee
56baac195b5be563353d111dca6
Successfully built transfer-learning
Installing collected packages: transfer-learning, sagemaker-jumpstart-script-utilities
Successfully installed sagemaker-jumpstart-script-utilities-1.0.1 transfer-learning-1.0.0
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system p
ackage manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
[notice] A new release of pip is available: 23.0.1 -> 23.1
[notice] To update, run: pip install --upgrade pip
[2023-04-17:02:37:17:INFO] No GPUs detected (normal if no gpus installed)
[2023-04-17:02:37:17:INFO] Invoking user script
Training Env:
{
    "additional_framework_parameters": {},
    "channel_input_dirs": {
        "model": "/opt/ml/input/data/model",
        "training": "/opt/ml/input/data/training",
```

```
    "validation": "/opt/ml/input/data/validation"
},
"current_host": "algo-1",
"framework_module": "sagemaker_xgboost_container.training:main",
"hosts": [
    "algo-1"
],
"hyperparameters": {
    "colsample_bytree": "1",
    "early_stopping_rounds": "30",
    "gamma": "0",
    "learning_rate": "0.3",
    "max_depth": "6",
    "min_child_weight": "1",
    "num_boost_round": "500",
    "reg_alpha": "0",
    "reg_lambda": "3",
    "subsample": "1"
},
"input_config_dir": "/opt/ml/input/config",
"input_data_config": {
    "model": {
        "ContentType": "application/x-sagemaker-model",
        "TrainingInputMode": "File",
        "S3DistributionType": "FullyReplicated",
        "RecordWrapperType": "None"
    },
    "training": {
        "TrainingInputMode": "File",
        "S3DistributionType": "FullyReplicated",
        "RecordWrapperType": "None"
    },
    "validation": {
        "TrainingInputMode": "File",
        "S3DistributionType": "FullyReplicated",
        "RecordWrapperType": "None"
    }
},
"input_dir": "/opt/ml/input",
"is_master": true,
"job_name": "clsm-rebal-xgboost-regression-model-tra-2023-04-17-02-35-14-214",
"log_level": 20,
"master_hostname": "algo-1",
"model_dir": "/opt/ml/model",
```

```
    "module_dir": "s3://jumpstart-cache-prod-us-east-1/source-directory-tarballs/xgboost/transfer_learning/regressio
n/v1.1.3/sourcedir.tar.gz",
    "module_name": "transfer_learning",
    "network_interface_name": "eth0",
    "num_cpus": 4,
    "num_gpus": 0,
    "output_data_dir": "/opt/ml/output/data",
    "output_dir": "/opt/ml/output",
    "output_intermediate_dir": "/opt/ml/output/intermediate",
    "resource_config": {
        "current_host": "algo-1",
        "current_instance_type": "ml.m5.xlarge",
        "current_group_name": "homogeneousCluster",
        "hosts": [
            "algo-1"
        ],
        "instance_groups": [
            {
                "instance_group_name": "homogeneousCluster",
                "instance_type": "ml.m5.xlarge",
                "hosts": [
                    "algo-1"
                ]
            }
        ],
        "network_interface_name": "eth0"
    },
    "user_entry_point": "transfer_learning.py"
}
Environment variables:
SM_HOSTS=["algo-1"]
SM_NETWORK_INTERFACE_NAME=eth0
SM_HPS={"colsample_bytree":"1","early_stopping_rounds":"30","gamma":"0","learning_rate":"0.3","max_depth":"6","min_c
hild_weight":"1","num_boost_round":"500","reg_alpha":"0","reg_lambda":"3","subsample":"1"}
SM_USER_ENTRY_POINT=transfer_learning.py
SM_FRAMEWORK_PARAMS={}
SM_RESOURCE_CONFIG={"current_group_name":"homogeneousCluster","current_host":"algo-1","current_instance_type":"ml.m
5.xlarge","hosts":["algo-1"],"instance_groups":[{"hosts":["algo-1"],"instance_group_name":"homogeneousCluster","inst
ance_type":"ml.m5.xlarge"}],"network_interface_name":"eth0"}
SM_INPUT_DATA_CONFIG={"model":{"ContentType":"application/x-sagemaker-model","RecordWrapperType":"None","S3Distribut
ionType":"FullyReplicated","TrainingInputMode":"File"},"training":{"RecordWrapperType":"None","S3DistributionType":"
FullyReplicated","TrainingInputMode":"File"},"validation":{"RecordWrapperType":"None","S3DistributionType":"FullyRep
licated","TrainingInputMode":"File"}}
SM_OUTPUT_DATA_DIR=/opt/ml/output/data
```

```
SM_CHANNELS=["model","training","validation"]
SM_CURRENT_HOST=algo-1
SM_MODULE_NAME=transfer_learning
SM_LOG_LEVEL=20
SM_FRAMEWORK_MODULE=sagemaker_xgboost_container.training:main
SM_INPUT_DIR=/opt/ml/input
SM_INPUT_CONFIG_DIR=/opt/ml/input/config
SM_OUTPUT_DIR=/opt/ml/output
SM_NUM_CPUS=4
SM_NUM_GPUS=0
SM_MODEL_DIR=/opt/ml/model
SM_MODULE_DIR=s3://jumpstart-cache-prod-us-east-1/source-directory-tarballs/xgboost/transfer_learning/regression/v1.
1.3/sourcedir.tar.gz
SM_TRAINING_ENV={"additional_framework_parameters":{},"channel_input_dirs":{"model":"/opt/ml/input/data/model","trai
ning":"/opt/ml/input/data/training","validation":"/opt/ml/input/data/validation"},"current_host":"algo-1","framework
_module":"sagemaker_xgboost_container.training:main","hosts":["algo-1"],"hyperparameters":{"colsample_bytree":"1","e
arly_stopping_rounds":"30","gamma":"0","learning_rate":"0.3","max_depth":"6","min_child_weight":"1","num_boost_roun
d":"500","reg_alpha":"0","reg_lambda":"3","subsample":"1"},"input_config_dir":"/opt/ml/input/config","input_data_con
fig":{"model":{"ContentType":"application/x-sagemaker-model","RecordWrapperType":"None","S3DistributionType":"FullyR
eplicated","TrainingInputMode":"File"},"training":{"RecordWrapperType":"None","S3DistributionType":"FullyReplicate
d","TrainingInputMode":"File"},"validation":{"RecordWrapperType":"None","S3DistributionType":"FullyReplicated","Trai
ningInputMode":"File"}},"input_dir":"/opt/ml/input","is_master":true,"job_name":"clsm-rebal-xgboost-regression-model
-tra-2023-04-17-02-35-14-214","log_level":20,"master_hostname":"algo-1","model_dir":"/opt/ml/model","module_dir":"s
3://jumpstart-cache-prod-us-east-1/source-directory-tarballs/xgboost/transfer_learning/regression/v1.1.3/sourcedir.t
ar.gz","module_name":"transfer_learning","network_interface_name":"eth0","num_cpus":4,"num_gpus":0,"output_data_di
r":"/opt/ml/output/data","output_dir":"/opt/ml/output","output_intermediate_dir":"/opt/ml/output/intermediate","reso
urce_config":{"current_group_name":"homogeneousCluster","current_host":"algo-1","current_instance_type":"ml.m5.xlarg
e","hosts":["algo-1"],"instance_groups":[{"hosts":["algo-1"],"instance_group_name":"homogeneousCluster","instance_ty
pe":"ml.m5.xlarge"}],"network_interface_name":"eth0"},"user_entry_point":"transfer_learning.py"}
SM_USER_ARGS=["--colsample_bytree","1","--early_stopping_rounds","30","--gamma","0","--learning_rate","0.3","--max_d
epth","6","--min_child_weight","1","--num_boost_round","500","--reg_alpha","0","--reg_lambda","3","--subsample","1"]
SM_OUTPUT_INTERMEDIATE_DIR=/opt/ml/output/intermediate
SM_CHANNEL_MODEL=/opt/ml/input/data/model
SM_CHANNEL_TRAINING=/opt/ml/input/data/training
SM_CHANNEL_VALIDATION=/opt/ml/input/data/validation
SM_HP_COLSAMPLE_BYTREE=1
SM_HP_EARLY_STOPPING_ROUNDS=30
SM_HP_GAMMA=0
SM_HP_LEARNING_RATE=0.3
SM_HP_MAX_DEPTH=6
SM_HP_MIN_CHILD_WEIGHT=1
SM_HP_NUM_BOOST_ROUND=500
SM_HP_REG_ALPHA=0
SM_HP_REG_LAMBDA=3
```

```
SM_HP_SUBSAMPLE=1
PYTHONPATH=/miniconda3/bin:/:/miniconda3/lib/python/site-packages/xgboost/dmlc-core/tracker:/miniconda3/lib/python3
7.zip:/miniconda3/lib/python3.7:/miniconda3/lib/python3.7/lib-dynload:/miniconda3/lib/python3.7/site-packages
Invoking script with the following command:
/miniconda3/bin/python3 -m transfer_learning --colsample_bytree 1 --early_stopping_rounds 30 --gamma 0 --learning_ra
te 0.3 --max_depth 6 --min_child_weight 1 --num_boost_round 500 --reg_alpha 0 --reg_lambda 3 --subsample 1
INFO:root:Data in the validation channel is found. Reading the train and validation data from the training and valid
ation channel, respectively.
INFO:root:'_input_model_extracted/__models_info__.json' file could not be found.
[0]#011train-rmse:0.39263#011validation-rmse:0.39462
[1]#011train-rmse:0.29895#011validation-rmse:0.29984
[2]#011train-rmse:0.22434#011validation-rmse:0.21613
[3]#011train-rmse:0.18208#011validation-rmse:0.17288
[4]#011train-rmse:0.14231#011validation-rmse:0.13153
[5]#011train-rmse:0.11810#011validation-rmse:0.10434
[6]#011train-rmse:0.09913#011validation-rmse:0.08728
[7]#011train-rmse:0.08668#011validation-rmse:0.07890
[8]#011train-rmse:0.07221#011validation-rmse:0.06390
[9]#011train-rmse:0.06475#011validation-rmse:0.05671
[10]#011train-rmse:0.05479#011validation-rmse:0.05137
[11]#011train-rmse:0.05005#011validation-rmse:0.04686
[12]#011train-rmse:0.04431#011validation-rmse:0.04442
[13]#011train-rmse:0.04052#011validation-rmse:0.04167
[14]#011train-rmse:0.03802#011validation-rmse:0.03883
[15]#011train-rmse:0.03394#011validation-rmse:0.03641
[16]#011train-rmse:0.03004#011validation-rmse:0.03517
[17]#011train-rmse:0.02525#011validation-rmse:0.03111
[18]#011train-rmse:0.02260#011validation-rmse:0.02985
[19]#011train-rmse:0.02099#011validation-rmse:0.02871
[20]#011train-rmse:0.01903#011validation-rmse:0.02821
[21]#011train-rmse:0.01807#011validation-rmse:0.02742
[22]#011train-rmse:0.01631#011validation-rmse:0.02656
[23]#011train-rmse:0.01576#011validation-rmse:0.02548
[24]#011train-rmse:0.01449#011validation-rmse:0.02505
[25]#011train-rmse:0.01406#011validation-rmse:0.02397
[26]#011train-rmse:0.01316#011validation-rmse:0.02315
[27]#011train-rmse:0.01253#011validation-rmse:0.02258
[28]#011train-rmse:0.01188#011validation-rmse:0.02267
[29]#011train-rmse:0.01150#011validation-rmse:0.02240
[30]#011train-rmse:0.01121#011validation-rmse:0.02218
[31]#011train-rmse:0.01075#011validation-rmse:0.02227
[32]#011train-rmse:0.01045#011validation-rmse:0.02227
[33]#011train-rmse:0.00985#011validation-rmse:0.02199
[34]#011train-rmse:0.00950#011validation-rmse:0.02179
```

```
[35]#011train-rmse:0.00913#011validation-rmse:0.02107
[36]#011train-rmse:0.00875#011validation-rmse:0.02109
[37]#011train-rmse:0.00827#011validation-rmse:0.02083
[38]#011train-rmse:0.00811#011validation-rmse:0.02064
[39]#011train-rmse:0.00779#011validation-rmse:0.02071
[40]#011train-rmse:0.00767#011validation-rmse:0.02060
[41]#011train-rmse:0.00742#011validation-rmse:0.02037
[42]#011train-rmse:0.00714#011validation-rmse:0.02019
[43]#011train-rmse:0.00705#011validation-rmse:0.01991
[44]#011train-rmse:0.00681#011validation-rmse:0.01997
[45]#011train-rmse:0.00655#011validation-rmse:0.01985
[46]#011train-rmse:0.00642#011validation-rmse:0.01969
[47]#011train-rmse:0.00613#011validation-rmse:0.01965
[48]#011train-rmse:0.00603#011validation-rmse:0.01951
[49]#011train-rmse:0.00581#011validation-rmse:0.01979
[50]#011train-rmse:0.00576#011validation-rmse:0.01967
[51]#011train-rmse:0.00567#011validation-rmse:0.01957
[52]#011train-rmse:0.00541#011validation-rmse:0.01951
[53]#011train-rmse:0.00535#011validation-rmse:0.01949
[54]#011train-rmse:0.00509#011validation-rmse:0.01938
[55]#011train-rmse:0.00503#011validation-rmse:0.01936
[56]#011train-rmse:0.00481#011validation-rmse:0.01930
[57]#011train-rmse:0.00470#011validation-rmse:0.01921
[58]#011train-rmse:0.00460#011validation-rmse:0.01912
[59]#011train-rmse:0.00445#011validation-rmse:0.01894
[60]#011train-rmse:0.00430#011validation-rmse:0.01876
[61]#011train-rmse:0.00418#011validation-rmse:0.01866
[62]#011train-rmse:0.00408#011validation-rmse:0.01858
[63]#011train-rmse:0.00393#011validation-rmse:0.01840
[64]#011train-rmse:0.00385#011validation-rmse:0.01843
[65]#011train-rmse:0.00375#011validation-rmse:0.01837
[66]#011train-rmse:0.00364#011validation-rmse:0.01831
[67]#011train-rmse:0.00352#011validation-rmse:0.01831
[68]#011train-rmse:0.00345#011validation-rmse:0.01827
[69]#011train-rmse:0.00339#011validation-rmse:0.01827
[70]#011train-rmse:0.00332#011validation-rmse:0.01826
[71]#011train-rmse:0.00328#011validation-rmse:0.01825
[72]#011train-rmse:0.00324#011validation-rmse:0.01824
[73]#011train-rmse:0.00321#011validation-rmse:0.01824
[74]#011train-rmse:0.00314#011validation-rmse:0.01822
[75]#011train-rmse:0.00311#011validation-rmse:0.01823
[76]#011train-rmse:0.00301#011validation-rmse:0.01824
[77]#011train-rmse:0.00295#011validation-rmse:0.01818
[78]#011train-rmse:0.00286#011validation-rmse:0.01827
```

```
[79]#011train-rmse:0.00281#011validation-rmse:0.01822
[80]#011train-rmse:0.00278#011validation-rmse:0.01816
[81]#011train-rmse:0.00274#011validation-rmse:0.01813
[82]#011train-rmse:0.00269#011validation-rmse:0.01808
[83]#011train-rmse:0.00266#011validation-rmse:0.01806
[84]#011train-rmse:0.00264#011validation-rmse:0.01802
[85]#011train-rmse:0.00259#011validation-rmse:0.01803
[86]#011train-rmse:0.00255#011validation-rmse:0.01795
[87]#011train-rmse:0.00252#011validation-rmse:0.01798
[88]#011train-rmse:0.00246#011validation-rmse:0.01803
[89]#011train-rmse:0.00243#011validation-rmse:0.01802
[90]#011train-rmse:0.00238#011validation-rmse:0.01801
[91]#011train-rmse:0.00237#011validation-rmse:0.01801
[92]#011train-rmse:0.00231#011validation-rmse:0.01809
[93]#011train-rmse:0.00226#011validation-rmse:0.01806
[94]#011train-rmse:0.00220#011validation-rmse:0.01811
[95]#011train-rmse:0.00214#011validation-rmse:0.01816
[96]#011train-rmse:0.00208#011validation-rmse:0.01815
[97]#011train-rmse:0.00198#011validation-rmse:0.01797
[98]#011train-rmse:0.00197#011validation-rmse:0.01796
[99]#011train-rmse:0.00192#011validation-rmse:0.01796
[100]#011train-rmse:0.00190#011validation-rmse:0.01796
[101]#011train-rmse:0.00189#011validation-rmse:0.01795
[102]#011train-rmse:0.00187#011validation-rmse:0.01794
[103]#011train-rmse:0.00187#011validation-rmse:0.01794
[104]#011train-rmse:0.00187#011validation-rmse:0.01794
[105]#011train-rmse:0.00187#011validation-rmse:0.01794
[106]#011train-rmse:0.00187#011validation-rmse:0.01794
[107]#011train-rmse:0.00187#011validation-rmse:0.01794
[108]#011train-rmse:0.00187#011validation-rmse:0.01794
[109]#011train-rmse:0.00187#011validation-rmse:0.01794
[110]#011train-rmse:0.00187#011validation-rmse:0.01794
[111]#011train-rmse:0.00187#011validation-rmse:0.01794
[112]#011train-rmse:0.00187#011validation-rmse:0.01794
[113]#011train-rmse:0.00187#011validation-rmse:0.01794
[114]#011train-rmse:0.00187#011validation-rmse:0.01794
[115]#011train-rmse:0.00187#011validation-rmse:0.01794
[116]#011train-rmse:0.00187#011validation-rmse:0.01794
[117]#011train-rmse:0.00187#011validation-rmse:0.01794
[118]#011train-rmse:0.00187#011validation-rmse:0.01794
[119]#011train-rmse:0.00187#011validation-rmse:0.01794
[120]#011train-rmse:0.00187#011validation-rmse:0.01794
[121]#011train-rmse:0.00187#011validation-rmse:0.01794
[122]#011train-rmse:0.00187#011validation-rmse:0.01794
```

```
[123]#011train-rmse:0.00187#011validation-rmse:0.01794
[124]#011train-rmse:0.00187#011validation-rmse:0.01794
[125]#011train-rmse:0.00187#011validation-rmse:0.01794
[126]#011train-rmse:0.00187#011validation-rmse:0.01794
[127]#011train-rmse:0.00187#011validation-rmse:0.01794
[128]#011train-rmse:0.00187#011validation-rmse:0.01794
[129]#011train-rmse:0.00187#011validation-rmse:0.01794
[130]#011train-rmse:0.00187#011validation-rmse:0.01794
[131]#011train-rmse:0.00187#011validation-rmse:0.01794
[132]#011train-rmse:0.00187#011validation-rmse:0.01794
INFO:root:Saving model...
INFO:root:Info file not found at '_input_model_extracted/__models_info__.json'.

2023-04-17 02:37:36 Completed - Training job completed
Training seconds: 78
Billable seconds: 78
```

Deploy and Run Inference on the Trained Tabular Model

In [131…

```
inference_instance_type = "ml.m5.large"

# Retrieve the inference docker container uri
deploy_image_uri = image_uris.retrieve(
    region=None,
    framework=None,
    image_scope="inference",
    model_id=train_model_id,
    model_version=train_model_version,
    instance_type=inference_instance_type,
)
# Retrieve the inference script uri
deploy_source_uri = script_uris.retrieve(
    model_id=train_model_id, model_version=train_model_version, script_scope="inference"
)

endpoint_name = name_from_base(f"clsm-train-{train_model_id}-")

# Use the estimator from the previous step to deploy to a SageMaker endpoint
predictor = tabular_estimator.deploy(
    initial_instance_count=1,
    instance_type=inference_instance_type,
    entry_point="inference.py",
    image_uri=deploy_image_uri,
    source_dir=deploy_source_uri,
    endpoint_name=endpoint_name,
    enable_network_isolation=True,
)
```

```
INFO:sagemaker.image_uris:Ignoring unnecessary Python version: py3.
INFO:sagemaker.image_uris:Ignoring unnecessary instance type: ml.m5.large.
INFO:sagemaker:Creating model with name: sagemaker-jumpstart-2023-04-17-02-38-07-750
INFO:sagemaker:Creating endpoint-config with name clsm-train-xgboost-regression-model--2023-04-17-02-38-07-750
INFO:sagemaker:Creating endpoint with name clsm-train-xgboost-regression-model--2023-04-17-02-38-07-750
----!
```

Test Data

In [132…

```
newline, bold, unbold = "\n", "\033[1m", "\033[0m"

from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
import matplotlib.pyplot as plt

# read the data
test_data = clsm_test
test_data.columns = ["AML_detected"] + [f"Feature_{i}" for i in range(1, test_data.shape[1])]
num_examples, num_columns = test_data.shape
print(
    f"{bold}The test dataset contains {num_examples} examples and {num_columns} columns.{unbold}\n"
)

# prepare the ground truth target and predicting features to send into the endpoint.
ground_truth_label, features = test_data.iloc[:, :1] , test_data.iloc[:, 1:]

print(
    f"{bold}The first 5 observations of the test data: {unbold}"
)  # Feature_1 is the categorical variables and rest of other features are numeric variables.
test_data.head(5)
```

The test dataset contains 68 examples and 20 columns.

The first 5 observations of the test data:

Out[132]:

	AML_detected	Feature_1	Feature_2	Feature_3	Feature_4	Feature_5	Feature_6	Feature_7	Feature_8	Feature_9	Feature_10	Featur
318	1	55.0	51.0	35.0	0	1	1	0	1	0	1	
300	1	22.0	36.0	854.0	1	0	1	0	1	0	1	
478	1	26.7	25.0	637.0	1	0	1	0	0	1	1	
155	1	6.0	10.0	357.0	1	0	1	0	1	0	1	
31	1	75.0	77.5	1735.0	0	1	0	1	1	0	1	

Predict

In [133…

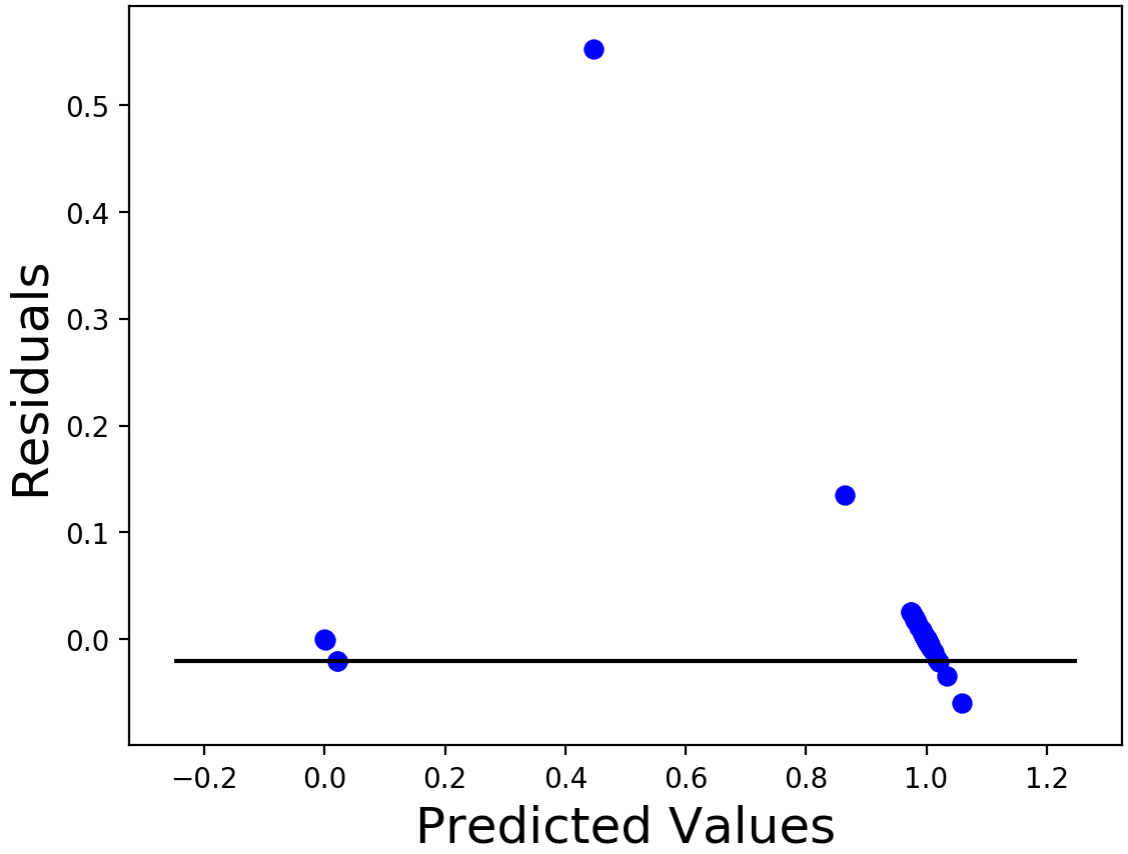
```
content_type = "text/csv"


def query_endpoint(encoded_tabular_data):
    client = boto3.client("runtime.sagemaker")
    response = client.invoke_endpoint(
        EndpointName=endpoint_name, ContentType=content_type, Body=encoded_tabular_data
    )
    return response


def parse_resonse(query_response):
    predictions = json.loads(query_response["Body"].read())
    return np.array(predictions["prediction"])


query_response = query_endpoint(features.to_csv(header=False, index=False).encode("utf-8"))
model_predictions = parse_resonse(query_response)
```

Evaluate Predictions

Visual

In [134…

```
# Visualization: a residual plot to compare the model predictions and ground truth targets.
# Binary results

residuals = ground_truth_label.values[:, 0] - model_predictions
plt.scatter(model_predictions, residuals, color="blue", s=40)
plt.hlines(y=-0.02, xmin=-0.25, xmax=1.25)
plt.xlabel("Predicted Values", fontsize=18)
plt.ylabel("Residuals", fontsize=18)
plt.show()
```



Quantitative

In [135…

```
# Evaluate the model predictions quantitatively.
eval_r2_score = r2_score(ground_truth_label.values, model_predictions)
eval_mse_score = mean_squared_error(ground_truth_label.values, model_predictions)
eval_mae_score = mean_absolute_error(ground_truth_label.values, model_predictions)
print(
    f"{bold}Evaluation result on test data{unbold}:{newline}"
    f"{bold}{r2_score.__name__}{unbold}: {eval_r2_score}{newline}"
    f"{bold}{mean_squared_error.__name__}{unbold}: {eval_mse_score}{newline}"
    f"{bold}{mean_absolute_error.__name__}{unbold}: {eval_mae_score}{newline}"
)
```

```
Evaluation result on test data:
r2_score: 0.9111131505823139
mean_squared_error: 0.0049210712480379825
mean_absolute_error: 0.01730402212791994
```

Delete SageMaker Endpoint

In [136…
```
# Delete the SageMaker endpoint and the attached resources
predictor.delete_model()
predictor.delete_endpoint()
```

```
INFO:sagemaker:Deleting model with name: sagemaker-jumpstart-2023-04-17-02-38-07-750
INFO:sagemaker:Deleting endpoint configuration with name: clsm-train-xgboost-regression-model--2023-04-17-02-38-07-7
50
INFO:sagemaker:Deleting endpoint with name: clsm-train-xgboost-regression-model--2023-04-17-02-38-07-750
```

XGBoost Metrics

In [137…
```
import sklearn.metrics as metrics
from sklearn.metrics import classification_report, roc_curve
```

In [138…
```
# Target values = Test dataset AML Detected

target = ground_truth_label.values[:, 0]
print(target)
```

```
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1]
```

In [139…
```
# Test dataset Predicted values

pred = (model_predictions> 0.5).astype(np.float32)
print(pred)
```

```
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 0. 0. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 0. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 0. 1. 1. 1. 1. 1. 0. 1. 1. 1. 1. 1. 1.]
```

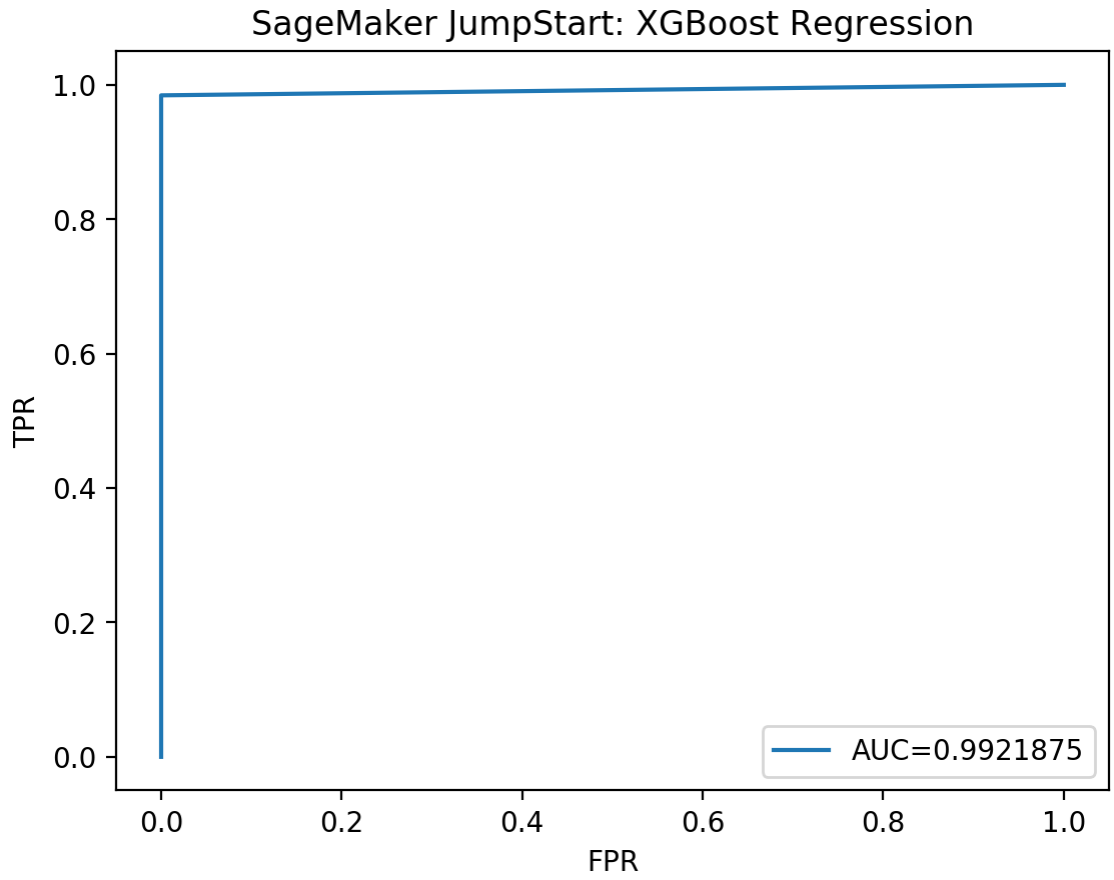
In [140…

```
#Cross Validation

ALM_detected = ['no', 'yes']
print('Cross Validation: \n',
      classification_report(target, pred, target_names=ALM_detected))
```

```
Cross Validation:
               precision    recall  f1-score   support

           no       0.80      1.00      0.89         4
          yes       1.00      0.98      0.99        64

    accuracy                           0.99        68
   macro avg       0.90      0.99      0.94        68
weighted avg       0.99      0.99      0.99        68
```

In [141…

```
#ROC curve for SageMaker JumpStart: XGBoost Regression
fpr, tpr, _ = metrics.roc_curve(target,  pred)
auc = metrics.roc_auc_score(target, pred)

plt.plot(fpr,tpr,label="AUC="+str(auc))
plt.legend(loc=4)
plt.title('SageMaker JumpStart: XGBoost Regression')
plt.xlabel('FPR')
plt.ylabel('TPR')
plt.show()
```



Release Resources

In [142…

```
%%html

<p><b>Shutting down your kernel for this notebook to release resources.</b></p>
<button class="sm-command-button" data-commandlinker-command="kernelmenu:shutdown" style="display:none;">Shutdown Ker

<script>
try {
    els = document.getElementsByClassName("sm-command-button");
    els[0].click();
}
catch(err) {
    // NoOp
}
</script>
```

Shutting down your kernel for this notebook to release resources.

In [143…

```
%%javascript

try {
    Jupyter.notebook.save_checkpoint();
    Jupyter.notebook.session.delete();
}
catch(err) {
    // NoOp
}
```

References

AWS SageMaker. Jumpstart. SGBoost Regression Model. SageMaker Built-In Algorithms: Tabular Regression using XGBoost and Linear Learner

In [ ]:

Logistic Regression approach

Dataset Sources

Beat Acute Myeloid Leukemia (AML) 1.0 was accessed on 13Mar2023 from https://registry.opendata.aws/beataml. OHSU BeatAML Datasets Link: https://ctd2-data.nci.nih.gov/Public/OHSU-1/BeatAML_Waves1_2/

OpenCell Datasets Link: https://opencell.czbiohub.org/download

Check Pre-requisites from the 01-setup Folder

In [4]:
```
%store -r setup_instance_check_passed
```

In [5]:
```
try:
    setup_instance_check_passed
except NameError:
    print("++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Instance Check.")
    print("++++++++++++++++++++++++++++++++")
```

In [6]:
```
print(setup_instance_check_passed)
```

```
True
```

In [7]:
```
%store -r setup_dependencies_passed
```

In [8]:
```
try:
    setup_dependencies_passed
except NameError:
    print("++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Setup Dependencies.")
    print("++++++++++++++++++++++++++++++++")
```

In [9]:
```
print(setup_dependencies_passed)
```

```
True
```

In [10]:
```
%store -r setup_s3_bucket_passed
```

In [11]:
```
try:
    setup_s3_bucket_passed
except NameError:
    print("++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Setup S3 Bucket.")
    print("++++++++++++++++++++++++++++++++")
```

In [12]:
```
print(setup_s3_bucket_passed)
```

```
True
```

In [13]:
```
%store -r setup_iam_roles_passed
```

In [14]:
```
try:
    setup_iam_roles_passed
except NameError:
    print("++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Setup IAM Roles.")
    print("++++++++++++++++++++++++++++++++")
```

In [15]:
```
print(setup_iam_roles_passed)
```

```
True
```

In [16]:
```
if not setup_instance_check_passed:
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Instance Check.")
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
if not setup_dependencies_passed:
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Setup Dependencies.")
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
if not setup_s3_bucket_passed:
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Setup S3 Bucket.")
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
if not setup_iam_roles_passed:
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Setup IAM Roles.")
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
```

In [17]:
```
import boto3
import sagemaker
import pandas as pd
import time
from time import gmtime, strftime

sess = sagemaker.Session()
role = sagemaker.get_execution_role()
bucket = sess.default_bucket()
region = boto3.Session().region_name
account_id = boto3.client("sts").get_caller_identity().get("Account")

sm = boto3.Session().client(service_name="sagemaker", region_name=region)
```

Data Cleaning

Import Tools:

In [127…
```
!pip install klib
```

```
Requirement already satisfied: klib in /opt/conda/lib/python3.7/site-packages (1.0.1)
Requirement already satisfied: Jinja2<4.0.0,>=3.0.3 in /opt/conda/lib/python3.7/site-packages (from klib) (3.1.2)
Requirement already satisfied: matplotlib<4.0.0,>=3.0.3 in /opt/conda/lib/python3.7/site-packages (from klib) (3.1.
3)
Requirement already satisfied: numpy<2.0.0,>=1.16.3 in /opt/conda/lib/python3.7/site-packages (from klib) (1.18.1)
Requirement already satisfied: pandas<2.0.0,>=1.1.2 in /opt/conda/lib/python3.7/site-packages (from klib) (1.3.5)
Requirement already satisfied: scipy<2.0.0,>=1.1.0 in /opt/conda/lib/python3.7/site-packages (from klib) (1.4.1)
Requirement already satisfied: seaborn<0.12.0,>=0.11.1 in /opt/conda/lib/python3.7/site-packages (from klib) (0.11.
2)
Requirement already satisfied: MarkupSafe>=2.0 in /opt/conda/lib/python3.7/site-packages (from Jinja2<4.0.0,>=3.0.3-
>klib) (2.1.2)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.7/site-packages (from matplotlib<4.0.0,>=3.0.3
->klib) (0.10.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /opt/conda/lib/python3.7/site-packages (from matplotlib<4.0.0,>=
3.0.3->klib) (1.1.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /opt/conda/lib/python3.7/site-packages (f
rom matplotlib<4.0.0,>=3.0.3->klib) (2.4.6)
Requirement already satisfied: python-dateutil>=2.1 in /opt/conda/lib/python3.7/site-packages (from matplotlib<4.0.
0,>=3.0.3->klib) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/lib/python3.7/site-packages (from pandas<2.0.0,>=1.1.2->kl
ib) (2019.3)
Requirement already satisfied: six in /opt/conda/lib/python3.7/site-packages (from cycler>=0.10->matplotlib<4.0.0,>=
3.0.3->klib) (1.14.0)
Requirement already satisfied: setuptools in /opt/conda/lib/python3.7/site-packages (from kiwisolver>=1.0.1->matplot
lib<4.0.0,>=3.0.3->klib) (45.2.0.post20200210)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system p
ackage manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
```

In [42]:
```
import numpy as np
import seaborn as sns
import klib
import matplotlib.pyplot as plt

%matplotlib inline
%config InlineBackend.figure_format='retina'
```

BeatAML Clinical Summary

OHSU BeatAML Clinical Summary Table

Download & Analyze data sets

In [44]:
```
import numpy as np
import seaborn as sns
import klib
import matplotlib.pyplot as plt
import boto3
import pandas as pd
import matplotlib.pyplot as plt
import json
import warnings
warnings.filterwarnings('ignore')

%matplotlib inline
%config InlineBackend.figure_format='retina'
```

In [45]:
```
!aws s3 cp 's3://ads508rawdatasets/OHSU_BeatAMLWaves1_2_Tyner_ClinicalSummary.csv' ./data/
```

```
download: s3://ads508rawdatasets/OHSU_BeatAMLWaves1_2_Tyner_ClinicalSummary.csv to data/OHSU_BeatAMLWaves1_2_Tyner_C
linicalSummary.csv
```

In [46]:
```
!aws s3 cp 's3://ads508rawdatasets/opencell-protein-interactions.csv' ./data/
```

```
download: s3://ads508rawdatasets/opencell-protein-interactions.csv to data/opencell-protein-interactions.csv
```

In [47]:
```
import csv

clsm = pd.read_csv('s3://ads508rawdatasets/OHSU_BeatAMLWaves1_2_Tyner_ClinicalSummary.csv')
clsm.head(5)
```

Out[47]:

	LabId	PatientId	consensus_sex	inferred_sex	inferred_ethnicity	centerID	CEBPA_Biallelic	ageAtDiagnosis	isRelapse	isDenovo	...
0	09-00705	163	Male	Male	White	1	n	73.0	False	True	...
1	10-00136	174	Male	Male	White	1	n	69.0	False	True	...
2	10-00172	175	Female	Male	White	1	n	59.0	False	True	...
3	10-00507	45	Female	Female	White	1	n	70.0	False	True	...
4	10-00542	174	Male	Male	White	1	n	69.0	True	False	...

5 rows × 159 columns

In [48]:
```
pi = pd.read_csv('s3://ads508rawdatasets/opencell-protein-interactions.csv')
pi.head(5)
```

Out[48]:

	target_gene_name	interactor_gene_name	target_ensg_id	interactor_ensg_id	interactor_uniprot_ids	
0	AAMP	ARGLU1	ENSG00000127837	ENSG00000134884	Q9NWB6;Q9NWB6-3;Q9NWB6-2	5
1	AAMP	CWF19L2	ENSG00000127837	ENSG00000152404	Q2TBE0;Q2TBE0-2;H7C3G7;Q2TBE0-3;H0YE03	5
2	AAMP	PRPF40A	ENSG00000127837	ENSG00000196504	A0A3F2YNY6;O75400-2;O75400-3;O75400;H0YG38;F5H578	5
3	AAMP	RPL10	ENSG00000127837	ENSG00000147403	X1WI28;P27635;B8A6G2;A6QRI9;Q96L21	15
4	AAMP	RSRC1	ENSG00000127837	ENSG00000174891	Q96IZ7-2;Q96IZ7;H7C5Q0;C9J713;C9J367;C9J8Q2;C9...	5

In [ ]:
```
```

Display clsm data set

In [49]:
```
clsm
```

Out[49]:

	LabId	PatientId	consensus_sex	inferred_sex	inferred_ethnicity	centerID	CEBPA_Biallelic	ageAtDiagnosis	isRelapse	isDenovo	..
0	09-00705	163	Male	Male	White	1	n	73.0	False	True	.
1	10-00136	174	Male	Male	White	1	n	69.0	False	True	.
2	10-00172	175	Female	Male	White	1	n	59.0	False	True	.
3	10-00507	45	Female	Female	White	1	n	70.0	False	True	.
4	10-00542	174	Male	Male	White	1	n	69.0	True	False	.
...	...	...	...	...	...	...	...	...	...	...	.
667	17-00072	4366	Male	Male	White	1	n	70.0	False	False	.
668	17-00077	4317	Female	Female	White	1	n	72.0	False	False	.
669	17-00093	4379	Female	Female	Black	2	n	43.0	False	False	.
670	17-00094	4380	Male	Male	White	6	n	57.0	False	False	.
671	17-00096	2747	Male	Male	White	6	n	62.0	False	False	.

672 rows × 159 columns

In [50]:
```
clsm.shape
```

Out[50]:
```
(672, 159)
```

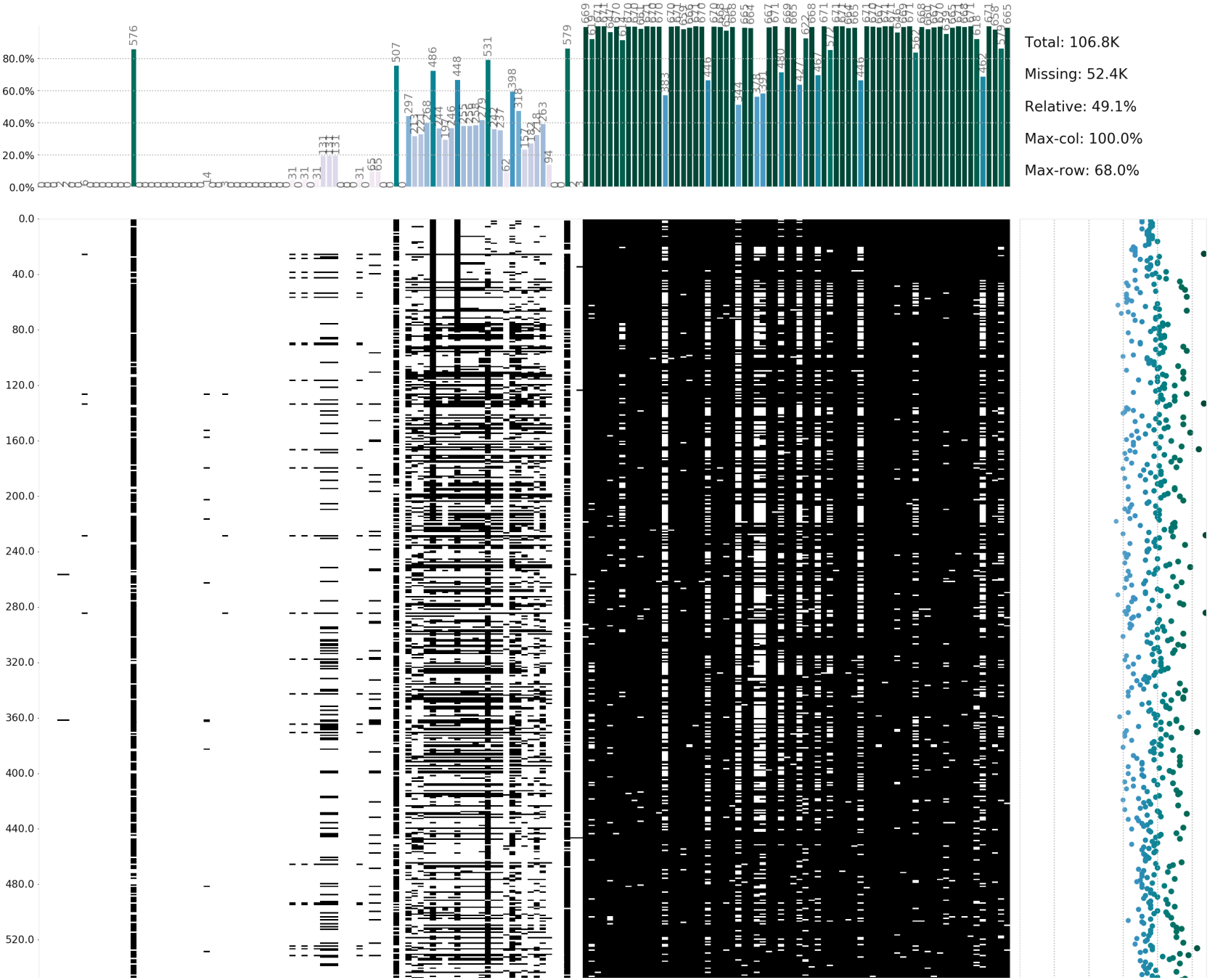
In [51]:
```
clsm = clsm.replace('', np.NAN)
clsm.info()
```

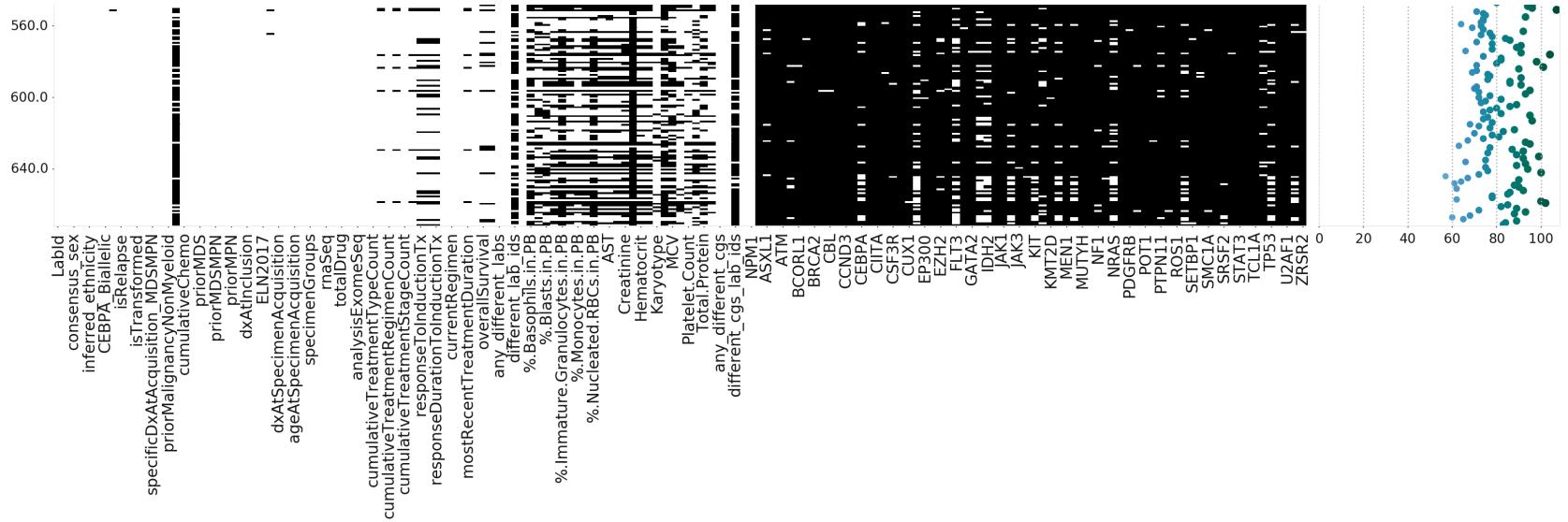
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 672 entries, 0 to 671
Columns: 159 entries, LabId to ZRSR2
dtypes: bool(9), float64(22), int64(7), object(121)
memory usage: 793.5+ KB
```

In [52]:
```
klib.missingval_plot(clsm)
```

Out[52]:
```
GridSpec(6, 6)
```

Missing value plot





Create new dataframe to retain relevant features for further use

In [53]:
```
clsm_cut = pd.DataFrame(clsm[['LabId', 'PatientId', 'consensus_sex', 'inferred_ethnicity', 'isRelapse',
                              'isTransformed', 'priorMalignancyNonMyeloid', 'priorMDS', 'priorMDSMPN', 'priorMPN',
                              'ELN2017', 'dxAtSpecimenAcquisition', 'vitalStatus', 'overallSurvival', '%.Blasts.in.BM
                               '%.Blasts.in.PB', 'FLT3-ITD', 'NPM1']])

clsm_cut
```

Out[53]:

	LabId	PatientId	consensus_sex	inferred_ethnicity	isRelapse	isTransformed	priorMalignancyNonMyeloid	priorMDS	priorMDSMP
0	09-00705	163	Male	White	False	False	n	n	
1	10-00136	174	Male	White	False	False	n	n	
2	10-00172	175	Female	White	False	False	n	n	
3	10-00507	45	Female	White	False	False	n	n	
4	10-00542	174	Male	White	True	False	n	n	
...	...	...	...	...	...	...	...	...	...
667	17-00072	4366	Male	White	False	True	n	n	
668	17-00077	4317	Female	White	False	False	n	n	
669	17-00093	4379	Female	Black	False	True	n	n	
670	17-00094	4380	Male	White	False	True	n	n	
671	17-00096	2747	Male	White	False	True	n	n	

672 rows × 18 columns

In [54]:
```
clsm_cut.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 672 entries, 0 to 671
Data columns (total 18 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   LabId                       672 non-null    object
 1   PatientId                   672 non-null    int64
 2   consensus_sex               672 non-null    object
 3   inferred_ethnicity          670 non-null    object
 4   isRelapse                   672 non-null    bool
 5   isTransformed               672 non-null    bool
 6   priorMalignancyNonMyeloid   672 non-null    object
 7   priorMDS                    672 non-null    object
 8   priorMDSMPN                 672 non-null    object
 9   priorMPN                    672 non-null    object
 10  ELN2017                     672 non-null    object
 11  dxAtSpecimenAcquisition     672 non-null    object
 12  vitalStatus                 672 non-null    object
 13  overallSurvival             607 non-null    float64
 14  %.Blasts.in.BM              459 non-null    object
 15  %.Blasts.in.PB              451 non-null    object
 16  FLT3-ITD                    670 non-null    object
 17  NPM1                        669 non-null    object
dtypes: bool(2), float64(1), int64(1), object(14)
memory usage: 85.4+ KB
```

In [55]:
```
clsm_cut.describe()
```

Out[55]:

	PatientId	overallSurvival
count	672.000000	607.000000
mean	2088.020833	441.881384
std	973.372734	479.180429
min	17.000000	-1.000000
25%	1450.750000	167.000000
50%	2016.000000	323.000000
75%	2501.500000	555.000000
max	4380.000000	5305.000000

Attribute Information

% Blasts Attributes Numerical Prep

%.blasts.in.bm Attribute:

In [56]:
```
#Attribute Transformation - %.Blasts.in.BM'
#Identify unique values in %.Blasts.in.BM'
clsm_cut['%.Blasts.in.BM'].unique()
```

Out[56]:
```
array(['94', '80', '91', '97', '87', nan, '40', '75', '83', '95', '85',
       '90', '70', '92', '72', '68', '88', '36', '81', '93', '34', '77.5',
       '46', '65', '50', '76', '71', '60', '73', '55', '0.5', '30', '62',
       '18', '82', '28', '41', '64', '84', '21', '51', '17', '49.4', '32',
       '29', '25', '59.3', '66', '20', '52', '54', '22', '10', '12', '13',
       '67', '39', '25.9', '45', '37', '78', '8', '3', '54.8', '74', '96',
       '4', '86.1', '42', '56', '69', '79', '33', '9', '0.4', '51.5',
       '15', '5', '24', '7', '2', '6', '1', '58', '>50', '35', '86',
       '93.2', '0', '27', '89.6', '23', '98', '19', '91.8', '>95', '57',
       '71.5', '78.3', '63', '1.5', '53.74', '59.5', '44', '42.5', '26',
       '3.5', '48', '26.3', '47', '88.5'], dtype=object)
```

In [57]:
```
# > and < will be changed to whole numbers less than or greater than.
clsm_cut['%.Blasts.in.BM'] = clsm_cut['%.Blasts.in.BM'].replace(['>50'], 51)
clsm_cut['%.Blasts.in.BM'] = clsm_cut['%.Blasts.in.BM'].replace(['>95'], 96)

clsm_cut['%.Blasts.in.BM'].unique()
```

Out[57]:
```
array(['94', '80', '91', '97', '87', nan, '40', '75', '83', '95', '85',
       '90', '70', '92', '72', '68', '88', '36', '81', '93', '34', '77.5',
       '46', '65', '50', '76', '71', '60', '73', '55', '0.5', '30', '62',
       '18', '82', '28', '41', '64', '84', '21', '51', '17', '49.4', '32',
       '29', '25', '59.3', '66', '20', '52', '54', '22', '10', '12', '13',
       '67', '39', '25.9', '45', '37', '78', '8', '3', '54.8', '74', '96',
       '4', '86.1', '42', '56', '69', '79', '33', '9', '0.4', '51.5',
       '15', '5', '24', '7', '2', '6', '1', '58', 51, '35', '86', '93.2',
       '0', '27', '89.6', '23', '98', '19', '91.8', 96, '57', '71.5',
       '78.3', '63', '1.5', '53.74', '59.5', '44', '42.5', '26', '3.5',
       '48', '26.3', '47', '88.5'], dtype=object)
```

In [58]:
```
#Attribute Transformation - %.Blasts.in.PB'
#Identify unique values in %.Blasts.in.PB'
clsm_cut['%.Blasts.in.PB'].unique()
```

Out[58]:
```
array(['97', '19', '99', '80', nan, '51', '30', '41', '84', '77', '75',
       '63', '60', '96', '66', '45', '93', '9', '82', '15', '33', '0',
       '13', '94', '89', '83', '>90', '78', '72', '59', '32', '6', '29',
       '24', '64', '57', '52', '2.1', '<5', '17', '22', '5', '47', '56',
       '25', '23', '42', '65', '71', '8', '3.5', '66.3', '95', '44', '10',
       '28.6', '18', '58', '67', '40', '92', '54', '1', '2', '20', '28',
       '35', '85', '42.4', '16', '49.1', '14', '88', '46', '7', '0.5',
       '79', '26', '87', '20.4', '68', '48', '5.3', '61', '90', '17.4',
       '57.4', '43.8', '50', '37', '4', '3', '12', '81', '11', '90.5',
       '"rare"', '90.2', '55', 'rare', '39', '31', '86', '47.4', '27.4',
       '39.6', '12.9', '15.4', '9.5', '62', '64.6', '27.8', '69.14',
       '52.2', '91', '67.25', '49', '23.7', '48.6', '98', '74.8', '2.6',
       '43', '29.6', '47.5', '38', '2.5', '25.2', '3.56', '70', '99.2',
       '73', '26.7', '38.5', '7.7', '74', '93.3', '12.1', '11.2', '92.9',
       '98.4', '6.8', '10.5', '53', '3.1', '28.9', '72.9', '40.2', '3.3',
       '42.1', '11.5', '77.8', '3.8', '59.5', '21.7', '53.2'],
      dtype=object)
```

In [59]:
```
#%.Blasts.in.PB attribute has 1 "rare" and 1 'rare' record with no flt3 nor npm1 input. This will be changed to NAN
clsm_cut['%.Blasts.in.PB'] = clsm_cut['%.Blasts.in.PB'].replace(['"rare"'], np.nan)
clsm_cut['%.Blasts.in.PB'] = clsm_cut['%.Blasts.in.PB'].replace(['rare'], np.nan)
# > and < will be changed to whole numbers less than or greater than.
clsm_cut['%.Blasts.in.PB'] = clsm_cut['%.Blasts.in.PB'].replace(['<5'], 4)
clsm_cut['%.Blasts.in.PB'] = clsm_cut['%.Blasts.in.PB'].replace(['>90'], 91)

clsm_cut['%.Blasts.in.PB'].unique()
```

Out[59]:
```
array(['97', '19', '99', '80', nan, '51', '30', '41', '84', '77', '75',
       '63', '60', '96', '66', '45', '93', '9', '82', '15', '33', '0',
       '13', '94', '89', '83', 91, '78', '72', '59', '32', '6', '29',
       '24', '64', '57', '52', '2.1', 4, '17', '22', '5', '47', '56',
       '25', '23', '42', '65', '71', '8', '3.5', '66.3', '95', '44', '10',
       '28.6', '18', '58', '67', '40', '92', '54', '1', '2', '20', '28',
       '35', '85', '42.4', '16', '49.1', '14', '88', '46', '7', '0.5',
       '79', '26', '87', '20.4', '68', '48', '5.3', '61', '90', '17.4',
       '57.4', '43.8', '50', '37', '4', '3', '12', '81', '11', '90.5',
       '90.2', '55', '39', '31', '86', '47.4', '27.4', '39.6', '12.9',
       '15.4', '9.5', '62', '64.6', '27.8', '69.14', '52.2', '91',
       '67.25', '49', '23.7', '48.6', '98', '74.8', '2.6', '43', '29.6',
       '47.5', '38', '2.5', '25.2', '3.56', '70', '99.2', '73', '26.7',
       '38.5', '7.7', '74', '93.3', '12.1', '11.2', '92.9', '98.4', '6.8',
       '10.5', '53', '3.1', '28.9', '72.9', '40.2', '3.3', '42.1', '11.5',
       '77.8', '3.8', '59.5', '21.7', '53.2'], dtype=object)
```

From Categorical to Numerical

Transform &.blasts.in.bm and %.blasts.in.pb from object to float:

In [60]:
```
clsm_cut['%.Blasts.in.BM'] = clsm_cut['%.Blasts.in.BM'].astype(float)
clsm_cut['%.Blasts.in.PB'] = clsm_cut['%.Blasts.in.PB'].astype(float)
```

In [61]:
```
clsm_cut.info()
```

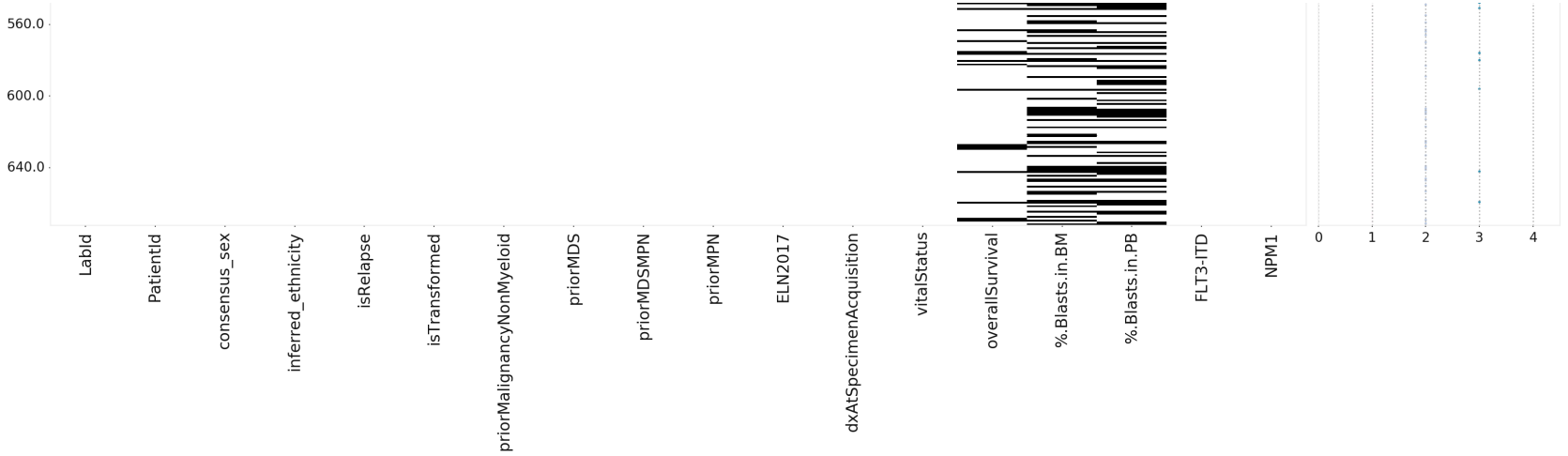
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 672 entries, 0 to 671
Data columns (total 18 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   LabId                       672 non-null    object
 1   PatientId                   672 non-null    int64
 2   consensus_sex               672 non-null    object
 3   inferred_ethnicity          670 non-null    object
 4   isRelapse                   672 non-null    bool
 5   isTransformed               672 non-null    bool
 6   priorMalignancyNonMyeloid   672 non-null    object
 7   priorMDS                    672 non-null    object
 8   priorMDSMPN                 672 non-null    object
 9   priorMPN                    672 non-null    object
 10  ELN2017                     672 non-null    object
 11  dxAtSpecimenAcquisition     672 non-null    object
 12  vitalStatus                 672 non-null    object
 13  overallSurvival             607 non-null    float64
 14  %.Blasts.in.BM              459 non-null    float64
 15  %.Blasts.in.PB              448 non-null    float64
 16  FLT3-ITD                    670 non-null    object
 17  NPM1                        669 non-null    object
dtypes: bool(2), float64(3), int64(1), object(12)
memory usage: 85.4+ KB
```

clsm_cut Identify Missing Values

In [62]:
```
klib.missingval_plot(clsm_cut)
```

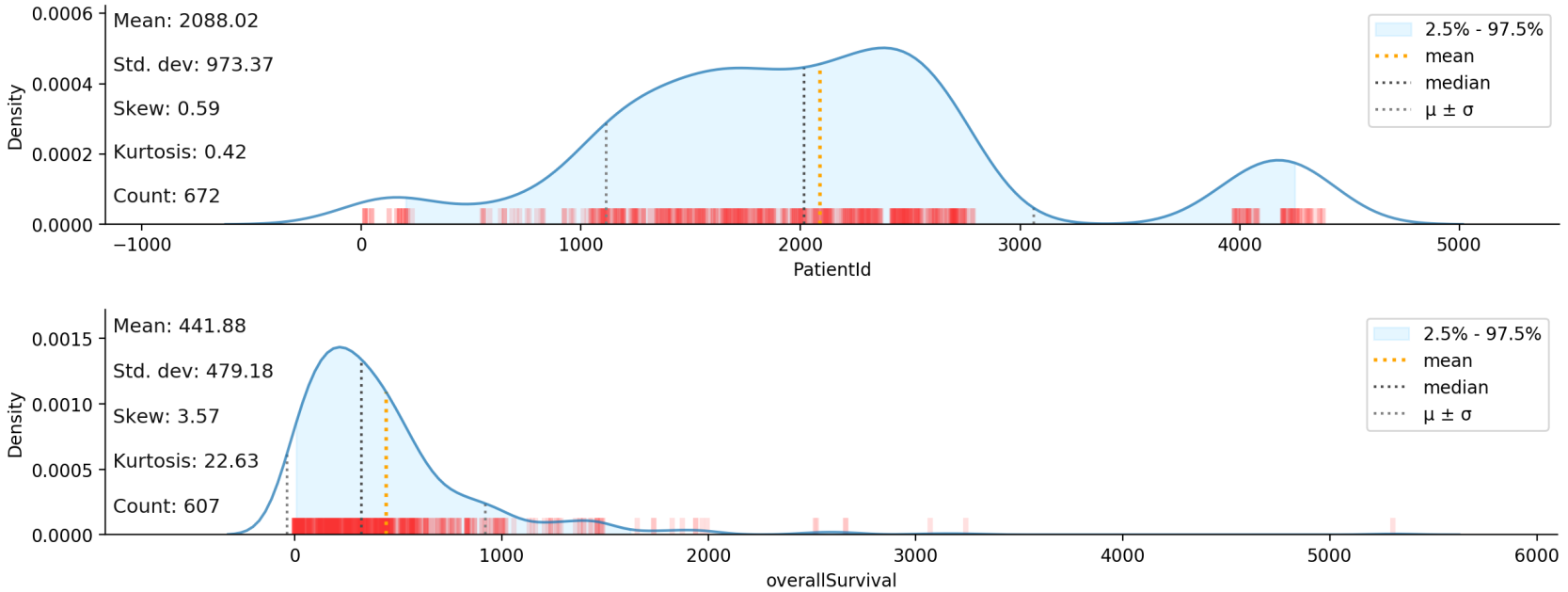
Out[62]:
```
GridSpec(6, 6)
```

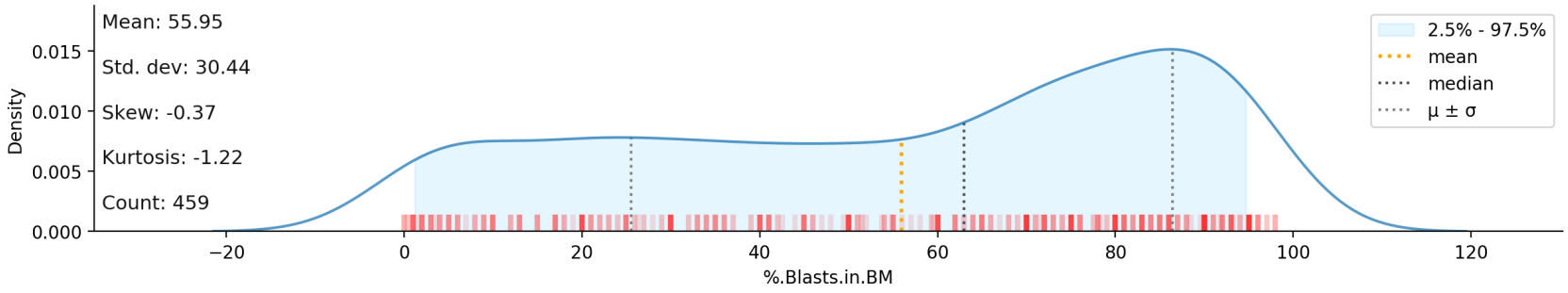
Missing value plot

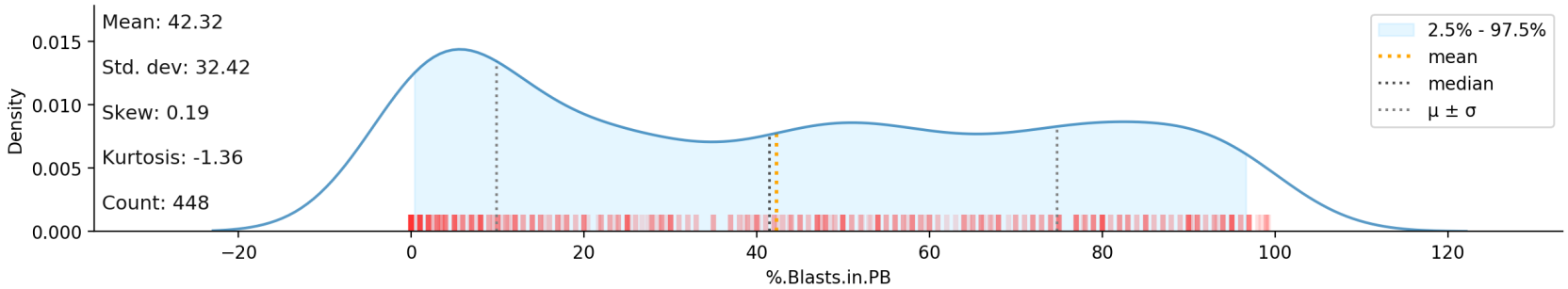


In [63]:
```
#Replace Missing Value
klib.dist_plot(clsm_cut)
```

Out[63]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb8a235cd10>
```







In [64]:
```
clsm_cut.describe()
```

Out[64]:

	PatientId	overallSurvival	%.Blasts.in.BM	%.Blasts.in.PB
count	672.000000	607.000000	459.000000	448.000000
mean	2088.020833	441.881384	55.949325	42.316629
std	973.372734	479.180429	30.440925	32.418249
min	17.000000	-1.000000	0.000000	0.000000
25%	1450.750000	167.000000	30.000000	10.000000
50%	2016.000000	323.000000	63.000000	41.500000
75%	2501.500000	555.000000	83.000000	72.000000
max	4380.000000	5305.000000	98.000000	99.200000

In [65]:
```
#From distibution, skewness suggest median is the best representation.
clsm_cut['overallSurvival'] = clsm_cut['overallSurvival'].fillna(clsm_cut['overallSurvival'].median())
clsm_cut['%.Blasts.in.BM'] = clsm_cut['%.Blasts.in.BM'].fillna(clsm_cut['%.Blasts.in.BM'].median())
clsm_cut['%.Blasts.in.PB'] = clsm_cut['%.Blasts.in.PB'].fillna(clsm_cut['%.Blasts.in.PB'].median())
```

In [66]:
```
#Replace categorical NaN with unknown
clsm_cut = clsm_cut.replace(np.nan, 'unknown', regex=True)
```

In [67]:
```
#Determine mode of inferred_ethnicity
clsm_cut['inferred_ethnicity'].mode()
```

Out[67]:
```
0    White
dtype: object
```

In [68]:
```
#In inferred_ethnicity, replace mode of unknown to white:
clsm_cut['inferred_ethnicity'] = clsm_cut['inferred_ethnicity'].replace(['unknown'], 'white')

clsm_cut['inferred_ethnicity'].unique()
```

Out[68]:
```
array(['White', 'HispNative', 'AdmixedBlack', 'Asian', 'Black',
       'AdmixedAsian', 'white', 'AdmixedWhite', 'AdmixedHispNative'],
      dtype=object)
```

In [69]:
```
#Determine mode of flt3-itd
clsm_cut['FLT3-ITD'].mode()
```

Out[69]:
```
0    negative
dtype: object
```

In [70]:
```
#In flt3-itd, replace mode of unknown to negative:
clsm_cut['FLT3-ITD'] = clsm_cut['FLT3-ITD'].replace(['unknown'], 'negative')
clsm_cut['FLT3-ITD'].unique()
```

Out[70]:
```
array(['positive', 'negative'], dtype=object)
```

In [71]:
```
clsm_cut['NPM1'].mode()
```

Out[71]:
```
0    negative
dtype: object
```

In [72]:
```
#In npm1, replace mode of unknown to negative:
clsm_cut['NPM1'] = clsm_cut['NPM1'].replace(['unknown'], 'negative')
clsm_cut['NPM1'].unique()
```

Out[72]:
```
array(['positive', 'negative'], dtype=object)
```

In [73]:
```
#Check for missing values
klib.missingval_plot(clsm_cut)
```

```
No missing values found in the dataset.
```

In [74]:
```
clsm_cut.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 672 entries, 0 to 671
Data columns (total 18 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   LabId                      672 non-null    object 
 1   PatientId                  672 non-null    int64  
 2   consensus_sex              672 non-null    object 
 3   inferred_ethnicity         672 non-null    object 
 4   isRelapse                  672 non-null    bool   
 5   isTransformed              672 non-null    bool   
 6   priorMalignancyNonMyeloid  672 non-null    object 
 7   priorMDS                   672 non-null    object 
 8   priorMDSMPN                672 non-null    object 
 9   priorMPN                   672 non-null    object 
 10  ELN2017                    672 non-null    object 
 11  dxAtSpecimenAcquisition    672 non-null    object 
 12  vitalStatus                672 non-null    object 
 13  overallSurvival            672 non-null    float64
 14  %.Blasts.in.BM             672 non-null    float64
 15  %.Blasts.in.PB             672 non-null    float64
 16  FLT3-ITD                   672 non-null    object 
 17  NPM1                       672 non-null    object 
dtypes: bool(2), float64(3), int64(1), object(12)
memory usage: 85.4+ KB
```

Check for Duplicates

In [75]:
```
#Remove duplicated columns
clsm_cut = clsm_cut.drop_duplicates(ignore_index=True)
clsm_cut.info()
```

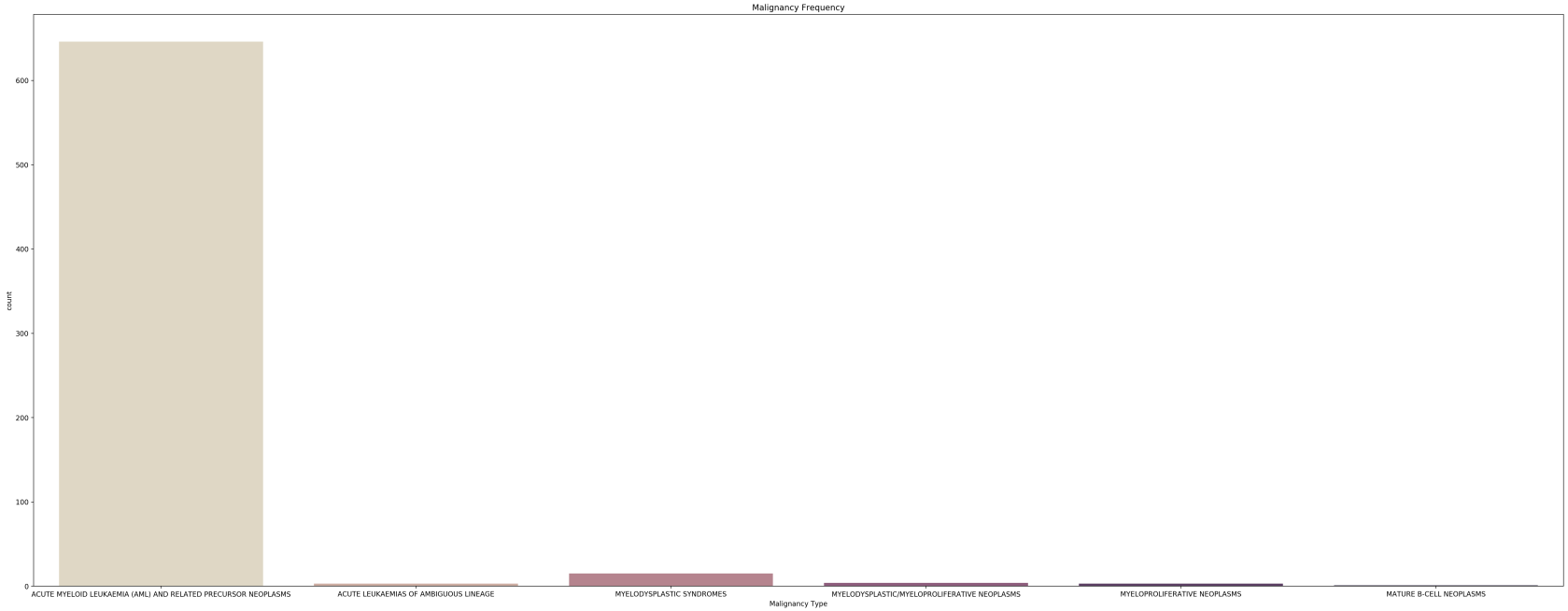
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 672 entries, 0 to 671
Data columns (total 18 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   LabId                       672 non-null    object
 1   PatientId                   672 non-null    int64
 2   consensus_sex               672 non-null    object
 3   inferred_ethnicity          672 non-null    object
 4   isRelapse                   672 non-null    bool
 5   isTransformed               672 non-null    bool
 6   priorMalignancyNonMyeloid   672 non-null    object
 7   priorMDS                    672 non-null    object
 8   priorMDSMPN                 672 non-null    object
 9   priorMPN                    672 non-null    object
 10  ELN2017                     672 non-null    object
 11  dxAtSpecimenAcquisition     672 non-null    object
 12  vitalStatus                 672 non-null    object
 13  overallSurvival             672 non-null    float64
 14  %.Blasts.in.BM              672 non-null    float64
 15  %.Blasts.in.PB              672 non-null    float64
 16  FLT3-ITD                    672 non-null    object
 17  NPM1                        672 non-null    object
dtypes: bool(2), float64(3), int64(1), object(12)
memory usage: 85.4+ KB
```

In [76]:
```
clsm_cut.duplicated().sum()
```

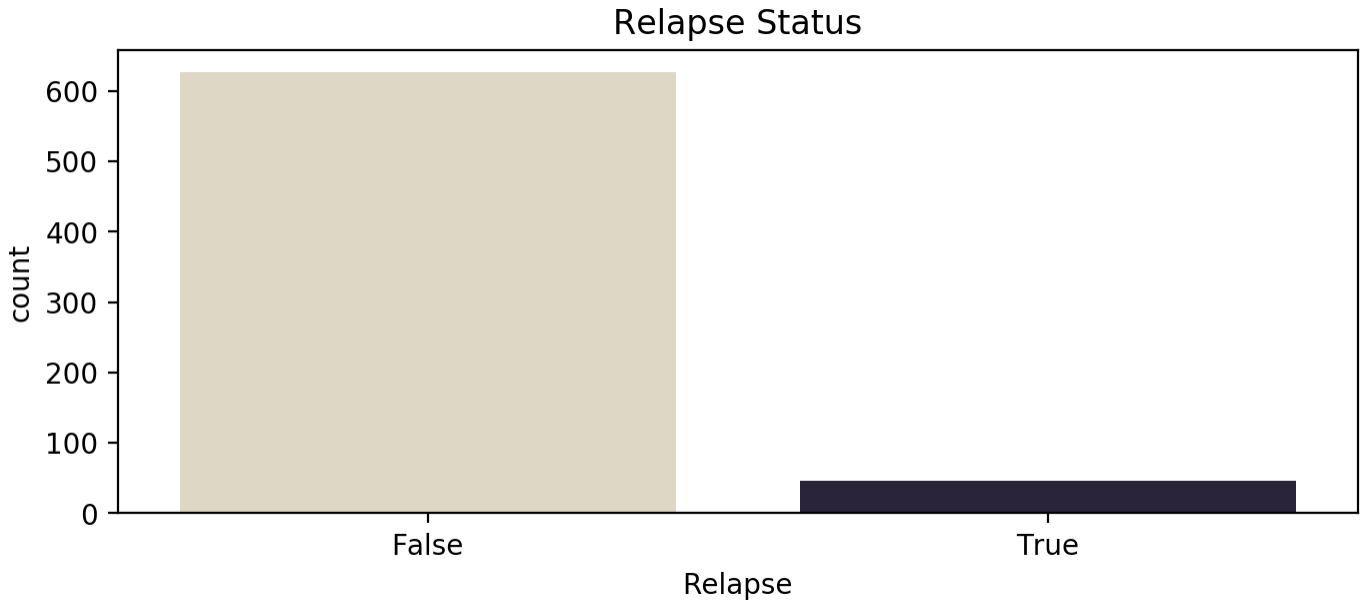
Out[76]:
```
0
```

Data Exploration

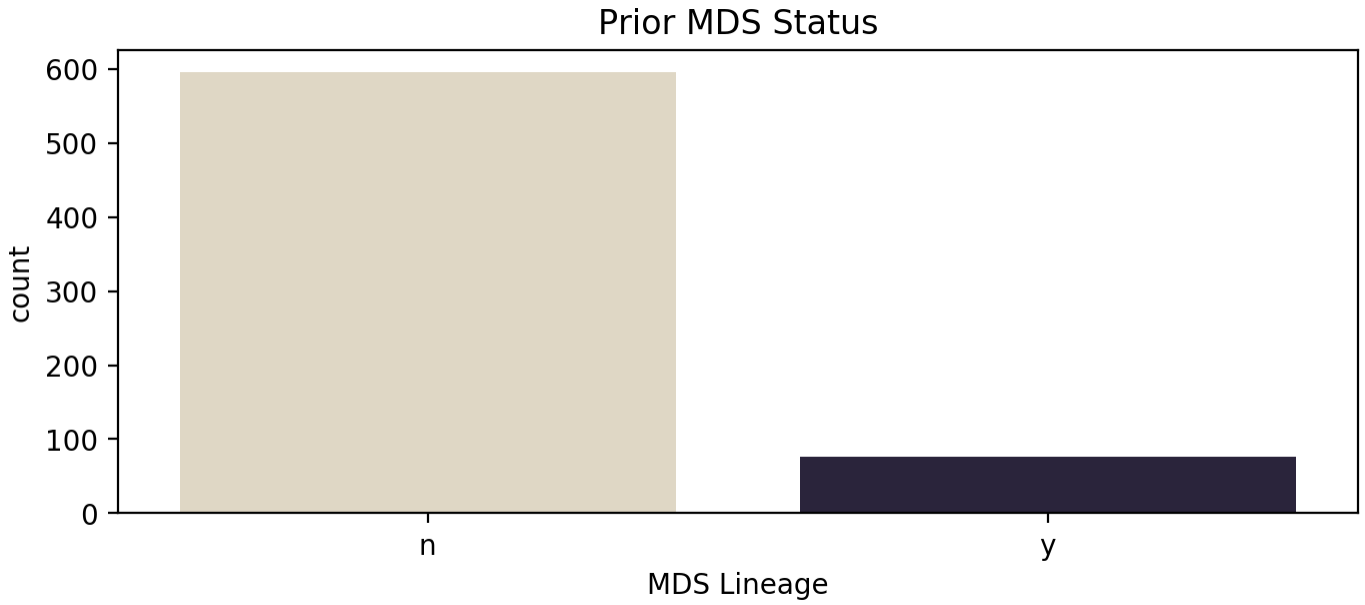
In [77]:
```
#Data Visualization - "dxAtSpecimenAcquisition
sns.countplot(x=clsm_cut["dxAtSpecimenAcquisition"], palette = "ch:s=-.2,r=.6")
plt.xlabel('Malignancy Type')
plt.title('Malignancy Frequency')
plt.gcf().set_size_inches(40, 15)
```



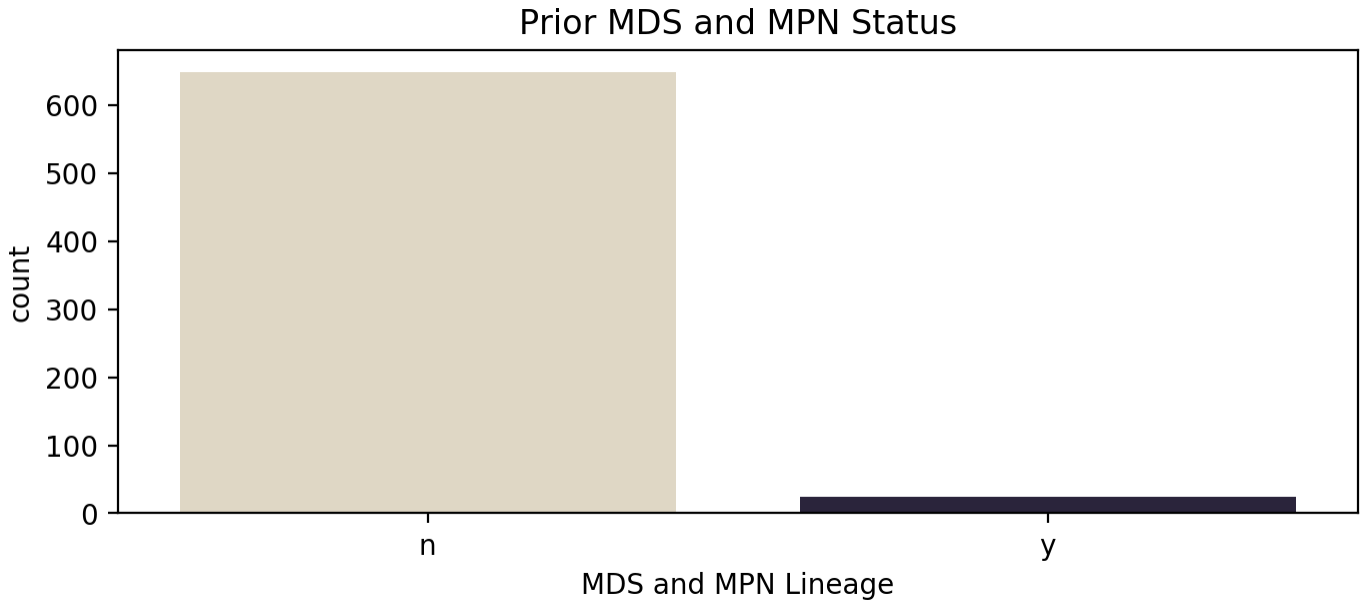
In [78]:
```
#Data Visualization - "isRelapse"
sns.countplot(x=clsm_cut["isRelapse"], palette = "ch:s=-.2,r=.6")
plt.xlabel('Relapse')
plt.title('Relapse Status')
plt.gcf().set_size_inches(8, 3)
```



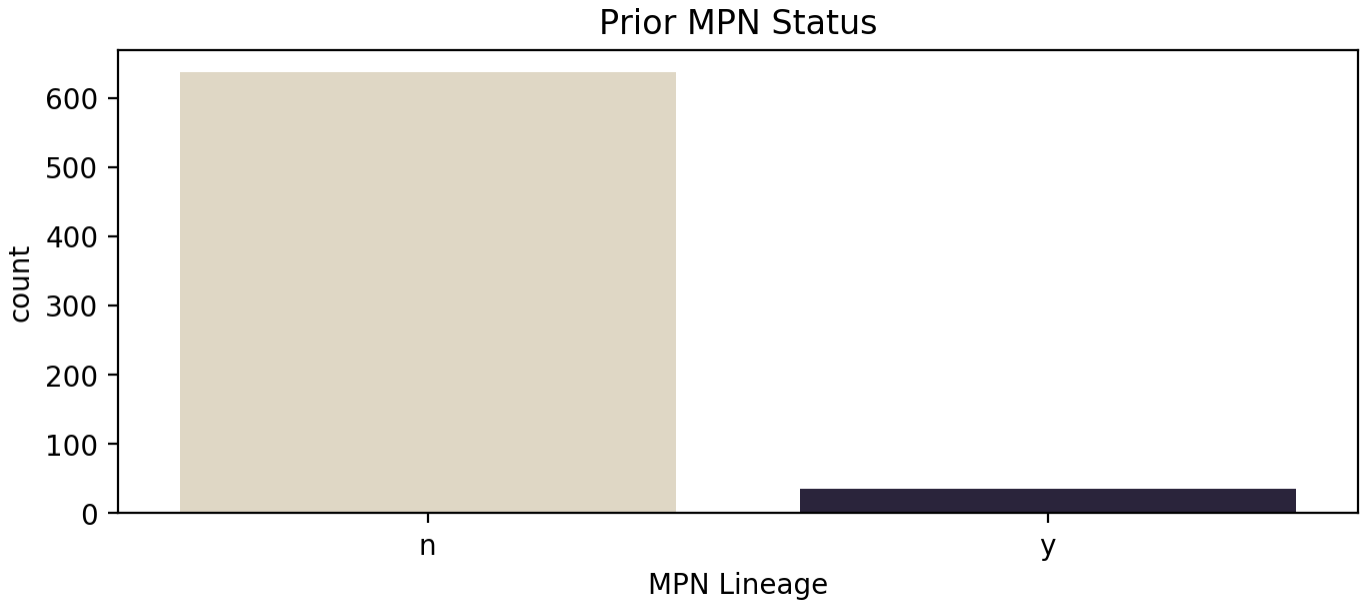
In [79]:
```
#Data Visualization - "priorMDS"
sns.countplot(x=clsm_cut["priorMDS"], palette = "ch:s=-.2,r=.6")
plt.xlabel('MDS Lineage')
plt.title('Prior MDS Status')
plt.gcf().set_size_inches(8, 3)
```



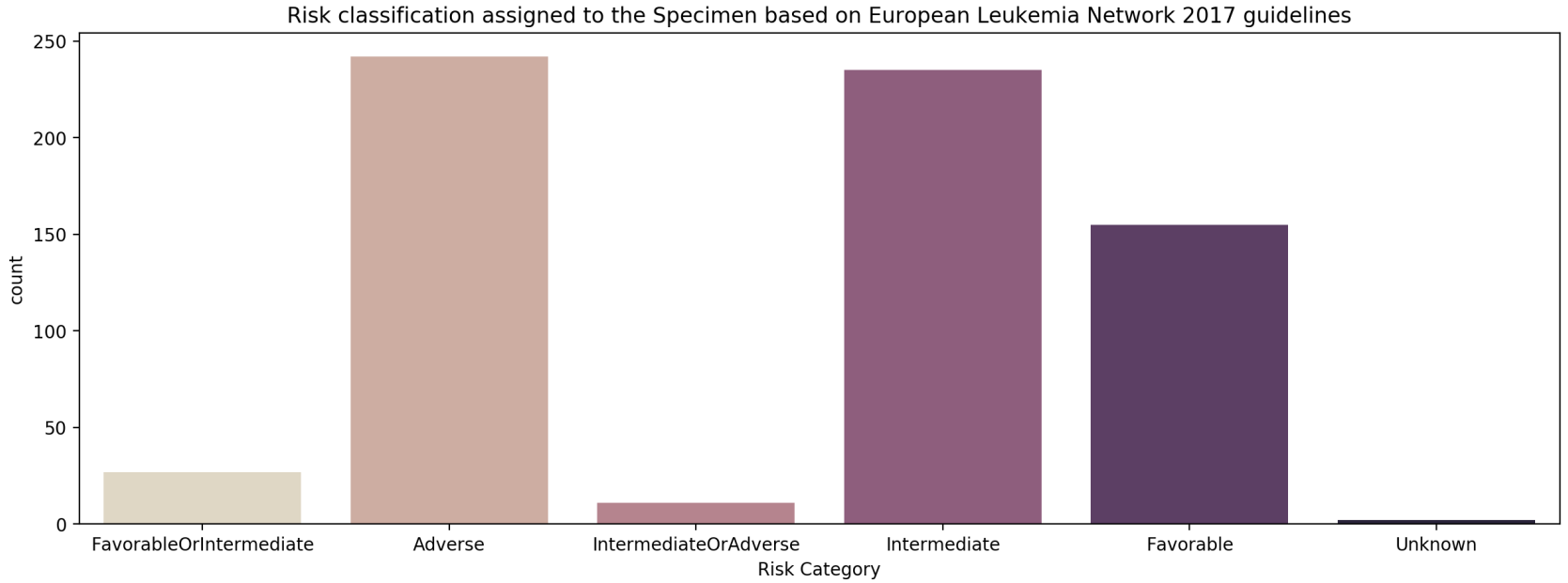
In [80]:
```
#Data Visualization - "priorMDSMPN"
sns.countplot(x=clsm_cut["priorMDSMPN"], palette = "ch:s=-.2,r=.6")
plt.xlabel('MDS and MPN Lineage')
plt.title('Prior MDS and MPN Status')
plt.gcf().set_size_inches(8, 3)
```



In [81]:
```
#Data Visualization - "priorMPN"
sns.countplot(x=clsm_cut["priorMPN"], palette = "ch:s=-.2,r=.6")
plt.xlabel('MPN Lineage')
plt.title('Prior MPN Status')
plt.gcf().set_size_inches(8, 3)
```

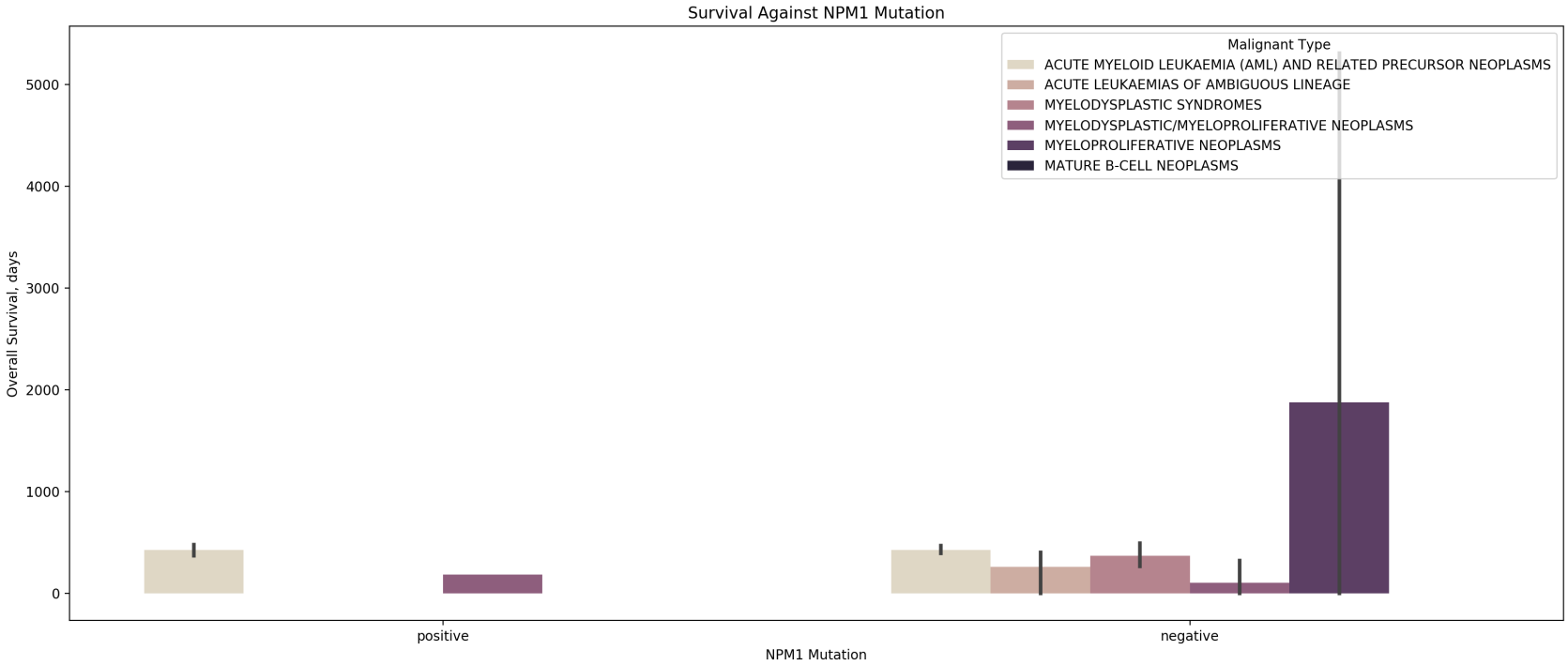


In [82]:
```
#Data Visualization - "ELN2017"
sns.countplot(x=clsm_cut["ELN2017"], palette = "ch:s=-.2,r=.6")
plt.xlabel('Risk Category')
plt.title('Risk classification assigned to the Specimen based on European Leukemia Network 2017 guidelines')
plt.gcf().set_size_inches(15, 5)
```



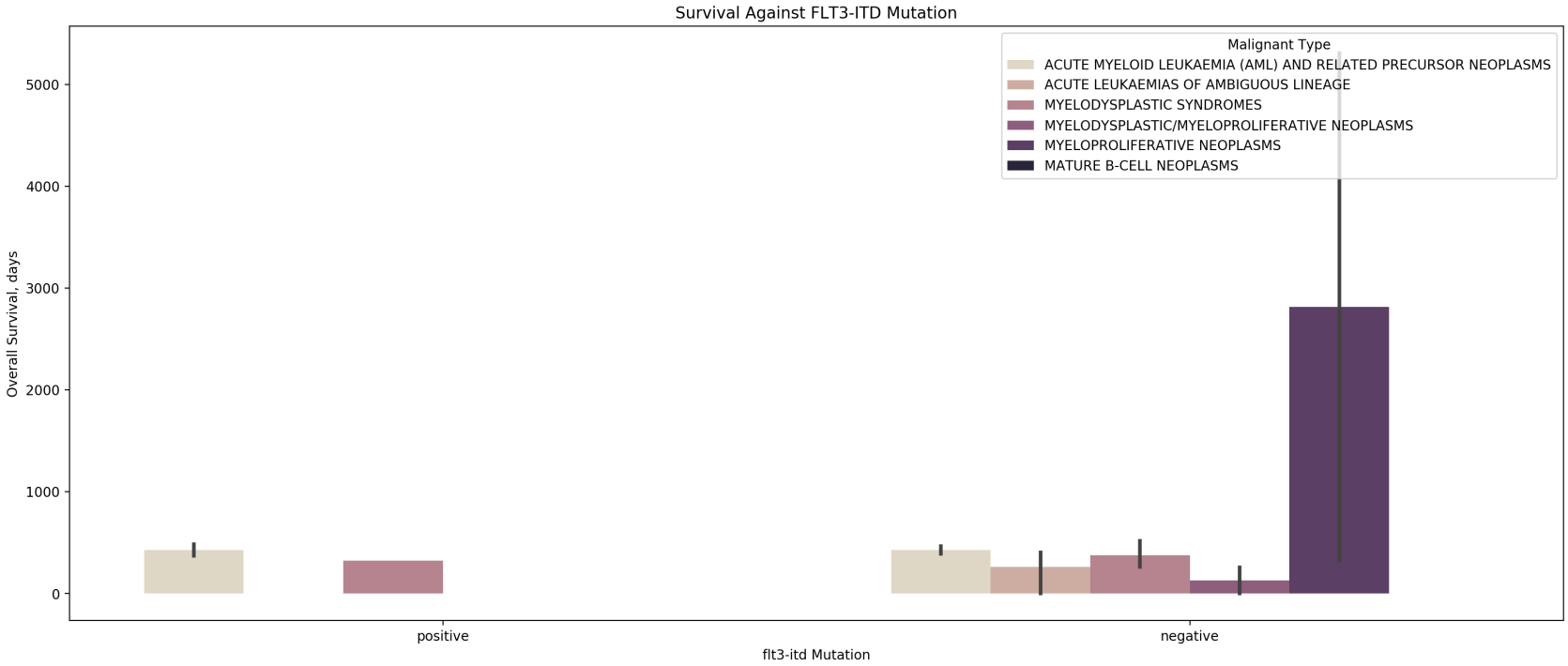
In [83]:
```
sns.barplot(data= clsm_cut,x = 'NPM1', y = 'overallSurvival',
            hue = 'dxAtSpecimenAcquisition', palette = "ch:s=-.2,r=.6")
plt.gcf().set_size_inches(20, 8)
plt.xlabel('NPM1 Mutation')
plt.ylabel('Overall Survival, days')
plt.legend(loc='upper right', title = 'Malignant Type')
plt.title("Survival Against NPM1 Mutation")
```

Out[83]:
```
Text(0.5, 1.0, 'Survival Against NPM1 Mutation')
```



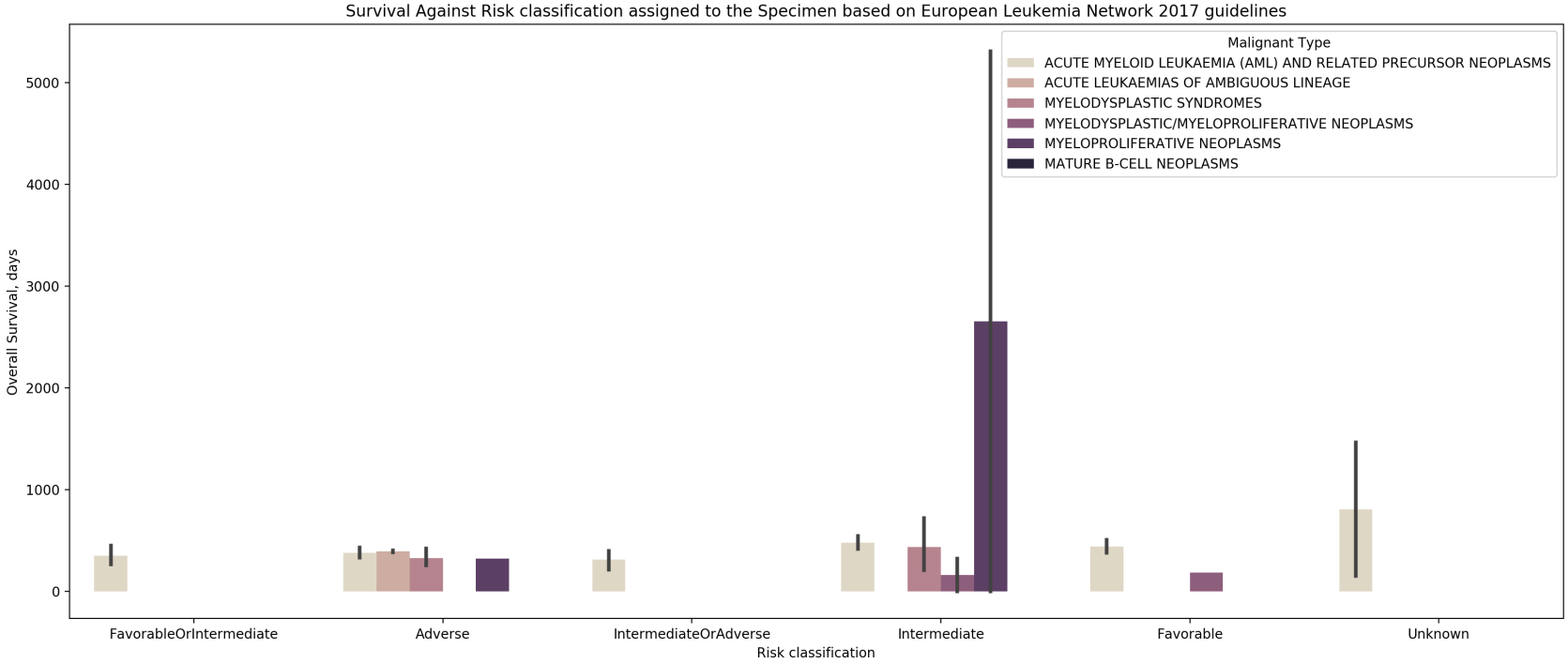
In [84]:
```
sns.barplot(data= clsm_cut,x = 'FLT3-ITD', y = 'overallSurvival',
            hue = 'dxAtSpecimenAcquisition', palette = "ch:s=-.2,r=.6")
plt.gcf().set_size_inches(20, 8)
plt.xlabel('flt3-itd Mutation')
plt.ylabel('Overall Survival, days')
plt.legend(loc='upper right', title = 'Malignant Type')
plt.title("Survival Against FLT3-ITD Mutation")
```

Out[84]:
```
Text(0.5, 1.0, 'Survival Against FLT3-ITD Mutation')
```



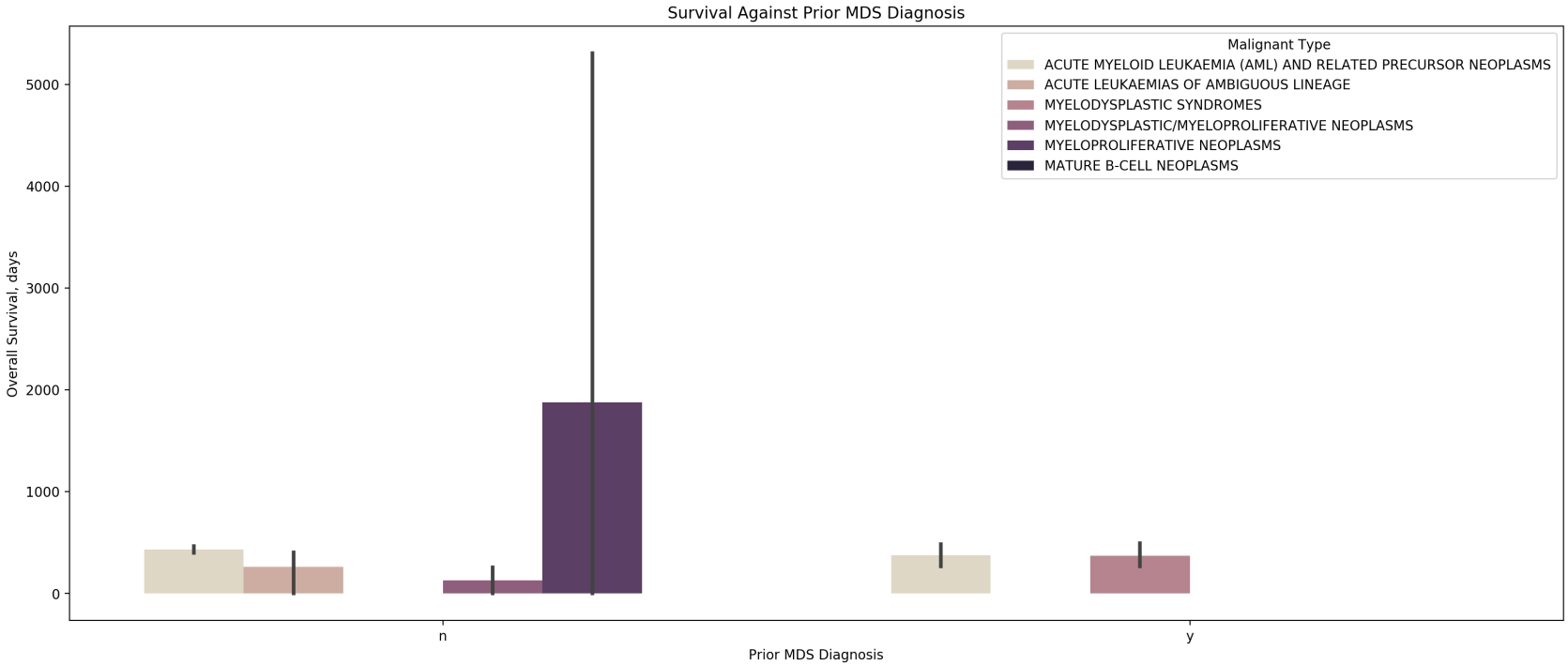
In [85]:
```
sns.barplot(data= clsm_cut,x = 'ELN2017', y = 'overallSurvival',
            hue = 'dxAtSpecimenAcquisition', palette = "ch:s=-.2,r=.6")
plt.gcf().set_size_inches(20, 8)
plt.xlabel('Risk classification')
plt.ylabel('Overall Survival, days')
plt.legend(loc='upper right', title = 'Malignant Type')
plt.title("Survival Against Risk classification assigned to the Specimen based on European Leukemia Network 2017 guic
```

Out[85]:
```
Text(0.5, 1.0, 'Survival Against Risk classification assigned to the Specimen based on European Leukemia Network 201
7 guidelines')
```



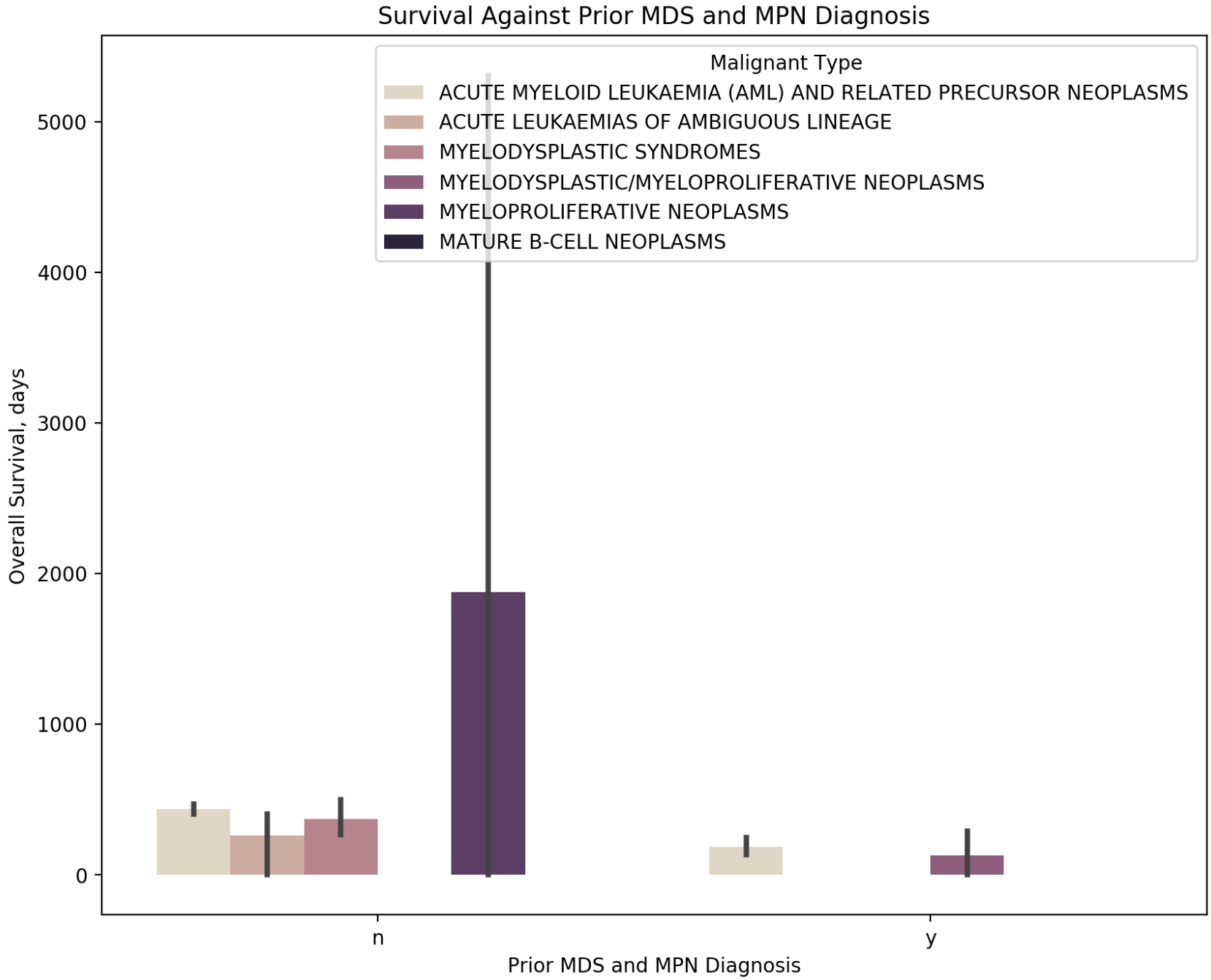
In [86]:
```
sns.barplot(data= clsm_cut,x = 'priorMDS', y = 'overallSurvival',
            hue = 'dxAtSpecimenAcquisition', palette = "ch:s=-.2,r=.6")
plt.gcf().set_size_inches(20, 8)
plt.xlabel('Prior MDS Diagnosis')
plt.ylabel('Overall Survival, days')
plt.legend(loc='upper right', title = 'Malignant Type')
plt.title("Survival Against Prior MDS Diagnosis")
```

Out[86]:
```
Text(0.5, 1.0, 'Survival Against Prior MDS Diagnosis')
```



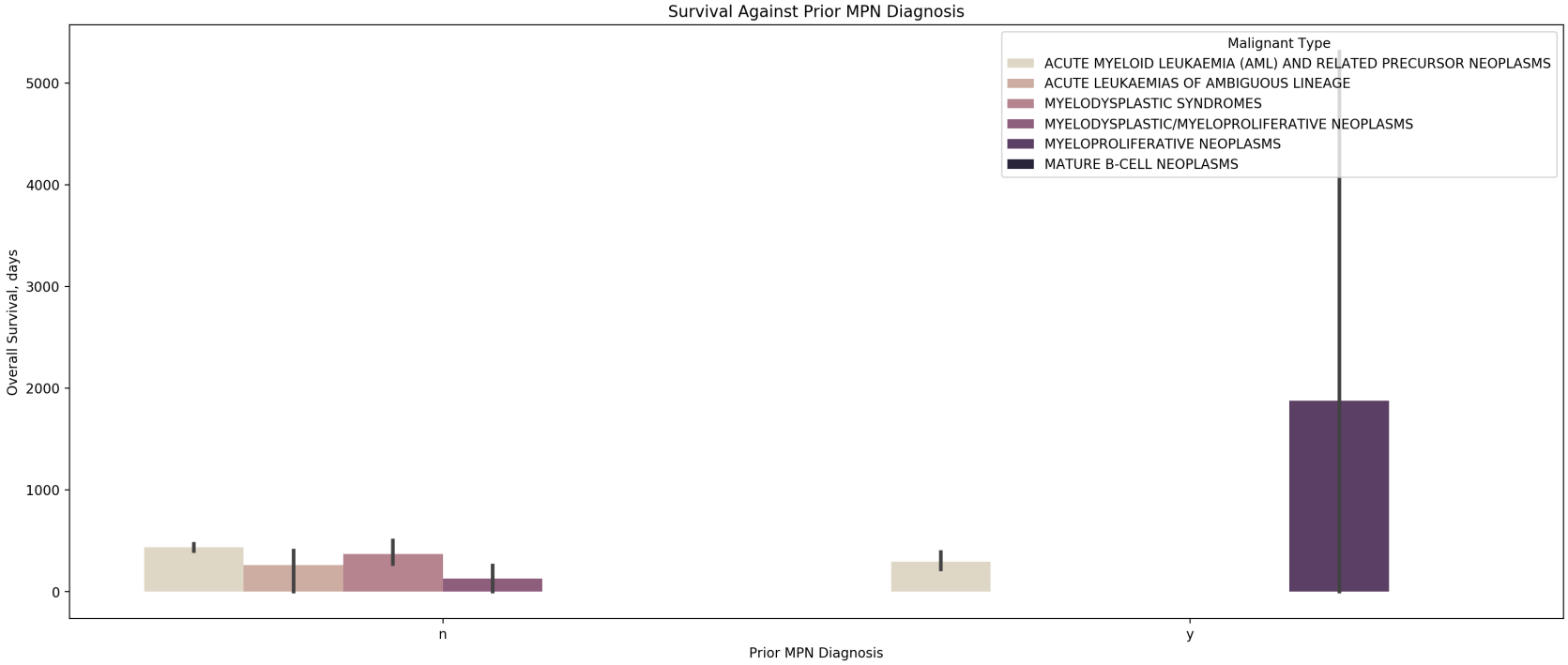
In [87]:
```
sns.barplot(data= clsm_cut,x = 'priorMDSMPN', y = 'overallSurvival',
            hue = 'dxAtSpecimenAcquisition', palette = "ch:s=-.2,r=.6")
plt.gcf().set_size_inches(10, 8)
plt.xlabel('Prior MDS and MPN Diagnosis')
plt.ylabel('Overall Survival, days')
plt.legend(loc='upper right', title = 'Malignant Type')
plt.title("Survival Against Prior MDS and MPN Diagnosis")
```

Out[87]:
```
Text(0.5, 1.0, 'Survival Against Prior MDS and MPN Diagnosis')
```


Survival Against Prior MDS and MPN Diagnosis
Malignant Type
ACUTE MYELOID LEUKAEMIA (AML) AND RELATED PRECURSOR NEOPLASMS
ACUTE LEUKAEMIAS OF AMBIGUOUS LINEAGE
MYELODYSPLASTIC SYNDROMES
MYELODYSPLASTIC/MYELOPROLIFERATIVE NEOPLASMS
MYELOPROLIFERATIVE NEOPLASMS
MATURE B-CELL NEOPLASMS
Overall Survival, days
5000
4000
3000
2000
1000
0
n
y
Prior MDS and MPN Diagnosis

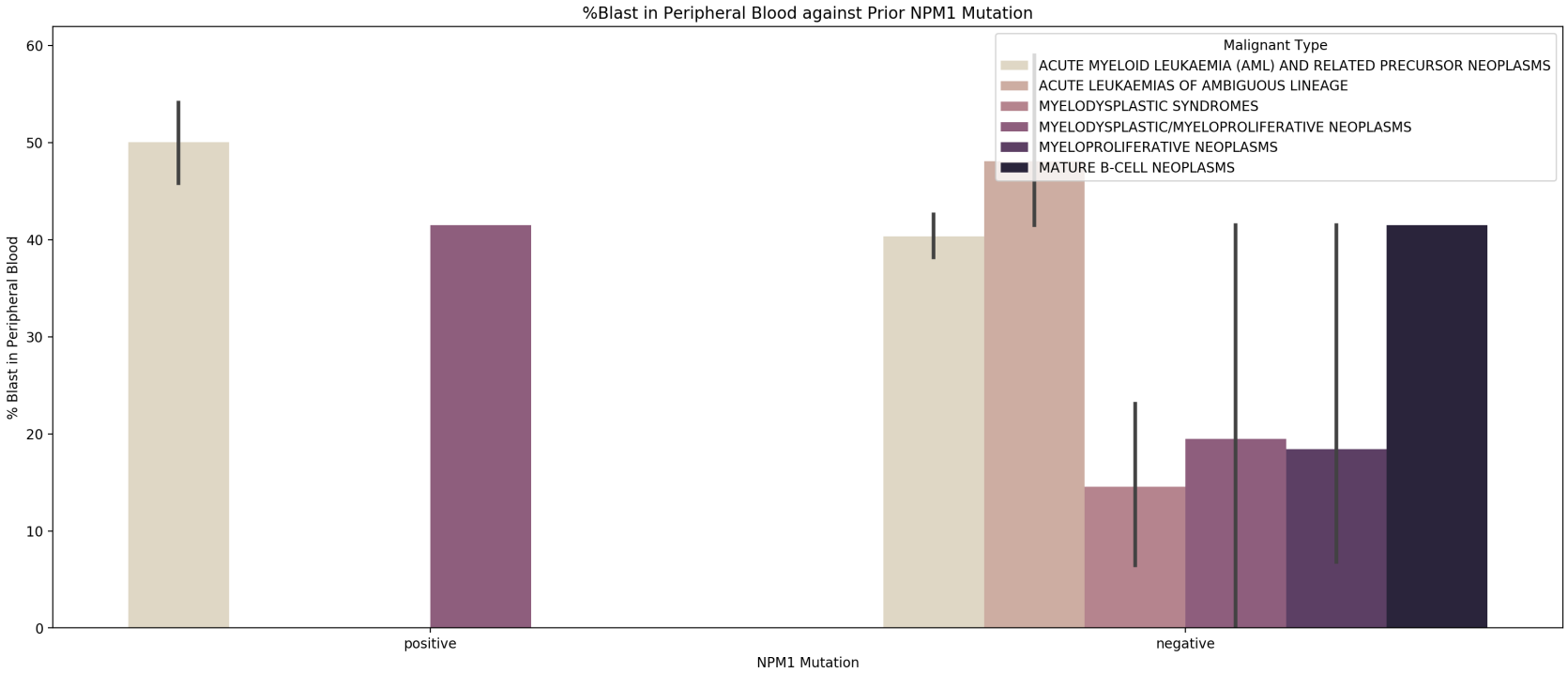
In [88]:
```
sns.barplot(data= clsm_cut,x = 'priorMPN', y = 'overallSurvival',
            hue = 'dxAtSpecimenAcquisition', palette = "ch:s=-.2,r=.6")
plt.gcf().set_size_inches(20, 8)
plt.xlabel('Prior MPN Diagnosis')
plt.ylabel('Overall Survival, days')
plt.legend(loc='upper right', title = 'Malignant Type')
plt.title("Survival Against Prior MPN Diagnosis")
```

Out[88]:
```
Text(0.5, 1.0, 'Survival Against Prior MPN Diagnosis')
```



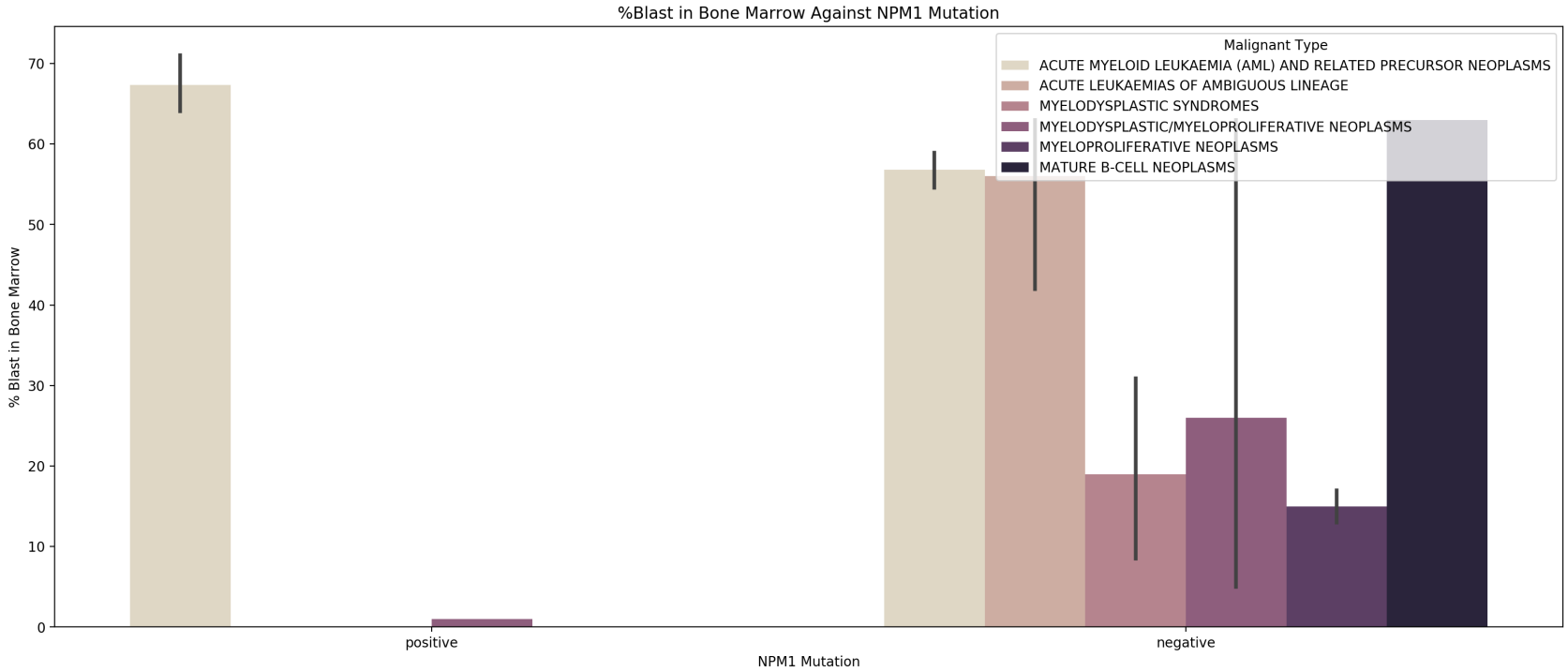
In [89]:
```
sns.barplot(data= clsm_cut,x = 'NPM1', y = '%.Blasts.in.PB',
            hue = 'dxAtSpecimenAcquisition', palette = "ch:s=-.2,r=.6")
plt.gcf().set_size_inches(20, 8)
plt.xlabel('NPM1 Mutation')
plt.ylabel('% Blast in Peripheral Blood')
plt.legend(loc='upper right', title = 'Malignant Type')
plt.title("%Blast in Peripheral Blood against Prior NPM1 Mutation")
```

Out[89]:
```
Text(0.5, 1.0, '%Blast in Peripheral Blood against Prior NPM1 Mutation')
```



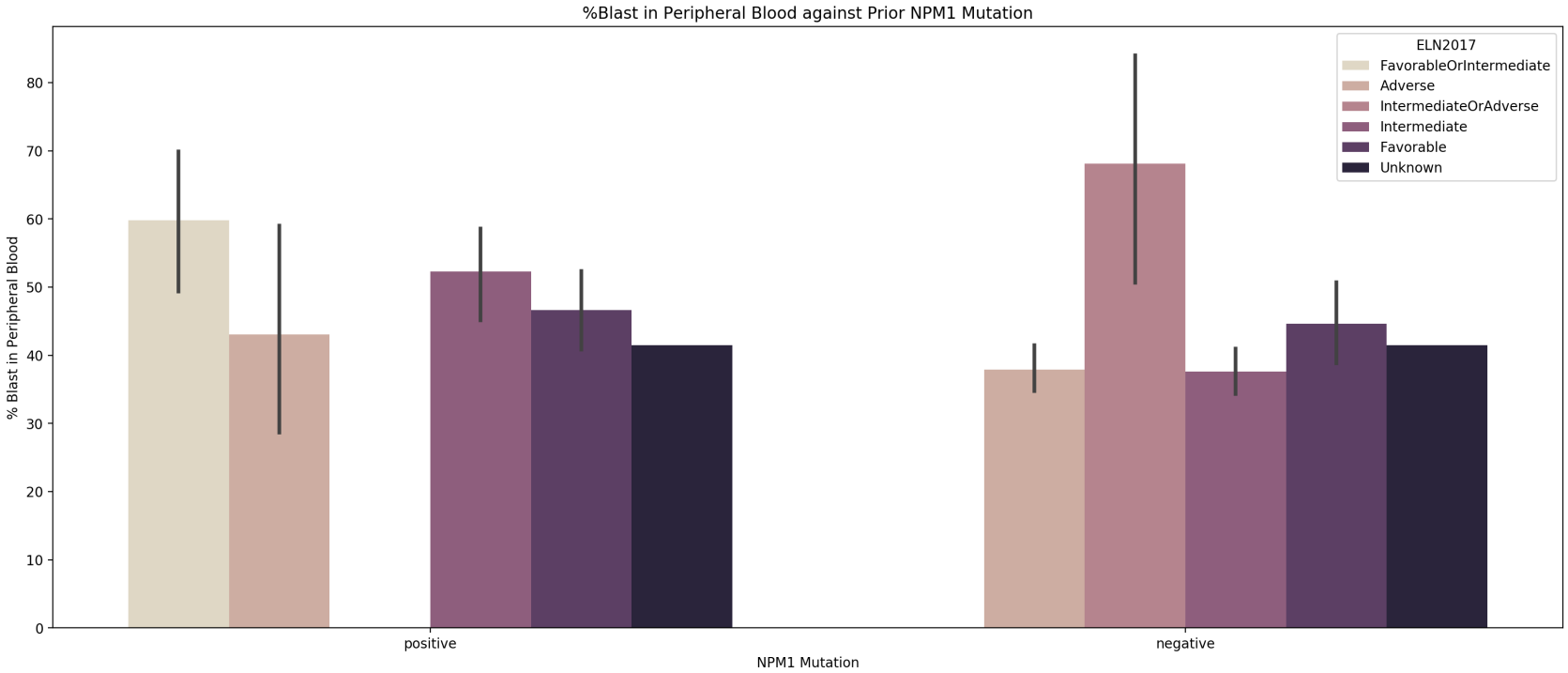
In [90]:
```
sns.barplot(data= clsm_cut,x = 'NPM1', y = '%.Blasts.in.BM',
            hue = 'dxAtSpecimenAcquisition', palette = "ch:s=-.2,r=.6")
plt.gcf().set_size_inches(20, 8)
plt.xlabel('NPM1 Mutation')
plt.ylabel('% Blast in Bone Marrow')
plt.legend(loc='upper right', title = 'Malignant Type')
plt.title("%Blast in Bone Marrow Against NPM1 Mutation")
```

Out[90]:
```
Text(0.5, 1.0, '%Blast in Bone Marrow Against NPM1 Mutation')
```



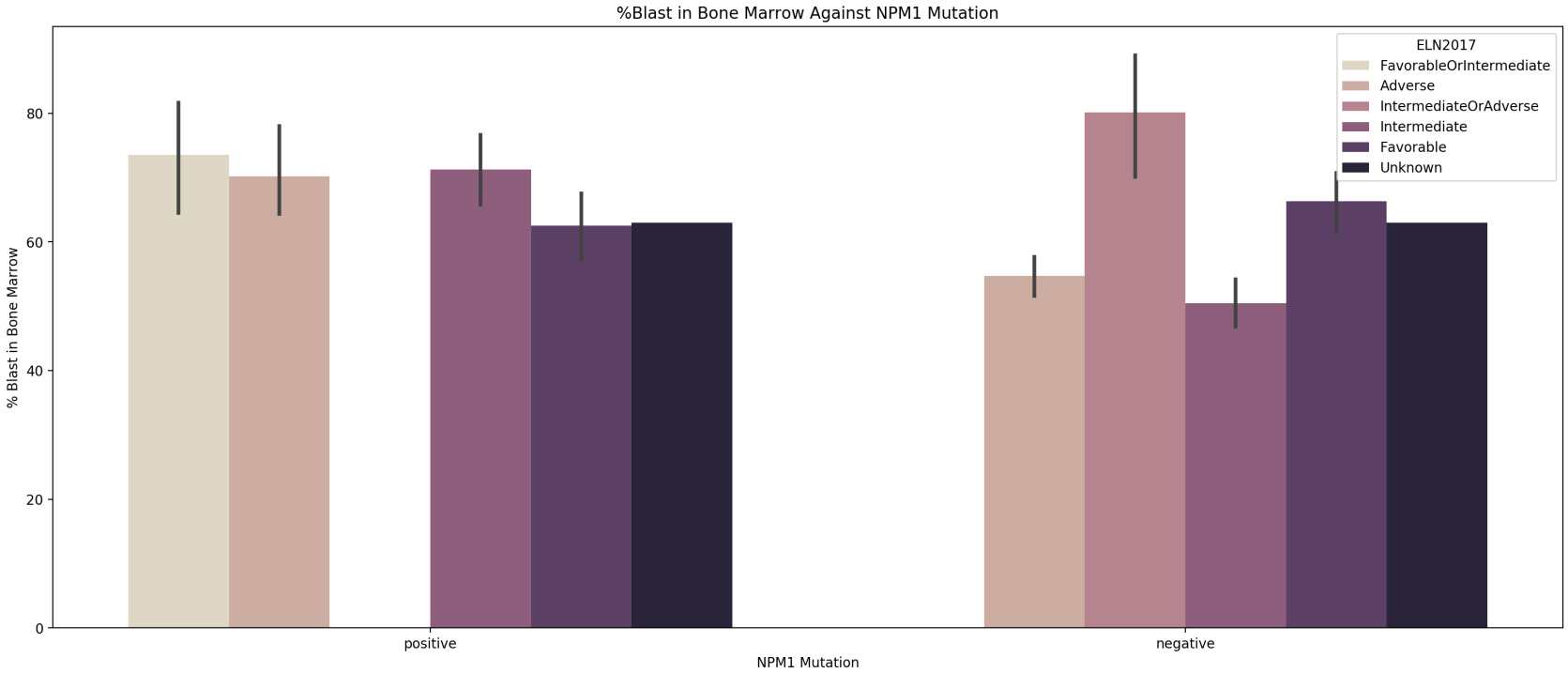
In [91]:
```
sns.barplot(data= clsm_cut,x = 'NPM1', y = '%.Blasts.in.PB',
            hue = 'ELN2017', palette = "ch:s=-.2,r=.6")
plt.gcf().set_size_inches(20, 8)
plt.xlabel('NPM1 Mutation')
plt.ylabel('% Blast in Peripheral Blood')
plt.legend(loc='upper right', title = 'ELN2017')
plt.title("%Blast in Peripheral Blood against Prior NPM1 Mutation")
```

Out[91]:
```
Text(0.5, 1.0, '%Blast in Peripheral Blood against Prior NPM1 Mutation')
```



In [92]:
```
sns.barplot(data= clsm_cut,x = 'NPM1', y = '%.Blasts.in.BM',
            hue = 'ELN2017', palette = "ch:s=-.2,r=.6")
plt.gcf().set_size_inches(20, 8)
plt.xlabel('NPM1 Mutation')
plt.ylabel('% Blast in Bone Marrow')
plt.legend(loc='upper right', title = 'ELN2017')
plt.title("%Blast in Bone Marrow Against NPM1 Mutation")
```

Out[92]:
```
Text(0.5, 1.0, '%Blast in Bone Marrow Against NPM1 Mutation')
```



Transformation: Final Prep prior to Data Modeling

Create Target Variable

In [93]:
```
clsm_cut['dxAtSpecimenAcquisition'].value_counts()
```

Out[93]:
```
ACUTE MYELOID LEUKAEMIA (AML) AND RELATED PRECURSOR NEOPLASMS    646
MYELODYSPLASTIC SYNDROMES                                         15
MYELODYSPLASTIC/MYELOPROLIFERATIVE NEOPLASMS                       4
ACUTE LEUKAEMIAS OF AMBIGUOUS LINEAGE                              3
MYELOPROLIFERATIVE NEOPLASMS                                       3
MATURE B-CELL NEOPLASMS                                            1
Name: dxAtSpecimenAcquisition, dtype: int64
```

In [94]:
```
#create column for AML detected
clsm_cut['AML_detected'] = ['yes' if x == 'ACUTE MYELOID LEUKAEMIA (AML) AND RELATED PRECURSOR NEOPLASMS'
                            else 'no' for x in clsm_cut['dxAtSpecimenAcquisition']]
clsm_cut.head()
```

Out[94]:

	LabId	PatientId	consensus_sex	inferred_ethnicity	isRelapse	isTransformed	priorMalignancyNonMyeloid	priorMDS	priorMDSMPN
0	09-00705	163	Male	White	False	False	n	n	n
1	10-00136	174	Male	White	False	False	n	n	n
2	10-00172	175	Female	White	False	False	n	n	n
3	10-00507	45	Female	White	False	False	n	n	n
4	10-00542	174	Male	White	True	False	n	n	n

In [95]:
```
clsm_cut['AML_detected'].value_counts()
```

Out[95]:
```
yes    646
no      26
Name: AML_detected, dtype: int64
```

New Dataframe for SageMaker JumpStart Regression Model

Transform select categorical attributes to numerical:

In [96]:
```
#AML_detected
clsm_cut['AML_detected'].replace(['no', 'yes'],
                         [0, 1], inplace=True)
#npm1
clsm_cut['NPM1'].replace(['negative', 'positive'],
                         [0, 1], inplace=True)
#flt3-itd
clsm_cut['FLT3-ITD'].replace(['negative', 'positive'],
                         [0, 1], inplace=True)
#priormalignancynonmyeloid
clsm_cut['priorMalignancyNonMyeloid'].replace(['n', 'y'],
                         [0, 1], inplace=True)
#priormds
clsm_cut['priorMDS'].replace(['n', 'y'],
                         [0, 1], inplace=True)
#priormdsmpn
clsm_cut['priorMDSMPN'].replace(['n', 'y'],
                         [0, 1], inplace=True)
#priormpn
clsm_cut['priorMPN'].replace(['n', 'y'],
                         [0, 1], inplace=True)
```

In [97]:
```
#Create new dataframe with transformed attributes and necessary attributes
clsm_cut_transform = pd.DataFrame(clsm_cut[['AML_detected', 'NPM1', 'FLT3-ITD', 'isRelapse', 'isTransformed',
                                'priorMalignancyNonMyeloid', 'priorMDS', 'priorMDSMPN', 'priorMPN',
                                '%.Blasts.in.PB', '%.Blasts.in.BM', 'overallSurvival']])
```

In [98]:
```
#Transform data type:
clsm_cut_transform['NPM1'] = clsm_cut['NPM1'].astype(int)
clsm_cut_transform['FLT3-ITD'] = clsm_cut['FLT3-ITD'].astype(int)
clsm_cut_transform['isRelapse'] = clsm_cut['isRelapse'].astype(int)
clsm_cut_transform['isTransformed'] = clsm_cut['isTransformed'].astype(int)
```

New Numerical Dataframe Correlation Matrix

In [99]:
```
clsm_cut_transform.corr()
```

Out[99]:

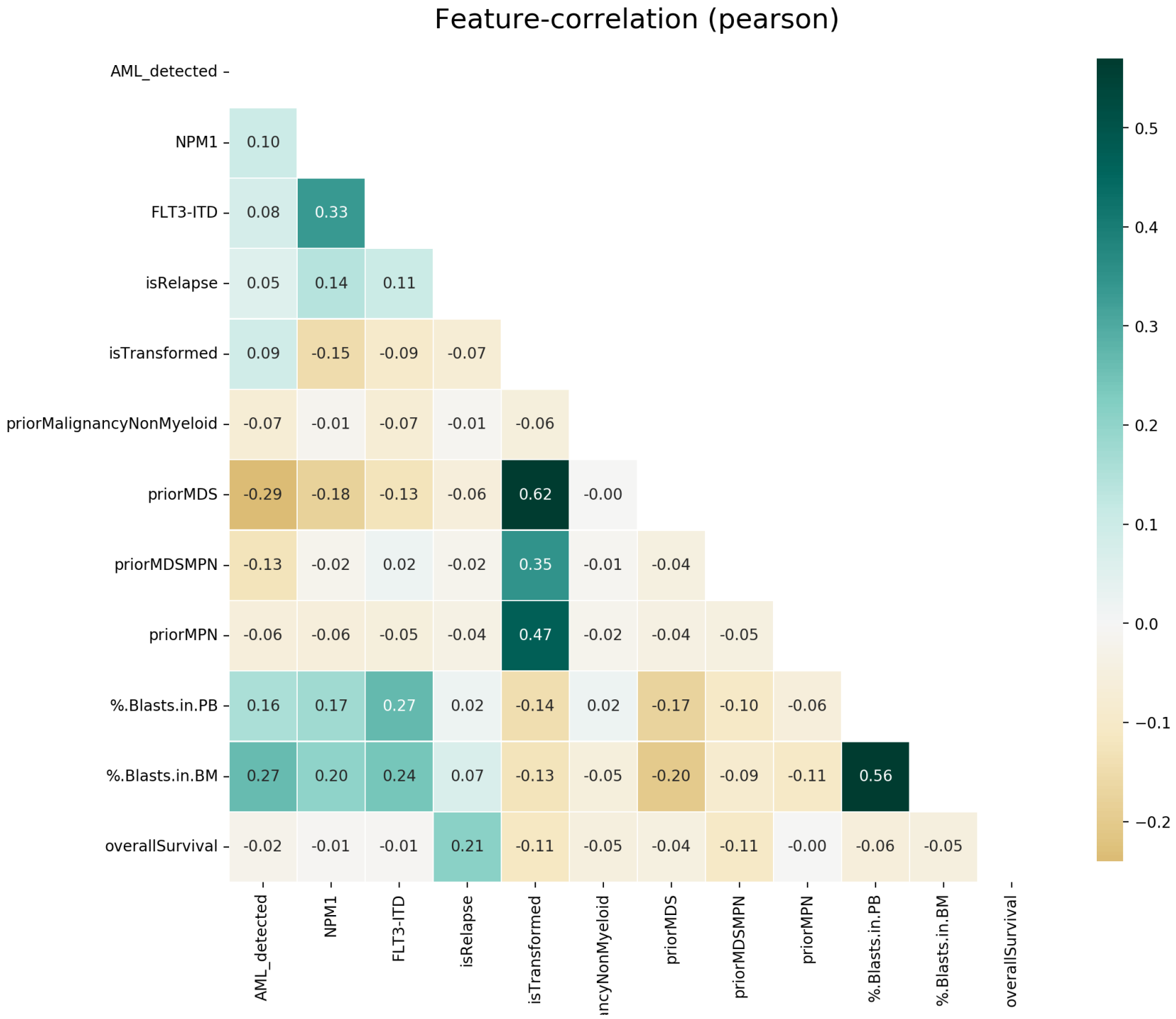
	AML_detected	NPM1	FLT3-ITD	isRelapse	isTransformed	priorMalignancyNonMyeloid	priorMDS	pri
AML_detected	1.000000	0.098997	0.077525	0.054383	0.089238	-7.245182e-02	-2.912038e-01	
NPM1	0.098997	1.000000	0.333543	0.140481	-0.148233	-1.257739e-02	-1.771024e-01	
FLT3-ITD	0.077525	0.333543	1.000000	0.107818	-0.092782	-7.228395e-02	-1.272762e-01	
isRelapse	0.054383	0.140481	0.107818	1.000000	-0.072971	-9.623173e-03	-6.051426e-02	
isTransformed	0.089238	-0.148233	-0.092782	-0.072971	1.000000	-5.562376e-02	6.200179e-01	
priorMalignancyNonMyeloid	-0.072452	-0.012577	-0.072284	-0.009623	-0.055624	1.000000e+00	-1.056121e-17	
priorMDS	-0.291204	-0.177102	-0.127276	-0.060514	0.620018	-1.056121e-17	1.000000e+00	
priorMDSMPN	-0.127707	-0.019763	0.022049	-0.020414	0.346275	-9.820928e-03	-4.405654e-02	
priorMPN	-0.057154	-0.059377	-0.054524	-0.037020	0.472862	-1.913898e-02	-4.227151e-02	
%.Blasts.in.PB	0.155752	0.174675	0.271851	0.020293	-0.141930	2.215108e-02	-1.739113e-01	
%.Blasts.in.BM	0.265724	0.201114	0.242420	0.069284	-0.128224	-5.402601e-02	-2.015171e-01	
overallSurvival	-0.022216	-0.006728	-0.008150	0.210147	-0.113017	-4.893419e-02	-4.466673e-02	

In [100…

```
klib.corr_plot(clsm_cut_transform)
```

Out[100]:

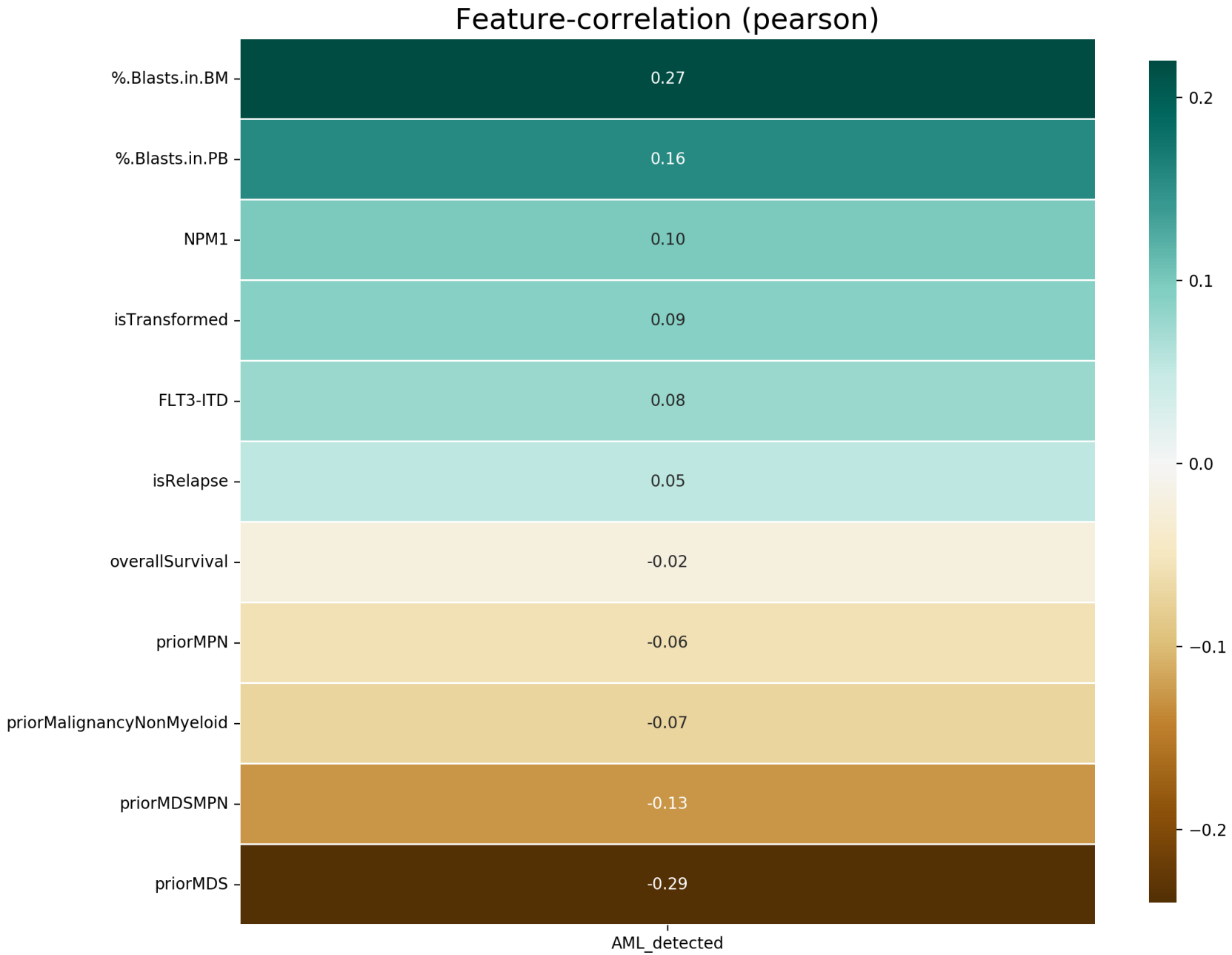
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb8a1237a50>
```


Feature-correlation (pearson)
AML_detected
NPM1
FLT3-ITD
isRelapse
isTransformed
priorMalignancyNonMyeloid
priorMDS
priorMDSMPN
priorMPN
%.Blasts.in.PB
%.Blasts.in.BM
overallSurvival
0.10
0.08 0.33
0.05 0.14 0.11
0.09 -0.15 -0.09 -0.07
-0.07 -0.01 -0.07 -0.01 -0.06
-0.29 -0.18 -0.13 -0.06 0.62 -0.00
-0.13 -0.02 0.02 -0.02 0.35 -0.01 -0.04
-0.06 -0.06 -0.05 -0.04 0.47 -0.02 -0.04 -0.05
0.16 0.17 0.27 0.02 -0.14 0.02 -0.17 -0.10 -0.06
0.27 0.20 0.24 0.07 -0.13 -0.05 -0.20 -0.09 -0.11 0.56
-0.02 -0.01 -0.01 0.21 -0.11 -0.05 -0.04 -0.11 -0.00 -0.06 -0.05
AML_detected
NPM1
FLT3-ITD
isRelapse
isTransformed
incyNonMyeloid
priorMDS
priorMDSMPN
priorMPN
%.Blasts.in.PB
%.Blasts.in.BM
overallSurvival
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2

priorMalign

In [101…
```
klib.corr_plot(clsm_cut_transform, target='AML_detected')
```

Out[101]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb8a110e550>
```



One-Hot Encoding

In [102…
```
clsm_t = pd.get_dummies(clsm_cut_transform, columns= ['NPM1', 'FLT3-ITD', 'priorMalignancyNonMyeloid',
                                                      'priorMDS', 'priorMDSMPN', 'priorMPN', 'isRelapse',
                                                      'isTransformed'])
```

In [103…
```
clsm_t.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 672 entries, 0 to 671
Data columns (total 20 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   AML_detected                 672 non-null    int64
 1   %.Blasts.in.PB               672 non-null    float64
 2   %.Blasts.in.BM               672 non-null    float64
 3   overallSurvival              672 non-null    float64
 4   NPM1_0                       672 non-null    uint8
 5   NPM1_1                       672 non-null    uint8
 6   FLT3-ITD_0                   672 non-null    uint8
 7   FLT3-ITD_1                   672 non-null    uint8
 8   priorMalignancyNonMyeloid_0  672 non-null    uint8
 9   priorMalignancyNonMyeloid_1  672 non-null    uint8
 10  priorMDS_0                   672 non-null    uint8
 11  priorMDS_1                   672 non-null    uint8
 12  priorMDSMPN_0                672 non-null    uint8
 13  priorMDSMPN_1                672 non-null    uint8
 14  priorMPN_0                   672 non-null    uint8
 15  priorMPN_1                   672 non-null    uint8
 16  isRelapse_0                  672 non-null    uint8
 17  isRelapse_1                  672 non-null    uint8
 18  isTransformed_0              672 non-null    uint8
 19  isTransformed_1              672 non-null    uint8
dtypes: float64(3), int64(1), uint8(16)
memory usage: 31.6 KB
```

In [104…
```
clsm_t.head()
```

Out[104]:

	AML_detected	%.Blasts.in.PB	%.Blasts.in.BM	overallSurvival	NPM1_0	NPM1_1	FLT3-ITD_0	FLT3-ITD_1	priorMalignancyNonMyeloid_0	priorMal
0	1	97.0	94.0	425.0	0	1	0	1	1	
1	1	19.0	80.0	419.0	1	0	0	1	1	
2	1	99.0	91.0	541.0	1	0	0	1	1	
3	1	97.0	97.0	511.0	0	1	0	1	1	
4	1	80.0	87.0	419.0	1	0	0	1	1	

In [105…

```
#Transform headers
clsm_t = clsm_t.rename(columns={ '%.Blasts.in.PB': 'Feature_1', '%.Blasts.in.BM': 'Feature_2',
                                 'overallSurvival': 'Feature_3',
                                  'NPM1_0': 'Feature_4', 'NPM1_1': 'Feature_5',
                                 'FLT3-ITD_0': 'Feature_6', 'FLT3-ITD_1': 'Feature_7',
                                  'priorMalignancyNonMyeloid_0': 'Feature_8', 'priorMalignancyNonMyeloid_1'
                                  'priorMDS_0': 'Feature_10', 'priorMDS_1': 'Feature_11',
                                 'priorMDSMPN_0': 'Feature_12', 'priorMDSMPN_1': 'Feature_13',
                                  'priorMPN_0': 'Feature_14', 'priorMPN_1': 'Feature_15',
                                  'isRelapse_0': 'Feature_16', 'isRelapse_1': 'Feature_17',
                                 'isTransformed_0': 'Feature_18', 'isTransformed_1': 'Feature_19' })
```

In [106…

```
clsm_t.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 672 entries, 0 to 671
Data columns (total 20 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   AML_detected  672 non-null    int64
 1   Feature_1     672 non-null    float64
 2   Feature_2     672 non-null    float64
 3   Feature_3     672 non-null    float64
 4   Feature_4     672 non-null    uint8
 5   Feature_5     672 non-null    uint8
 6   Feature_6     672 non-null    uint8
 7   Feature_7     672 non-null    uint8
 8   Feature_8     672 non-null    uint8
 9   Feature_9     672 non-null    uint8
 10  Feature_10    672 non-null    uint8
 11  Feature_11    672 non-null    uint8
 12  Feature_12    672 non-null    uint8
 13  Feature_13    672 non-null    uint8
 14  Feature_14    672 non-null    uint8
 15  Feature_15    672 non-null    uint8
 16  Feature_16    672 non-null    uint8
 17  Feature_17    672 non-null    uint8
 18  Feature_18    672 non-null    uint8
 19  Feature_19    672 non-null    uint8
dtypes: float64(3), int64(1), uint8(16)
memory usage: 31.6 KB
```

Split the Data into Train, Test, and Validation Sets

In [129…

```
clsm_t
```

Out[129]:

	AML_detected	Feature_1	Feature_2	Feature_3	Feature_4	Feature_5	Feature_6	Feature_7	Feature_8	Feature_9	Feature_10	Featur
0	1	97.0	94.0	425.0	0	1	0	1	1	0	1	
1	1	19.0	80.0	419.0	1	0	0	1	1	0	1	
2	1	99.0	91.0	541.0	1	0	0	1	1	0	1	
3	1	97.0	97.0	511.0	0	1	0	1	1	0	1	
4	1	80.0	87.0	419.0	1	0	0	1	1	0	1	
...	...	...	...	...	...	...	...	...	...	...	...	
667	1	53.2	63.0	362.0	1	0	1	0	1	0	1	
668	1	74.0	90.0	323.0	1	0	1	0	1	0	1	
669	1	48.0	63.0	323.0	1	0	0	1	1	0	1	
670	1	41.5	20.0	153.0	1	0	1	0	1	0	1	
671	1	41.5	63.0	256.0	1	0	1	0	1	0	1	

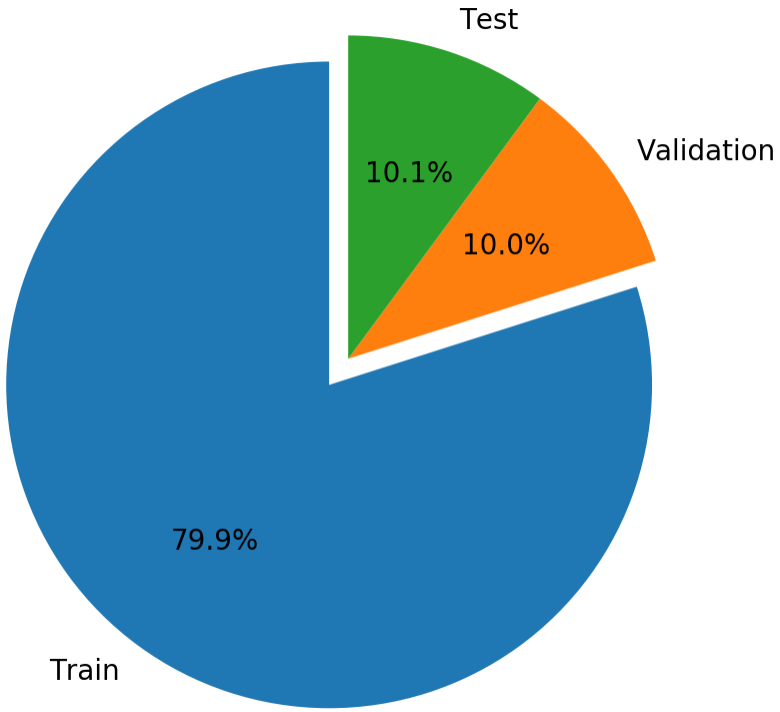
672 rows × 20 columns

In [130…

```
from sklearn.model_selection import train_test_split

# Split all data into 80% train and 20% holdout
clsm_train, clsm_holdout = train_test_split(clsm_t, test_size=0.20, random_state=42)

# Split holdout data into 50% validation and 50% test
clsm_validation, clsm_test = train_test_split(clsm_holdout, test_size=0.50, random_state=42)
```

In [131…

```
# Pie chart, where the slices will be ordered and plotted counter-clockwise:
labels = ["Train", "Validation", "Test"]
sizes = [len(clsm_train.index), len(clsm_validation.index), len(clsm_test.index)]
explode = (0.1, 0, 0)

fig1, ax1 = plt.subplots()

ax1.pie(sizes, explode=explode, labels=labels, autopct="%1.1f%%", startangle=90)

# Equal aspect ratio ensures that pie is drawn as a circle.
ax1.axis("equal")

plt.show()
```



In [132…

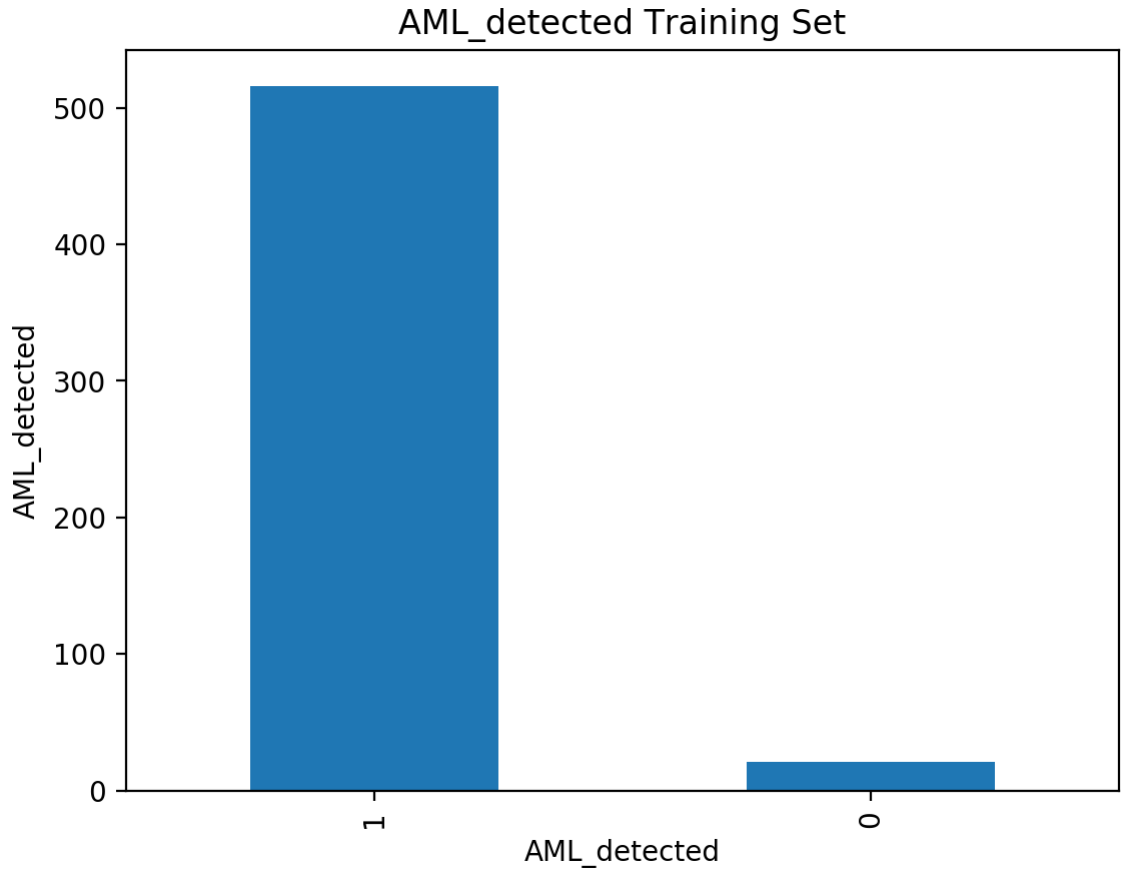
```
#Show 80% Train Data Split
clsm_train.shape
```

Out[132]:

```
(537, 20)
```

In [133…

```
clsm_train["AML_detected"].value_counts().plot(kind="bar", title="AML_detected Training Set")
plt.xlabel("AML_detected")
plt.ylabel("AML_detected")

plt.show()
```



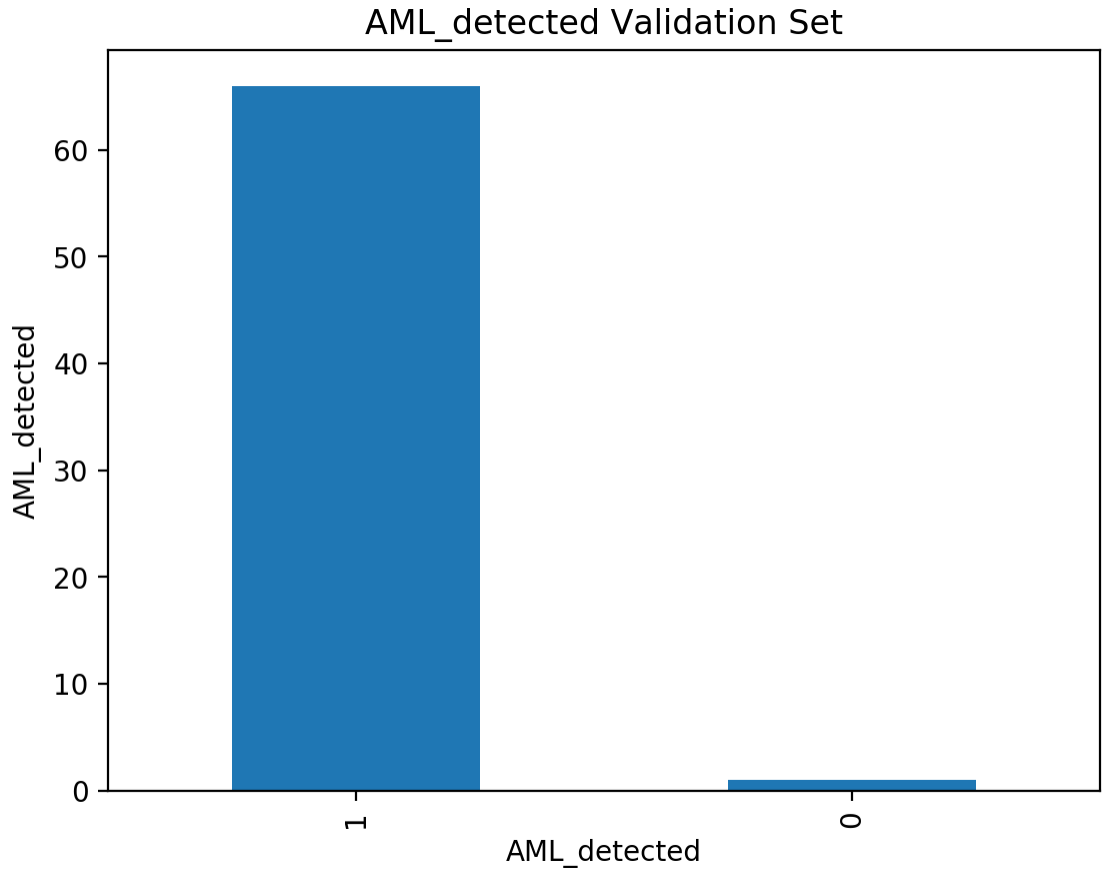
In [134…

```
#Show 5% Validation Data Split
clsm_validation.shape
```

Out[134]:

```
(67, 20)
```

In [135…
```
clsm_validation["AML_detected"].value_counts().plot(kind="bar", title="AML_detected Validation Set")
plt.xlabel("AML_detected")
plt.ylabel("AML_detected")

plt.show()
```

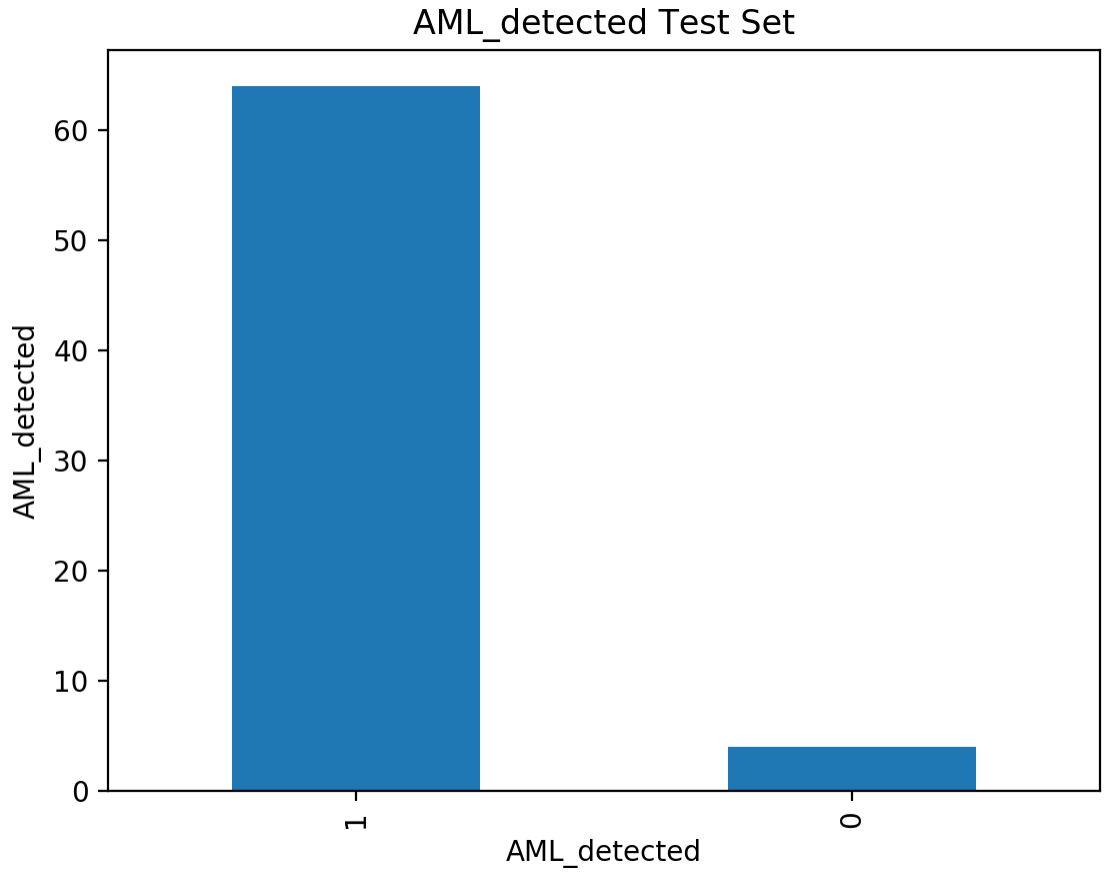


In [138…
```
#Show 5% Test Data Split
clsm_test.shape
```

Out[138]:
```
(68, 20)
```

In [139…

```
clsm_test["AML_detected"].value_counts().plot(kind="bar", title="AML_detected Test Set")
plt.xlabel("AML_detected")
plt.ylabel("AML_detected")

plt.show()
```



Balance Training Data

In [140…

```
clsm_train['AML_detected'].value_counts()
```

Out[140]:

```
1    516
0     21
Name: AML_detected, dtype: int64
```

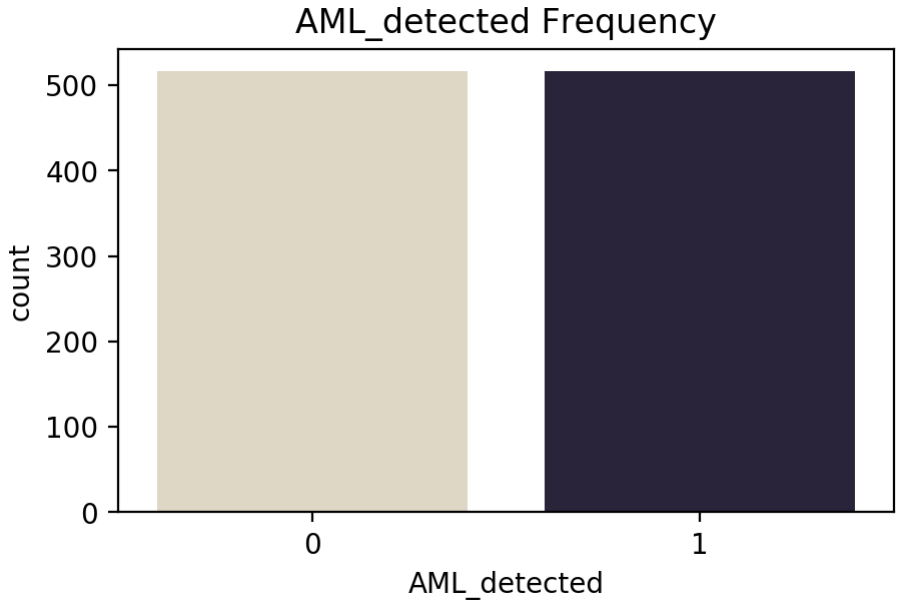
In [141…
```
#resampling of training data set
to_resample= clsm_train.loc[clsm_train['AML_detected'] == 0] #isolate all records of AML_detected
our_resample=to_resample.sample(n=495, replace=True) #sample w/ replacement
clsm_t_rebal=pd.concat([clsm_train, our_resample]) #combine original training set w/ resampled records
clsm_t_rebal['AML_detected'].value_counts()
```

Out[141]:
```
1    516
0    516
Name: AML_detected, dtype: int64
```

In [142…
```
clsm_t_rebal.shape
```

Out[142]:
```
(1032, 20)
```

In [143…
```
sns.countplot(x=clsm_t_rebal["AML_detected"], palette = "ch:s=-.2,r=.6")
plt.xlabel('AML_detected')
plt.title('AML_detected Frequency')
plt.gcf().set_size_inches(5, 3)
```



Model: Logistic Regression

In [145…

```
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import NearestNeighbors, KNeighborsClassifier
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.metrics import accuracy_score, plot_confusion_matrix, classification_report, confusion_matrix, roc_curve
import statistics as stats
```

In [146…

```
#Define/List the features
X_var = list(clsm_t.columns)
X_var
```

Out[146]:

```
['AML_detected',
 'Feature_1',
 'Feature_2',
 'Feature_3',
 'Feature_4',
 'Feature_5',
 'Feature_6',
 'Feature_7',
 'Feature_8',
 'Feature_9',
 'Feature_10',
 'Feature_11',
 'Feature_12',
 'Feature_13',
 'Feature_14',
 'Feature_15',
 'Feature_16',
 'Feature_17',
 'Feature_18',
 'Feature_19']
```

In [147…

```
#Define the target
target ='AML_detected'
X_var.remove(target)

x_train = clsm_t_rebal[X_var]
y_train = clsm_t_rebal[target]
x_test = clsm_test[X_var]
y_test = clsm_test[target]
x_valid = clsm_validation[X_var]
y_valid = clsm_validation[target]
```

In [148…
```
x_train.shape
```

Out[148]:
```
(1032, 19)
```

In [149…
```
x_valid.shape
```

Out[149]:
```
(67, 19)
```

In [150…
```
x_test.shape
```

Out[150]:
```
(68, 19)
```

In [151…
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()

x_train_scaled = scaler.fit_transform(x_train)
x_test_scaled = scaler.fit_transform(x_test)
```

In [152…
```
x_train_scaled
```

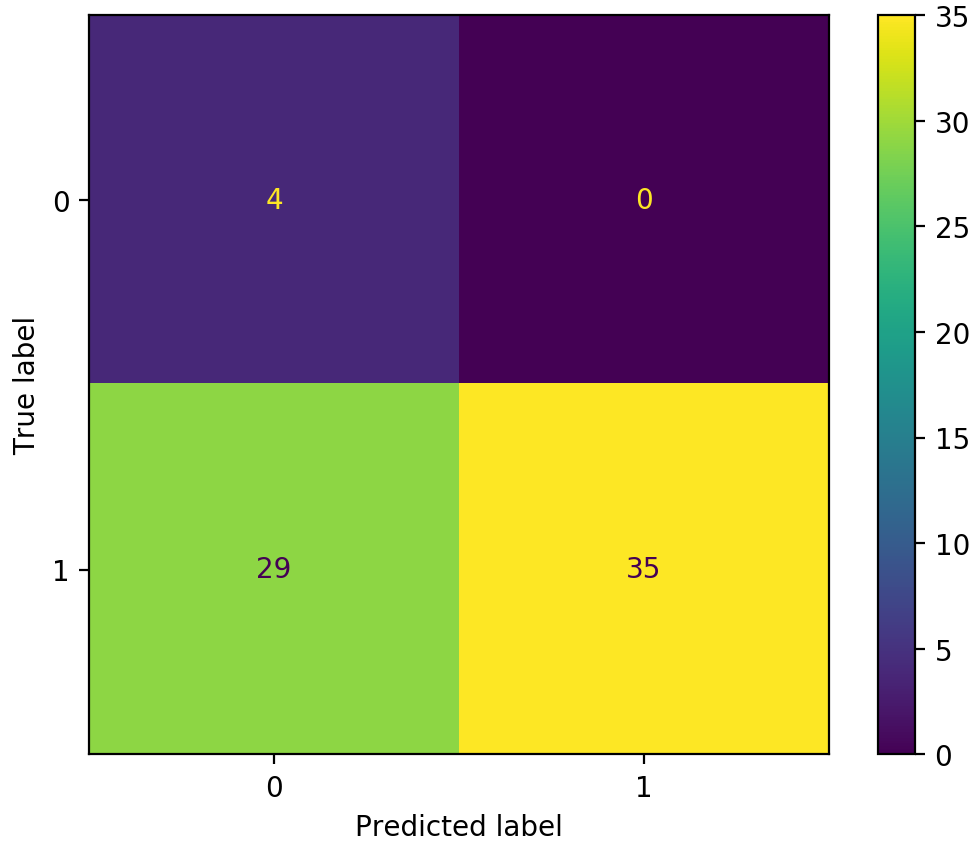
Out[152]:
```
array([[ 2.46104494,  1.73896145, -0.31081992, ..., -0.18174992,
         0.31841389, -0.31841389],
       [ 0.98032538, -0.14328915, -0.33863929, ..., -0.18174992,
         0.31841389, -0.31841389],
       [ 0.37634767,  0.64383383,  1.955194  , ..., -0.18174992,
         0.31841389, -0.31841389],
       ...,
       [ 0.37634767, -1.47797593, -0.32725864, ..., -0.18174992,
         0.31841389, -0.31841389],
       [-0.96798983, -0.93041212, -0.15654888, ..., -0.18174992,
         0.31841389, -0.31841389],
       [-1.16282135, -1.23841677, -0.15654888, ..., -0.18174992,
         0.31841389, -0.31841389]])
```

In [153…
```
x_test_scaled
```

Out[153]:
```
array([[ 0.53249353, -0.20540826, -1.10042836, ..., -0.28171808,
         0.43929769, -0.43929769],
       [-0.75236183, -0.77474636,  0.98176022, ..., -0.28171808,
         0.43929769, -0.43929769],
       [-0.56936728, -1.19226097,  0.43006922, ..., -0.28171808,
         0.43929769, -0.43929769],
       ...,
       [ 0.00687088,  0.25006223,  0.24956203, ..., -0.28171808,
         0.43929769, -0.43929769],
       [-0.90810187,  0.59166509,  2.57327432, ..., -0.28171808,
         0.43929769, -0.43929769],
       [ 0.04191239, -1.0024816 , -0.67839747, ..., -0.28171808,
         0.43929769, -0.43929769]])
```

In [154…
```
#Logistic Regression Model
logit_reg = LogisticRegression(random_state=27)
logit_reg.fit(x_train_scaled, y_train)
y_pred = logit_reg.predict(x_test)

#Predict on validation set
logit_reg_pred1 = logit_reg.predict(x_valid)

plot_confusion_matrix(logit_reg, x_test_scaled, y_test)
plt.grid(False)

#accuracy and classification report on untuned model
print('Untuned Logistic Regression Model')
print('Accuracy Score')
print(accuracy_score(y_valid, logit_reg_pred1))
print('Cross Validation: \n',
      classification_report(y_valid, logit_reg_pred1))
```

```
Untuned Logistic Regression Model
Accuracy Score
0.9850746268656716
Cross Validation:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         1
           1       0.99      1.00      0.99        66

    accuracy                           0.99        67
   macro avg       0.49      0.50      0.50        67
weighted avg       0.97      0.99      0.98        67
```



In [155…

```
#Tune logistic regression model using RandomizedSearchCV() and cross validate with repeated stratified kfold with fiv
#Generate overall best accuracy score with optimal hyperparameters
from sklearn.model_selection import RepeatedStratifiedKFold, RandomizedSearchCV
from scipy.stats import loguniform
model1 = LogisticRegression(random_state=27)
cv = RepeatedStratifiedKFold(n_splits=5, n_repeats=2, random_state=27)

space = dict()

# define search space
space['solver'] = ['newton-cg', 'lbfgs', 'liblinear']
space['penalty'] = ['none', 'l1', 'l2', 'elasticnet']
space['C'] = loguniform(1e-5, 100)

# define search
search = RandomizedSearchCV(model1, space,
                            scoring='accuracy',
n_jobs=-1, cv=cv, random_state=777)

# execute search
result = search.fit(x_train, y_train)

# summarize result
print('Best Score: %s' % result.best_score_)
print('Best Hyperparameters: %s' % result.best_params_)
```

```
Best Score: 0.9103489517377235
Best Hyperparameters: {'C': 0.005198849908368508, 'penalty': 'none', 'solver': 'lbfgs'}
```

Release Resources

In [96]:
```
%%html

<p><b>Shutting down your kernel for this notebook to release resources.</b></p>
<button class="sm-command-button" data-commandlinker-command="kernelmenu:shutdown" style="display:none;">Shutdown Ker

<script>
try {
    els = document.getElementsByClassName("sm-command-button");
    els[0].click();
}
catch(err) {
    // NoOp
}
</script>
```

Shutting down your kernel for this notebook to release resources.

In [132…
```
%%javascript

try {
    Jupyter.notebook.save_checkpoint();
    Jupyter.notebook.session.delete();
}
catch(err) {
    // NoOp
}
```

AutoML approach

Prepare Dataset for Model Training and Evaluating

Cell Data set

https://s3.amazonaws.com/cell_data/

Checking Pre-Requisites from the Previous `01_setup/` Folder

In [2]:
```
%store -r setup_instance_check_passed
```

In [3]:
```
try:
    setup_instance_check_passed
except NameError:
    print("++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Instance Check.")
    print("++++++++++++++++++++++++++++++++")
```

In [4]:
```
print(setup_instance_check_passed)
```

```
True
```

In [5]:
```
%store -r setup_dependencies_passed
```

In [6]:
```
try:
    setup_dependencies_passed
except NameError:
    print("++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Setup Dependencies.")
    print("++++++++++++++++++++++++++++++++")
```

In [7]:
```
print(setup_dependencies_passed)
```

```
True
```

In [8]:
```
%store -r setup_s3_bucket_passed
```

In [9]:
```
try:
    setup_s3_bucket_passed
except NameError:
    print("++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Setup S3 Bucket.")
    print("++++++++++++++++++++++++++++++++")
```

In [10]:
```
print(setup_s3_bucket_passed)
```

```
True
```

In [11]:
```
%store -r setup_iam_roles_passed
```

In [12]:
```
try:
    setup_iam_roles_passed
except NameError:
    print("++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Setup IAM Roles.")
    print("++++++++++++++++++++++++++++++++")
```

In [13]:
```
print(setup_iam_roles_passed)
```

```
True
```

Check if requirements passed

In [14]:
```
if not setup_instance_check_passed:
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Instance Check.")
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
if not setup_dependencies_passed:
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Setup Dependencies.")
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
if not setup_s3_bucket_passed:
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Setup S3 Bucket.")
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
if not setup_iam_roles_passed:
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Setup IAM Roles.")
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
```

In [15]:
```
import boto3
import sagemaker
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import json

sess = sagemaker.Session()
bucket = sess.default_bucket()
role = sagemaker.get_execution_role()
region = boto3.Session().region_name

sm = boto3.Session().client(service_name="sagemaker", region_name=region)
```

Download

Let's start by retrieving a subset of the Amazon Customer Reviews dataset.

In [16]:
```
!aws s3 cp 's3://sagemaker-us-east-1-614093401978/cell_data/OHSU_BeatAMLWaves1_2_Tyner_ClinicalSummary.csv' ./data/
```

```
download: s3://sagemaker-us-east-1-614093401978/cell_data/OHSU_BeatAMLWaves1_2_Tyner_ClinicalSummary.csv to data/OHS
U_BeatAMLWaves1_2_Tyner_ClinicalSummary.csv
```

In [17]:
```
import csv

df = pd.read_csv(
    's3://sagemaker-us-east-1-614093401978/cell_data/OHSU_BeatAMLWaves1_2_Tyner_ClinicalSummary.csv')
df.shape
```

Out[17]:
```
(672, 159)
```

In [18]:
```
df.head(5)
```

Out[18]:

	LabId	PatientId	consensus_sex	inferred_sex	inferred_ethnicity	centerID	CEBPA_Biallelic	ageAtDiagnosis	isRelapse	isDenovo	...
0	09-00705	163	Male	Male	White	1	n	73.0	False	True	...
1	10-00136	174	Male	Male	White	1	n	69.0	False	True	...
2	10-00172	175	Female	Male	White	1	n	59.0	False	True	...
3	10-00507	45	Female	Female	White	1	n	70.0	False	True	...
4	10-00542	174	Male	Male	White	1	n	69.0	True	False	...

5 rows × 159 columns

In [19]:
```
!aws s3 cp 's3://sagemaker-us-east-1-614093401978/cell_data/OHSU_BeatAMLWaves1_2_Tyner_DrugResponse.csv' ./data/
```

```
download: s3://sagemaker-us-east-1-614093401978/cell_data/OHSU_BeatAMLWaves1_2_Tyner_DrugResponse.csv to data/OHSU_BeatAMLWaves1_2_Tyner_DrugResponse.csv
```

In [20]:
```
df1 = pd.read_csv(
    's3://sagemaker-us-east-1-614093401978/cell_data/OHSU_BeatAMLWaves1_2_Tyner_DrugResponse.csv')
df1.shape
```

Out[20]:
```
(47650, 4)
```

In [21]:
```
df1.head(5)
```

Out[21]:

	inhibitor	lab_id	ic50	auc
0	17-AAG (Tanespimycin)	12-00211	10.000000	225.918025
1	17-AAG (Tanespimycin)	12-00219	0.276661	135.264409
2	17-AAG (Tanespimycin)	12-00258	2.722845	164.561227
3	17-AAG (Tanespimycin)	12-00262	0.123136	111.555971
4	17-AAG (Tanespimycin)	12-00268	10.000000	226.805281

In [22]:
```
clsm = df.replace('', np.NAN)
clsm.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 672 entries, 0 to 671
Columns: 159 entries, LabId to ZRSR2
dtypes: bool(9), float64(22), int64(7), object(121)
memory usage: 793.5+ KB
```

In [23]:
```
clsm.info(2)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 672 entries, 0 to 671
Data columns (total 159 columns):
 #    Column                                          Dtype
---   ------                                          -----
 0    LabId                                           object
 1    PatientId                                       int64
 2    consensus_sex                                   object
 3    inferred_sex                                    object
 4    inferred_ethnicity                              object
 5    centerID                                        int64
 6    CEBPA_Biallelic                                 object
 7    ageAtDiagnosis                                  float64
 8    isRelapse                                       bool
 9    isDenovo                                        bool
 10   isTransformed                                   bool
 11   finalFusion                                     object
 12   specificDxAtAcquisition_MDSMPN                  bool
 13   nonAML_MDSMPN_specificDxAtAcquisition           bool
 14   priorMalignancyNonMyeloid                       object
 15   priorMalignancyType                             object
 16   cumulativeChemo                                 object
 17   priorMalignancyRadiationTx                      object
 18   priorMDS                                        object
 19   priorMDSMoreThanTwoMths                         object
 20   priorMDSMPN                                     object
 21   priorMDSMPNMoreThanTwoMths                      object
 22   priorMPN                                        object
 23   priorMPNMoreThanTwoMths                         object
 24   dxAtInclusion                                   object
 25   specificDxAtInclusion                           object
 26   ELN2017                                         object
 27   ELN2008                                         object
 28   dxAtSpecimenAcquisition                         object
 29   specificDxAtAcquisition                         object
 30   ageAtSpecimenAcquisition                        float64
 31   timeOfSampleCollectionRelativeToInclusion       int64
 32   specimenGroups                                  object
 33   specimenType                                    object
 34   rnaSeq                                          object
 35   exomeSeq                                        object
 36   totalDrug                                       object
 37   rnaSeqAnalysis                                  object
 38   analysisExomeSeq                                object
```

```
39  analysisDrug                                    object
40  cumulativeTreatmentTypeCount                    int64
41  cumulativeTreatmentTypes                        object
42  cumulativeTreatmentRegimenCount                 int64
43  cumulativeTreatmentRegimens                     object
44  cumulativeTreatmentStageCount                   int64
45  cumulativeTreatmentStages                       object
46  responseToInductionTx                           object
47  typeInductionTx                                 object
48  responseDurationToInductionTx                   float64
49  mostRecentTreatmentType                         object
50  currentRegimen                                  object
51  currentStage                                    object
52  mostRecentTreatmentDuration                     float64
53  vitalStatus                                     object
54  overallSurvival                                 float64
55  causeOfDeath                                    object
56  any_different_labs                              bool
57  any_different_labs_also_beataml                 bool
58  different_lab_ids                               object
59  different_id_karyotype_interval                 int64
60  %.Basophils.in.PB                               float64
61  %.Blasts.in.BM                                  object
62  %.Blasts.in.PB                                  object
63  %.Eosinophils.in.PB                             float64
64  %.Immature.Granulocytes.in.PB                   float64
65  %.Lymphocytes.in.PB                             float64
66  %.Monocytes.in.PB                               float64
67  %.Neutrophils.in.PB                             float64
68  %.Nucleated.RBCs.in.PB                          float64
69  ALT                                             object
70  AST                                             float64
71  Albumin                                         float64
72  Creatinine                                      float64
73  FAB/Blast.Morphology                            object
74  Hematocrit                                      float64
75  Hemoglobin                                      float64
76  Karyotype                                       object
77  LDH                                             float64
78  MCV                                             float64
79  Other.Cytogenetics                              object
80  Platelet.Count                                  float64
81  Surface.Antigens.(Immunohistochemical.Stains)   object
82  Total.Protein                                   float64
```

83	WBC.Count	float64
84	any_different_cgs	bool
85	any_different_cgs_also_beataml	bool
86	different_cgs_lab_ids	object
87	FLT3-ITD	object
88	NPM1	object
89	ABL1	object
90	ASXL1	object
91	ASXL2	object
92	ATM	object
93	BCOR	object
94	BCORL1	object
95	BRAF	object
96	BRCA2	object
97	CALR	object
98	CBL	object
99	CCND2	object
100	CCND3	object
101	CD36	object
102	CEBPA	object
103	CHEK2	object
104	CIITA	object
105	CREBBP	object
106	CSF3R	object
107	CTCF	object
108	CUX1	object
109	DNMT3A	object
110	EP300	object
111	ETV6	object
112	EZH2	object
113	FBXW7	object
114	FLT3	object
115	GATA1	object
116	GATA2	object
117	IDH1	object
118	IDH2	object
119	IKZF1	object
120	JAK1	object
121	JAK2	object
122	JAK3	object
123	KDM6A	object
124	KIT	object
125	KMT2A	object
126	KMT2D	object

```
 127  KRAS                                               object
 128  MEN1                                               object
 129  MPL                                                object
 130  MUTYH                                              object
 131  MYD88                                              object
 132  NF1                                                object
 133  NOTCH1                                             object
 134  NRAS                                               object
 135  PAX5                                               object
 136  PDGFRB                                             object
 137  PHF6                                               object
 138  POT1                                               object
 139  PRDM1                                              object
 140  PTPN11                                             object
 141  RAD21                                              object
 142  ROS1                                               object
 143  RUNX1                                              object
 144  SETBP1                                             object
 145  SF3B1                                              object
 146  SMC1A                                              object
 147  SOCS1                                              object
 148  SRSF2                                              object
 149  STAG2                                              object
 150  STAT3                                              object
 151  SUZ12                                              object
 152  TCL1A                                              object
 153  TET2                                               object
 154  TP53                                               object
 155  TYK2                                               object
 156  U2AF1                                              object
 157  WT1                                                object
 158  ZRSR2                                              object
dtypes: bool(9), float64(22), int64(7), object(121)
memory usage: 793.5+ KB
```

In [24]:
```
!pip install klib
```

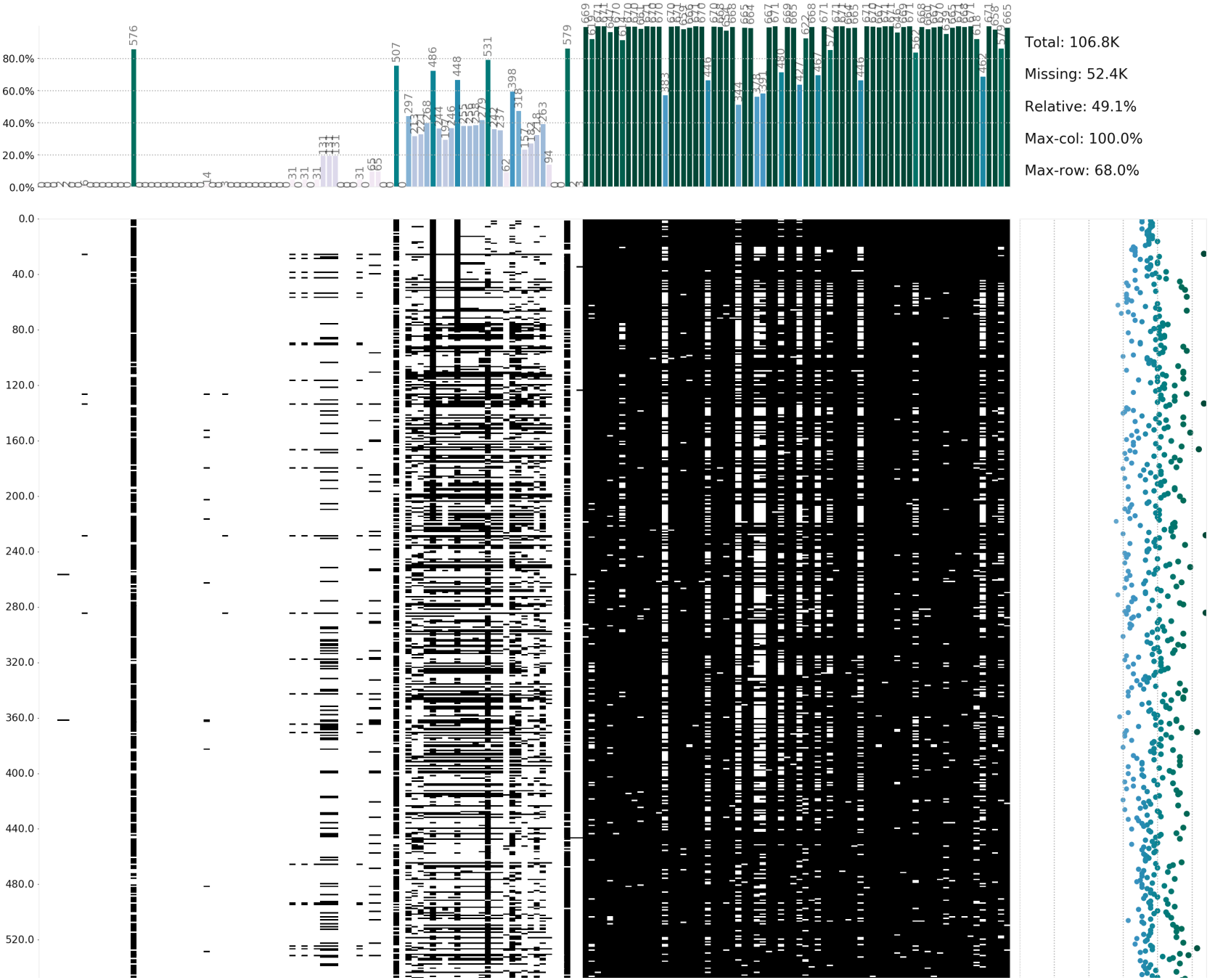
```
Collecting klib
  Using cached klib-1.0.1-py3-none-any.whl (20 kB)
Requirement already satisfied: scipy<2.0.0,>=1.1.0 in /opt/conda/lib/python3.7/site-packages (from klib) (1.4.1)
Requirement already satisfied: numpy<2.0.0,>=1.16.3 in /opt/conda/lib/python3.7/site-packages (from klib) (1.21.6)
Collecting seaborn<0.12.0,>=0.11.1
  Using cached seaborn-0.11.2-py3-none-any.whl (292 kB)
Requirement already satisfied: matplotlib<4.0.0,>=3.0.3 in /opt/conda/lib/python3.7/site-packages (from klib) (3.1.
3)
Requirement already satisfied: pandas<2.0.0,>=1.1.2 in /opt/conda/lib/python3.7/site-packages (from klib) (1.3.5)
Requirement already satisfied: Jinja2<4.0.0,>=3.0.3 in /opt/conda/lib/python3.7/site-packages (from klib) (3.1.2)
Requirement already satisfied: MarkupSafe>=2.0 in /opt/conda/lib/python3.7/site-packages (from Jinja2<4.0.0,>=3.0.3-
>klib) (2.1.2)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.7/site-packages (from matplotlib<4.0.0,>=3.0.3
->klib) (0.10.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /opt/conda/lib/python3.7/site-packages (f
rom matplotlib<4.0.0,>=3.0.3->klib) (2.4.6)
Requirement already satisfied: kiwisolver>=1.0.1 in /opt/conda/lib/python3.7/site-packages (from matplotlib<4.0.0,>=
3.0.3->klib) (1.1.0)
Requirement already satisfied: python-dateutil>=2.1 in /opt/conda/lib/python3.7/site-packages (from matplotlib<4.0.
0,>=3.0.3->klib) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/lib/python3.7/site-packages (from pandas<2.0.0,>=1.1.2->kl
ib) (2019.3)
Requirement already satisfied: six in /opt/conda/lib/python3.7/site-packages (from cycler>=0.10->matplotlib<4.0.0,>=
3.0.3->klib) (1.14.0)
Requirement already satisfied: setuptools in /opt/conda/lib/python3.7/site-packages (from kiwisolver>=1.0.1->matplot
lib<4.0.0,>=3.0.3->klib) (59.3.0)
Installing collected packages: seaborn, klib
  Attempting uninstall: seaborn
    Found existing installation: seaborn 0.10.0
    Uninstalling seaborn-0.10.0:
      Successfully uninstalled seaborn-0.10.0
Successfully installed klib-1.0.1 seaborn-0.11.2
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system p
ackage manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv

[notice] A new release of pip is available: 23.0.1 -> 23.1
[notice] To update, run: pip install --upgrade pip
```

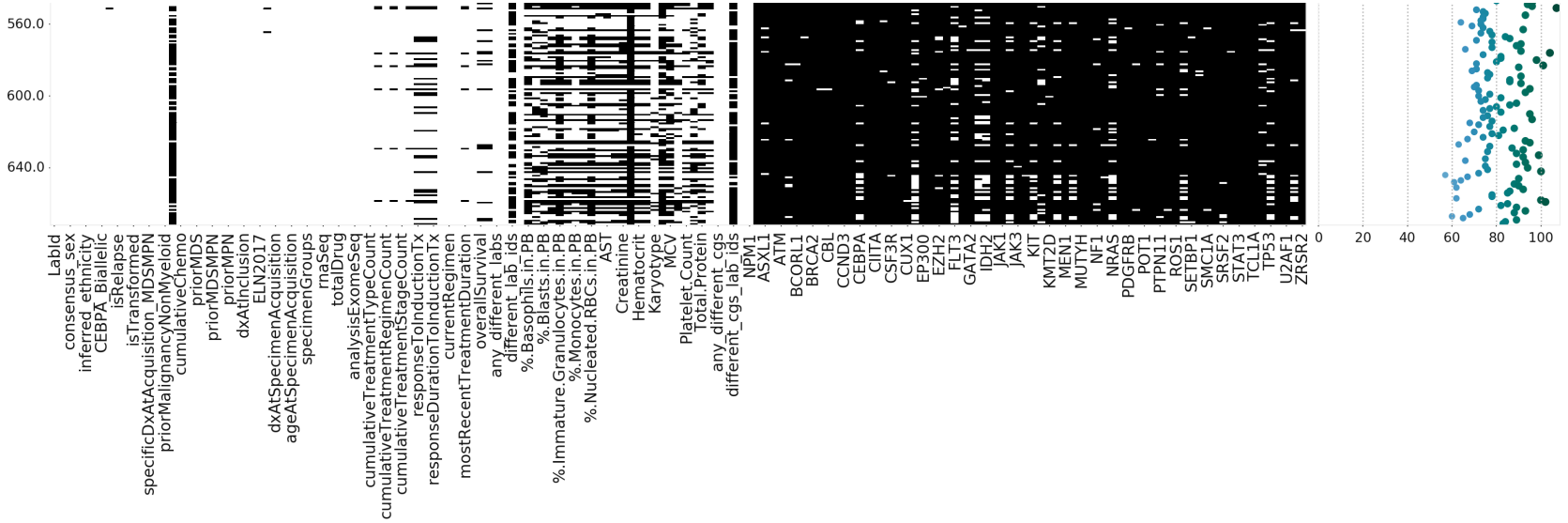
In [25]:
```
import numpy as np
import seaborn as sns
import klib
import matplotlib.pyplot as plt

%matplotlib inline
%config InlineBackend.figure_format='retina'
```

In [26]:
```
klib.missingval_plot(clsm)
```

Out[26]:
```
GridSpec(6, 6)
```

Missing value plot





Select Relevant Features

In [27]:
```
clsm.columns = map(str.lower, clsm.columns)
```

In [28]:
```
clsm_cut = pd.DataFrame(clsm[['labid', 'patientid', 'consensus_sex', 'inferred_ethnicity', 'isrelapse',
                              'istransformed', 'priormalignancynonmyeloid', 'priormds', 'priormdsmpn', 'priormpn',
                              'eln2017', 'dxatspecimenacquisition', 'vitalstatus', 'overallsurvival', '%.blasts.in.bm'
                               '%.blasts.in.pb','flt3-itd', 'npm1']])

clsm_cut.head(5)
```

Out[28]:

	labid	patientid	consensus_sex	inferred_ethnicity	isrelapse	istransformed	priormalignancynonmyeloid	priormds	priormdsmpn	p
0	09-00705	163	Male	White	False	False	n	n	n	
1	10-00136	174	Male	White	False	False	n	n	n	
2	10-00172	175	Female	White	False	False	n	n	n	
3	10-00507	45	Female	White	False	False	n	n	n	
4	10-00542	174	Male	White	True	False	n	n	n	

In [29]:
```
clsm_cut.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 672 entries, 0 to 671
Data columns (total 18 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   labid                       672 non-null    object
 1   patientid                   672 non-null    int64
 2   consensus_sex               672 non-null    object
 3   inferred_ethnicity          670 non-null    object
 4   isrelapse                   672 non-null    bool
 5   istransformed               672 non-null    bool
 6   priormalignancynonmyeloid   672 non-null    object
 7   priormds                    672 non-null    object
 8   priormdsmpn                 672 non-null    object
 9   priormpn                    672 non-null    object
 10  eln2017                     672 non-null    object
 11  dxatspecimenacquisition     672 non-null    object
 12  vitalstatus                 672 non-null    object
 13  overallsurvival             607 non-null    float64
 14  %.blasts.in.bm              459 non-null    object
 15  %.blasts.in.pb              451 non-null    object
 16  flt3-itd                    670 non-null    object
 17  npm1                        669 non-null    object
dtypes: bool(2), float64(1), int64(1), object(14)
memory usage: 85.4+ KB
```

In [30]:
```
clsm_cut.describe()
```

Out[30]:

	patientid	overallsurvival
count	672.000000	607.000000
mean	2088.020833	441.881384
std	973.372734	479.180429
min	17.000000	-1.000000
25%	1450.750000	167.000000
50%	2016.000000	323.000000
75%	2501.500000	555.000000
max	4380.000000	5305.000000

Attribute Information

% Blasts Attributes Numerical Prep

%.blasts.in.bm Attribute:

In [31]:
```
clsm_cut['%.blasts.in.bm'].unique()
```

Out[31]:
```
array(['94', '80', '91', '97', '87', nan, '40', '75', '83', '95', '85',
       '90', '70', '92', '72', '68', '88', '36', '81', '93', '34', '77.5',
       '46', '65', '50', '76', '71', '60', '73', '55', '0.5', '30', '62',
       '18', '82', '28', '41', '64', '84', '21', '51', '17', '49.4', '32',
       '29', '25', '59.3', '66', '20', '52', '54', '22', '10', '12', '13',
       '67', '39', '25.9', '45', '37', '78', '8', '3', '54.8', '74', '96',
       '4', '86.1', '42', '56', '69', '79', '33', '9', '0.4', '51.5',
       '15', '5', '24', '7', '2', '6', '1', '58', '>50', '35', '86',
       '93.2', '0', '27', '89.6', '23', '98', '19', '91.8', '>95', '57',
       '71.5', '78.3', '63', '1.5', '53.74', '59.5', '44', '42.5', '26',
       '3.5', '48', '26.3', '47', '88.5'], dtype=object)
```

In [32]:
```
# > and < will be changed to whole numbers less than or greater than.
clsm_cut['%.blasts.in.bm'] = clsm_cut['%.blasts.in.bm'].replace(['>50'], 51)
clsm_cut['%.blasts.in.bm'] = clsm_cut['%.blasts.in.bm'].replace(['>95'], 96)

clsm_cut['%.blasts.in.bm'].unique()
```

Out[32]:
```
array(['94', '80', '91', '97', '87', nan, '40', '75', '83', '95', '85',
       '90', '70', '92', '72', '68', '88', '36', '81', '93', '34', '77.5',
       '46', '65', '50', '76', '71', '60', '73', '55', '0.5', '30', '62',
       '18', '82', '28', '41', '64', '84', '21', '51', '17', '49.4', '32',
       '29', '25', '59.3', '66', '20', '52', '54', '22', '10', '12', '13',
       '67', '39', '25.9', '45', '37', '78', '8', '3', '54.8', '74', '96',
       '4', '86.1', '42', '56', '69', '79', '33', '9', '0.4', '51.5',
       '15', '5', '24', '7', '2', '6', '1', '58', 51, '35', '86', '93.2',
       '0', '27', '89.6', '23', '98', '19', '91.8', 96, '57', '71.5',
       '78.3', '63', '1.5', '53.74', '59.5', '44', '42.5', '26', '3.5',
       '48', '26.3', '47', '88.5'], dtype=object)
```

%.blasts.in.pb Attribute:

In [33]:
```
clsm_cut['%.blasts.in.pb'].unique()
```

Out[33]:
```
array(['97', '19', '99', '80', nan, '51', '30', '41', '84', '77', '75',
       '63', '60', '96', '66', '45', '93', '9', '82', '15', '33', '0',
       '13', '94', '89', '83', '>90', '78', '72', '59', '32', '6', '29',
       '24', '64', '57', '52', '2.1', '<5', '17', '22', '5', '47', '56',
       '25', '23', '42', '65', '71', '8', '3.5', '66.3', '95', '44', '10',
       '28.6', '18', '58', '67', '40', '92', '54', '1', '2', '20', '28',
       '35', '85', '42.4', '16', '49.1', '14', '88', '46', '7', '0.5',
       '79', '26', '87', '20.4', '68', '48', '5.3', '61', '90', '17.4',
       '57.4', '43.8', '50', '37', '4', '3', '12', '81', '11', '90.5',
       '"rare"', '90.2', '55', 'rare', '39', '31', '86', '47.4', '27.4',
       '39.6', '12.9', '15.4', '9.5', '62', '64.6', '27.8', '69.14',
       '52.2', '91', '67.25', '49', '23.7', '48.6', '98', '74.8', '2.6',
       '43', '29.6', '47.5', '38', '2.5', '25.2', '3.56', '70', '99.2',
       '73', '26.7', '38.5', '7.7', '74', '93.3', '12.1', '11.2', '92.9',
       '98.4', '6.8', '10.5', '53', '3.1', '28.9', '72.9', '40.2', '3.3',
       '42.1', '11.5', '77.8', '3.8', '59.5', '21.7', '53.2'],
      dtype=object)
```

In [34]:
```
#%.Blasts.in.PB attribute has 1 "rare" record with no flt3 nor npm1 input. This will be changed to NAN
clsm_cut['%.blasts.in.pb'] = clsm_cut['%.blasts.in.pb'].replace(['"""rare"""'], np.nan)
clsm_cut['%.blasts.in.pb'] = clsm_cut['%.blasts.in.pb'].replace(['"rare"'], np.nan)
clsm_cut['%.blasts.in.pb'] = clsm_cut['%.blasts.in.pb'].replace(['rare'], np.nan)
# > and < will be changed to whole numbers less than or greater than.
clsm_cut['%.blasts.in.pb'] = clsm_cut['%.blasts.in.pb'].replace(['<5'], 4)
clsm_cut['%.blasts.in.pb'] = clsm_cut['%.blasts.in.pb'].replace(['>90'], 91)

clsm_cut['%.blasts.in.pb'].unique()
```

Out[34]:
```
array(['97', '19', '99', '80', nan, '51', '30', '41', '84', '77', '75',
       '63', '60', '96', '66', '45', '93', '9', '82', '15', '33', '0',
       '13', '94', '89', '83', 91, '78', '72', '59', '32', '6', '29',
       '24', '64', '57', '52', '2.1', 4, '17', '22', '5', '47', '56',
       '25', '23', '42', '65', '71', '8', '3.5', '66.3', '95', '44', '10',
       '28.6', '18', '58', '67', '40', '92', '54', '1', '2', '20', '28',
       '35', '85', '42.4', '16', '49.1', '14', '88', '46', '7', '0.5',
       '79', '26', '87', '20.4', '68', '48', '5.3', '61', '90', '17.4',
       '57.4', '43.8', '50', '37', '4', '3', '12', '81', '11', '90.5',
       '90.2', '55', '39', '31', '86', '47.4', '27.4', '39.6', '12.9',
       '15.4', '9.5', '62', '64.6', '27.8', '69.14', '52.2', '91',
       '67.25', '49', '23.7', '48.6', '98', '74.8', '2.6', '43', '29.6',
       '47.5', '38', '2.5', '25.2', '3.56', '70', '99.2', '73', '26.7',
       '38.5', '7.7', '74', '93.3', '12.1', '11.2', '92.9', '98.4', '6.8',
       '10.5', '53', '3.1', '28.9', '72.9', '40.2', '3.3', '42.1', '11.5',
       '77.8', '3.8', '59.5', '21.7', '53.2'], dtype=object)
```

From Categorical to Numerical

Transform %.blasts.in.bm and %.blasts.in.pb from object to float:

In [35]:
```
clsm_cut['%.blasts.in.bm'] = clsm_cut['%.blasts.in.bm'].astype(float)
clsm_cut['%.blasts.in.pb'] = clsm_cut['%.blasts.in.pb'].astype(float)


clsm_cut.info()
```

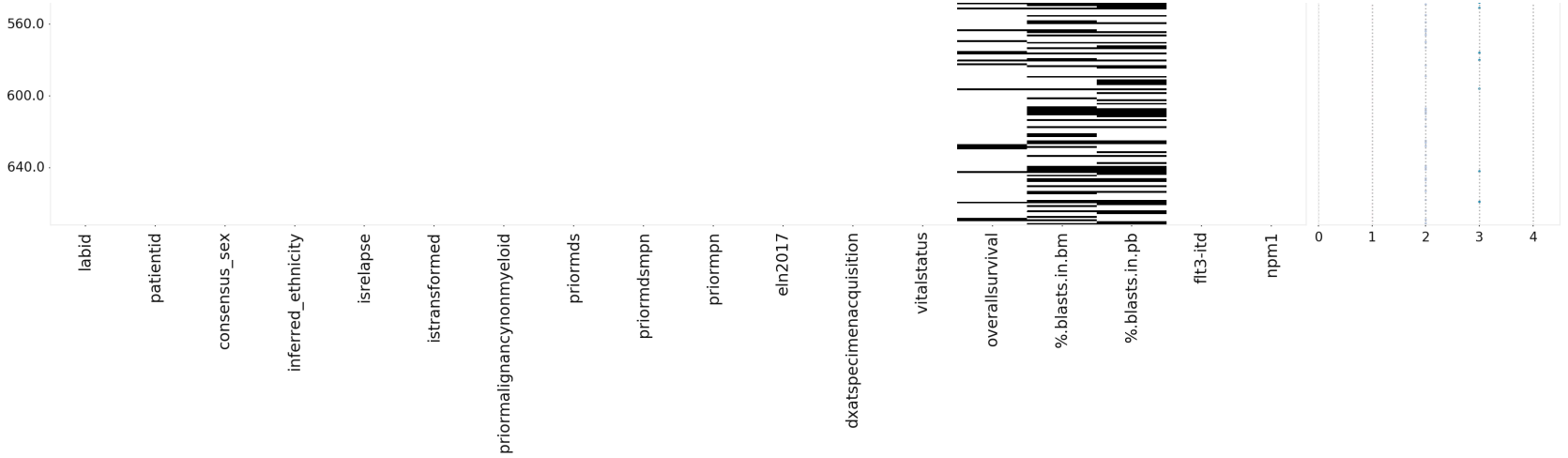
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 672 entries, 0 to 671
Data columns (total 18 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   labid                       672 non-null    object
 1   patientid                   672 non-null    int64
 2   consensus_sex               672 non-null    object
 3   inferred_ethnicity          670 non-null    object
 4   isrelapse                   672 non-null    bool
 5   istransformed               672 non-null    bool
 6   priormalignancynonmyeloid   672 non-null    object
 7   priormds                    672 non-null    object
 8   priormdsmpn                 672 non-null    object
 9   priormpn                    672 non-null    object
 10  eln2017                     672 non-null    object
 11  dxatspecimenacquisition     672 non-null    object
 12  vitalstatus                 672 non-null    object
 13  overallsurvival             607 non-null    float64
 14  %.blasts.in.bm              459 non-null    float64
 15  %.blasts.in.pb              448 non-null    float64
 16  flt3-itd                    670 non-null    object
 17  npm1                        669 non-null    object
dtypes: bool(2), float64(3), int64(1), object(12)
memory usage: 85.4+ KB
```

clsm_cut Identify Missing Values

In [36]:
```
klib.missingval_plot(clsm_cut)
```

Out[36]:
```
GridSpec(6, 6)
```

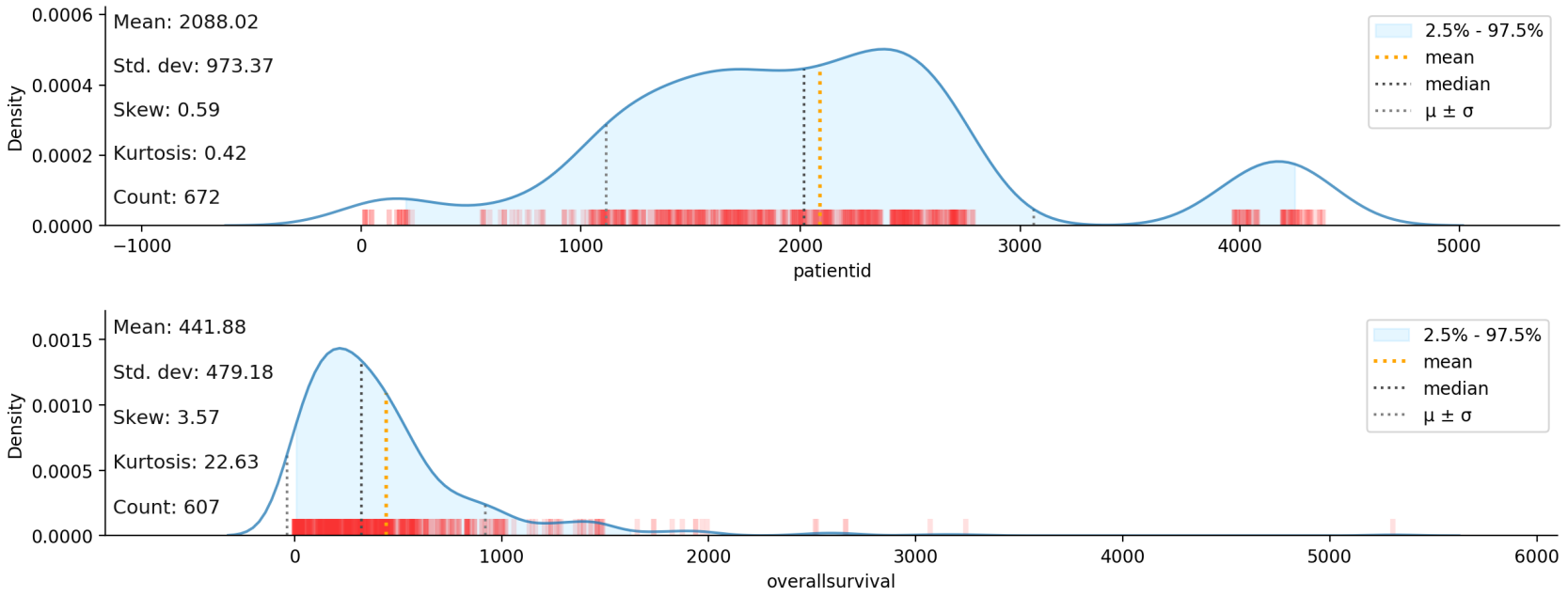
Missing value plot

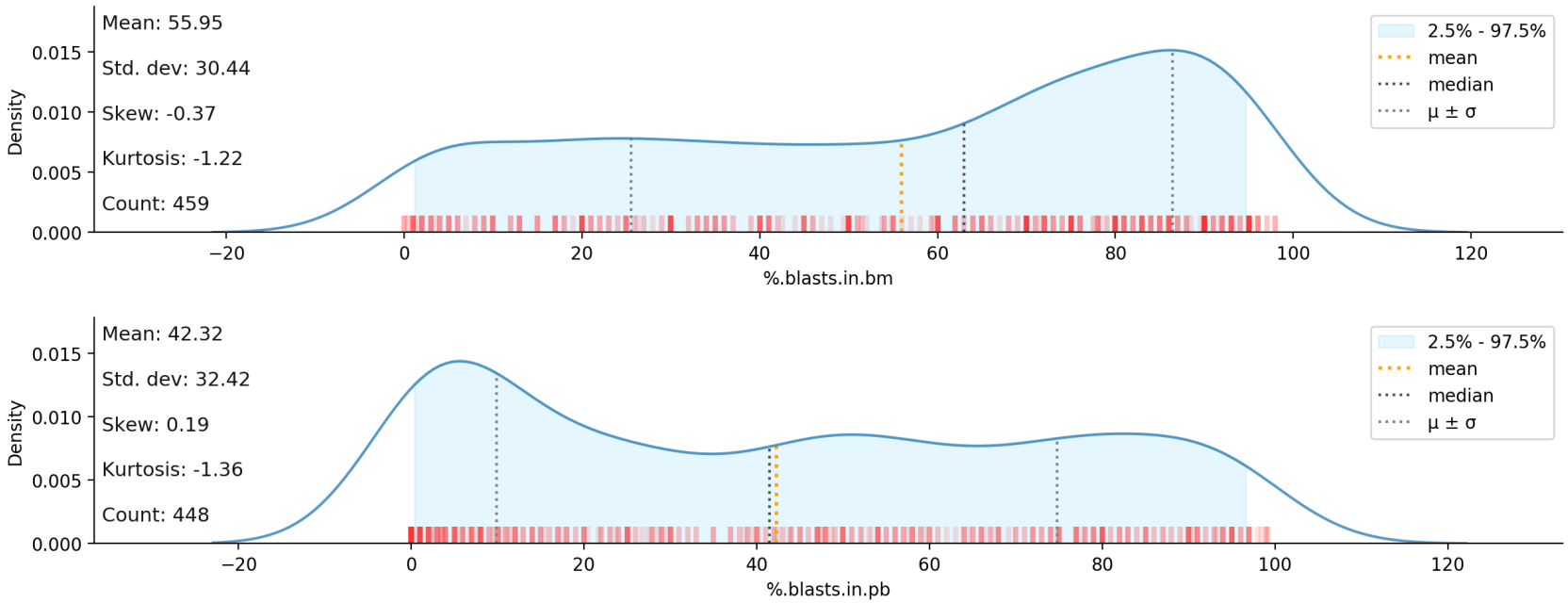


Replace Missing Values

In [37]:
```
klib.dist_plot(clsm_cut)
```

Out[37]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fceaa50a350>
```





In [38]:
```
clsm_cut.describe()
```

Out[38]:

	patientid	overallsurvival	%.blasts.in.bm	%.blasts.in.pb
count	672.000000	607.000000	459.000000	448.000000
mean	2088.020833	441.881384	55.949325	42.316629
std	973.372734	479.180429	30.440925	32.418249
min	17.000000	-1.000000	0.000000	0.000000
25%	1450.750000	167.000000	30.000000	10.000000
50%	2016.000000	323.000000	63.000000	41.500000
75%	2501.500000	555.000000	83.000000	72.000000
max	4380.000000	5305.000000	98.000000	99.200000

In [39]:
```
#From distibution, skewness suggest median is the best representation.
clsm_cut['overallsurvival'] = clsm_cut['overallsurvival'].fillna(clsm_cut['overallsurvival'].median())
clsm_cut['%.blasts.in.bm'] = clsm_cut['%.blasts.in.bm'].fillna(clsm_cut['%.blasts.in.bm'].median())
clsm_cut['%.blasts.in.pb'] = clsm_cut['%.blasts.in.pb'].fillna(clsm_cut['%.blasts.in.pb'].median())
```

In [40]:
```
#Replace categorical NaN with unknown
clsm_cut = clsm_cut.replace(np.nan, 'unknown', regex=True)
```

In [41]:
```
#Determine mode of inferred_ethnicity:
clsm_cut['inferred_ethnicity'].mode()
```

Out[41]:
```
0    White
dtype: object
```

In [42]:
```
#In inferred_ethnicity, replace mode of unknown to white:
clsm_cut['inferred_ethnicity'] = clsm_cut['inferred_ethnicity'].replace(['unknown'], 'white')

clsm_cut['inferred_ethnicity'].unique()
```

Out[42]:
```
array(['White', 'HispNative', 'AdmixedBlack', 'Asian', 'Black',
       'AdmixedAsian', 'white', 'AdmixedWhite', 'AdmixedHispNative'],
      dtype=object)
```

In [43]:
```
#Determine mode of flt3-itd:
clsm_cut['flt3-itd'].mode()
```

Out[43]:
```
0    negative
dtype: object
```

In [44]:
```
#In flt3-itd, replace mode of unknown to negative:
clsm_cut['flt3-itd'] = clsm_cut['flt3-itd'].replace(['unknown'], 'negative')

clsm_cut['flt3-itd'].unique()
```

Out[44]:
```
array(['positive', 'negative'], dtype=object)
```

In [45]:
```
#Determine mode of npm1:
clsm_cut['npm1'].mode()
```

Out[45]:
```
0    negative
dtype: object
```

In [46]:
```
#In npm1, replace mode of unknown to negative:
clsm_cut['npm1'] = clsm_cut['npm1'].replace(['unknown'], 'negative')

clsm_cut['npm1'].unique()
```

Out[46]:
```
array(['positive', 'negative'], dtype=object)
```

In [47]:
```
klib.missingval_plot(clsm_cut)
```

```
No missing values found in the dataset.
```

In [48]:
```
clsm_cut.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 672 entries, 0 to 671
Data columns (total 18 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   labid                       672 non-null    object 
 1   patientid                   672 non-null    int64  
 2   consensus_sex               672 non-null    object 
 3   inferred_ethnicity          672 non-null    object 
 4   isrelapse                   672 non-null    bool   
 5   istransformed               672 non-null    bool   
 6   priormalignancynonmyeloid   672 non-null    object 
 7   priormds                    672 non-null    object 
 8   priormdsmpn                 672 non-null    object 
 9   priormpn                    672 non-null    object 
 10  eln2017                     672 non-null    object 
 11  dxatspecimenacquisition     672 non-null    object 
 12  vitalstatus                 672 non-null    object 
 13  overallsurvival             672 non-null    float64
 14  %.blasts.in.bm              672 non-null    float64
 15  %.blasts.in.pb              672 non-null    float64
 16  flt3-itd                    672 non-null    object 
 17  npm1                        672 non-null    object 
dtypes: bool(2), float64(3), int64(1), object(12)
memory usage: 85.4+ KB
```

Check for Duplicates

In [49]:
```
clsm_cut = clsm_cut.drop_duplicates(ignore_index=True)
clsm_cut.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 672 entries, 0 to 671
Data columns (total 18 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   labid                       672 non-null    object
 1   patientid                   672 non-null    int64
 2   consensus_sex               672 non-null    object
 3   inferred_ethnicity          672 non-null    object
 4   isrelapse                   672 non-null    bool
 5   istransformed               672 non-null    bool
 6   priormalignancynonmyeloid   672 non-null    object
 7   priormds                    672 non-null    object
 8   priormdsmpn                 672 non-null    object
 9   priormpn                    672 non-null    object
 10  eln2017                     672 non-null    object
 11  dxatspecimenacquisition     672 non-null    object
 12  vitalstatus                 672 non-null    object
 13  overallsurvival             672 non-null    float64
 14  %.blasts.in.bm              672 non-null    float64
 15  %.blasts.in.pb              672 non-null    float64
 16  flt3-itd                    672 non-null    object
 17  npm1                        672 non-null    object
dtypes: bool(2), float64(3), int64(1), object(12)
memory usage: 85.4+ KB
```

Transformation (Final Preparation before Data Modeling)

Create Target Variable

In [50]:
```
clsm_cut['dxatspecimenacquisition'].value_counts()
```

Out[50]:
```
ACUTE MYELOID LEUKAEMIA (AML) AND RELATED PRECURSOR NEOPLASMS    646
MYELODYSPLASTIC SYNDROMES                                         15
MYELODYSPLASTIC/MYELOPROLIFERATIVE NEOPLASMS                       4
ACUTE LEUKAEMIAS OF AMBIGUOUS LINEAGE                             3
MYELOPROLIFERATIVE NEOPLASMS                                      3
MATURE B-CELL NEOPLASMS                                           1
Name: dxatspecimenacquisition, dtype: int64
```

In [51]:
```
#create column for AML detected
clsm_cut['AML_detected'] = ['yes' if x == 'ACUTE MYELOID LEUKAEMIA (AML) AND RELATED PRECURSOR NEOPLASMS'
                            else 'no' for x in clsm_cut['dxatspecimenacquisition']]
```

In [52]:
```
clsm_cut.head()
```

Out[52]:

	labid	patientid	consensus_sex	inferred_ethnicity	isrelapse	istransformed	priormalignancynonmyeloid	priormds	priormdsmpn	p
0	09-00705	163	Male	White	False	False	n	n	n	
1	10-00136	174	Male	White	False	False	n	n	n	
2	10-00172	175	Female	White	False	False	n	n	n	
3	10-00507	45	Female	White	False	False	n	n	n	
4	10-00542	174	Male	White	True	False	n	n	n	

Transform select categorical attributes to numerical:

In [53]:
```
#AML_detected
clsm_cut['AML_detected'].replace(['no', 'yes'],
                         [0, 1], inplace=True)

#npm1
clsm_cut['npm1'].replace(['negative', 'positive'],
                         [0, 1], inplace=True)

#flt3-itd
clsm_cut['flt3-itd'].replace(['negative', 'positive'],
                         [0, 1], inplace=True)

#priormalignancynonmyeloid
clsm_cut['priormalignancynonmyeloid'].replace(['n', 'y'],
                         [0, 1], inplace=True)

#priormds
clsm_cut['priormds'].replace(['y', 'n'],
                         [1, 0], inplace=True)

#priormdsmpn
clsm_cut['priormdsmpn'].replace(['n', 'y'],
                         [0, 1], inplace=True)

#priormpn
clsm_cut['priormpn'].replace(['n', 'y'],
                         [0, 1], inplace=True)

#isrelapse
clsm_cut['isrelapse'].replace(['False', 'True'],
                         [0, 1], inplace=True)

#istransformed
clsm_cut['istransformed'].replace(['True', 'False'],
                         [1, 0], inplace=True)
```

In [54]:
```
clsm_t = pd.DataFrame(clsm_cut[['AML_detected', 'npm1', 'flt3-itd', 'isrelapse', 'istransformed',
                                'priormalignancynonmyeloid', 'priormds', 'priormdsmpn', 'priormpn',
                                '%.blasts.in.pb', '%.blasts.in.bm', 'overallsurvival']])
```

In [55]:
```
#Transform data type:
clsm_t['npm1'] = clsm_cut['npm1'].astype(int)
clsm_t['flt3-itd'] = clsm_cut['flt3-itd'].astype(int)

clsm_t['isrelapse'] = clsm_cut['isrelapse'].astype(int)
clsm_t['istransformed'] = clsm_cut['istransformed'].astype(int)
```

One Hot encoding

In [56]:
```
clsm_t = pd.get_dummies(clsm_t, columns= ['npm1', 'flt3-itd', 'priormalignancynonmyeloid',
                                          'priormds', 'priormdsmpn', 'priormpn', 'isrelapse', 'istransformed'
```

In [57]:
```
clsm_t.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 672 entries, 0 to 671
Data columns (total 20 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   AML_detected                  672 non-null    int64
 1   %.blasts.in.pb                672 non-null    float64
 2   %.blasts.in.bm                672 non-null    float64
 3   overallsurvival               672 non-null    float64
 4   npm1_0                        672 non-null    uint8
 5   npm1_1                        672 non-null    uint8
 6   flt3-itd_0                    672 non-null    uint8
 7   flt3-itd_1                    672 non-null    uint8
 8   priormalignancynonmyeloid_0   672 non-null    uint8
 9   priormalignancynonmyeloid_1   672 non-null    uint8
 10  priormds_0                    672 non-null    uint8
 11  priormds_1                    672 non-null    uint8
 12  priormdsmpn_0                 672 non-null    uint8
 13  priormdsmpn_1                 672 non-null    uint8
 14  priormpn_0                    672 non-null    uint8
 15  priormpn_1                    672 non-null    uint8
 16  isrelapse_0                   672 non-null    uint8
 17  isrelapse_1                   672 non-null    uint8
 18  istransformed_0               672 non-null    uint8
 19  istransformed_1               672 non-null    uint8
dtypes: float64(3), int64(1), uint8(16)
memory usage: 31.6 KB
```

In [58]:
```
clsm_t.head()
```

Out[58]:

	AML_detected	%.blasts.in.pb	%.blasts.in.bm	overallsurvival	npm1_0	npm1_1	flt3-itd_0	flt3-itd_1	priormalignancynonmyeloid_0	priormalign
0	1	97.0	94.0	425.0	0	1	0	1	1	
1	1	19.0	80.0	419.0	1	0	0	1	1	
2	1	99.0	91.0	541.0	1	0	0	1	1	
3	1	97.0	97.0	511.0	0	1	0	1	1	
4	1	80.0	87.0	419.0	1	0	0	1	1	

Transform Headers

In [59]:
```
clsm_t = clsm_t.rename(columns={ '%.blasts.in.pb': 'Feature_1', '%.blasts.in.bm': 'Feature_2',
                                 'overallsurvival': 'Feature_3',
                                  'npm1_0': 'Feature_4', 'npm1_1': 'Feature_5',
                                 'flt3-itd_0': 'Feature_6', 'flt3-itd_1': 'Feature_7',
                                  'priormalignancynonmyeloid_0': 'Feature_8', 'priormalignancynonmyeloid_1'
                                  'priormds_0': 'Feature_10', 'priormds_1': 'Feature_11',
                                 'priormdsmpn_0': 'Feature_12', 'priormdsmpn_1': 'Feature_13',
                                  'priormpn_0': 'Feature_14', 'priormpn_1': 'Feature_15',
                                  'isrelapse_0': 'Feature_16', 'isrelapse_1': 'Feature_17',
                                 'istransformed_0': 'Feature_18', 'istransformed_1': 'Feature_19' })
```

In [60]:
```
clsm_t.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 672 entries, 0 to 671
Data columns (total 20 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   AML_detected  672 non-null    int64  
 1   Feature_1     672 non-null    float64
 2   Feature_2     672 non-null    float64
 3   Feature_3     672 non-null    float64
 4   Feature_4     672 non-null    uint8  
 5   Feature_5     672 non-null    uint8  
 6   Feature_6     672 non-null    uint8  
 7   Feature_7     672 non-null    uint8  
 8   Feature_8     672 non-null    uint8  
 9   Feature_9     672 non-null    uint8  
 10  Feature_10    672 non-null    uint8  
 11  Feature_11    672 non-null    uint8  
 12  Feature_12    672 non-null    uint8  
 13  Feature_13    672 non-null    uint8  
 14  Feature_14    672 non-null    uint8  
 15  Feature_15    672 non-null    uint8  
 16  Feature_16    672 non-null    uint8  
 17  Feature_17    672 non-null    uint8  
 18  Feature_18    672 non-null    uint8  
 19  Feature_19    672 non-null    uint8  
dtypes: float64(3), int64(1), uint8(16)
memory usage: 31.6 KB
```

Auto ML

In [61]:
```
df_automl = clsm_t
clsm_t.shape
```

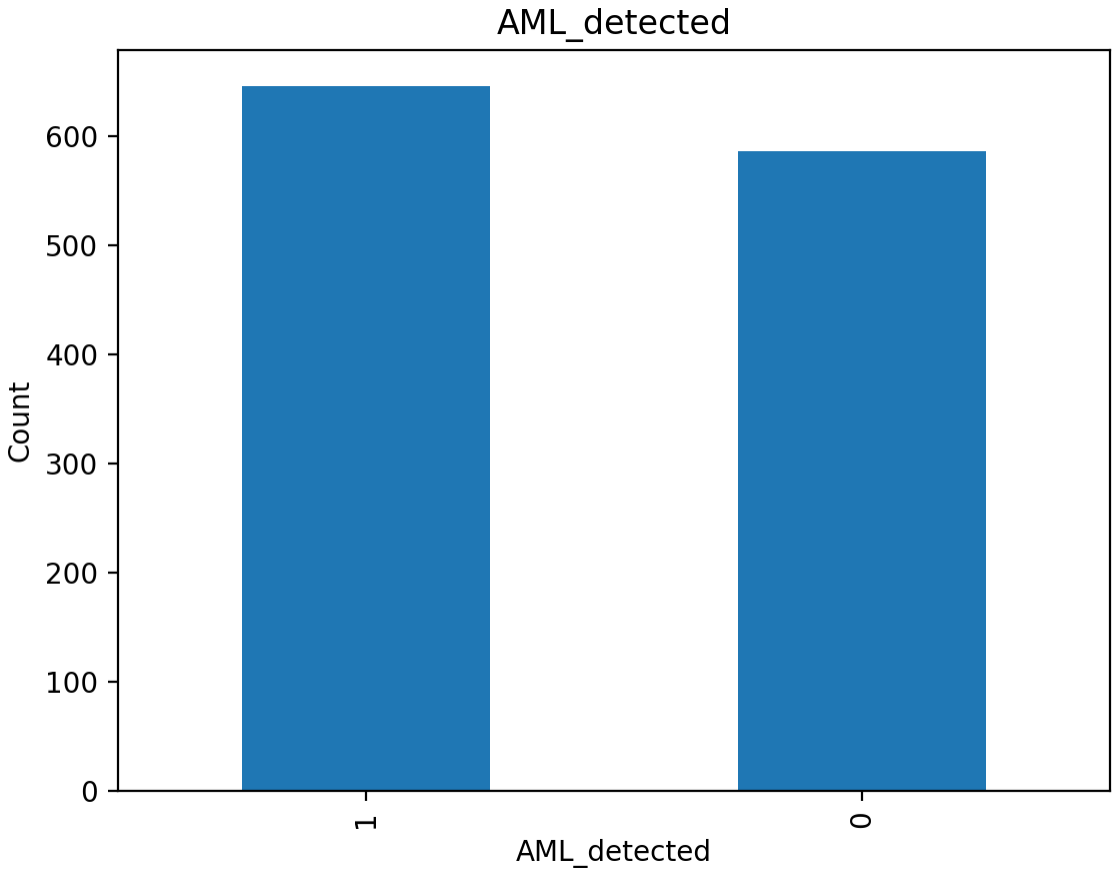
Out[61]:
```
(672, 20)
```

Balance the data set

In [62]:
```
#resampling of training data set
to_resample= df_automl.loc[df_automl["AML_detected"] == 0] #isolate all records of AML_detected
our_resample=to_resample.sample(n=560, replace=True) #sample w/ replacement
df_rebal=pd.concat([df_automl, our_resample]) #combine original training set w/ resampled records
df_rebal["AML_detected"].value_counts()
```

Out[62]:
```
1    646
0    586
Name: AML_detected, dtype: int64
```

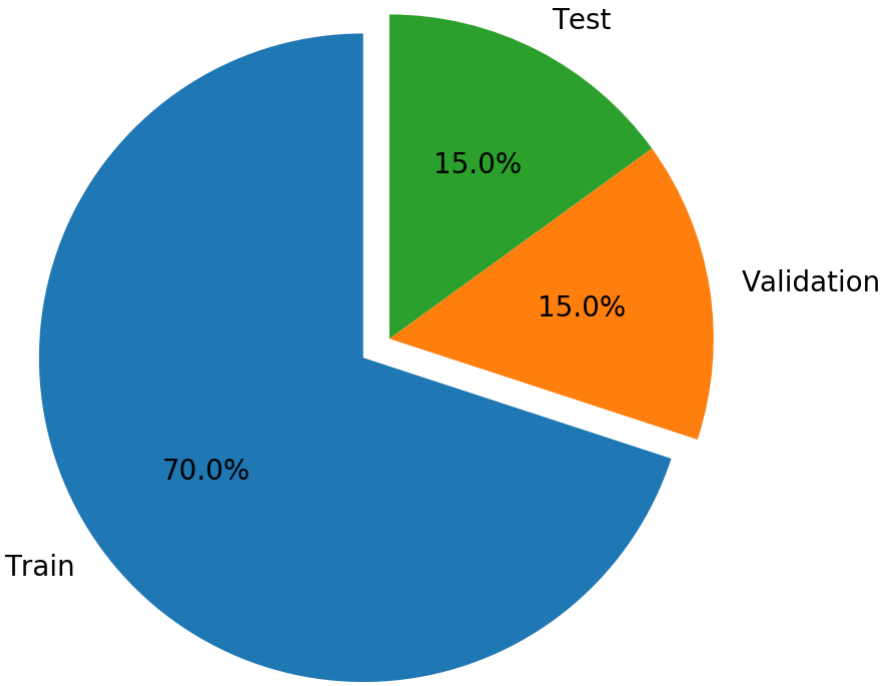
In [63]:
```
import matplotlib.pyplot as plt

df_rebal["AML_detected"].value_counts().plot(kind="bar", title="AML_detected")
plt.xlabel("AML_detected")
plt.ylabel("Count")

plt.show()
```



Split into Train, Validation, and Test Sets

In [64]:
```
from sklearn.model_selection import train_test_split

# Split all data into 70% train and 30% holdout
df_train, df_holdout = train_test_split(df_rebal, test_size=0.30, stratify=df_rebal["AML_detected"])

# Split holdout data into 50% validation and 50% test
df_validation, df_test = train_test_split(df_holdout, test_size=0.50, stratify=df_holdout["AML_detected"])
```

In [65]:
```
# Pie chart, where the slices will be ordered and plotted counter-clockwise:
labels = ["Train", "Validation", "Test"]
sizes = [len(df_train.index), len(df_validation.index), len(df_test.index)]
explode = (0.1, 0, 0)

fig1, ax1 = plt.subplots()

ax1.pie(sizes, explode=explode, labels=labels, autopct="%1.1f%%", startangle=90)

# Equal aspect ratio ensures that pie is drawn as a circle.
ax1.axis("equal")

plt.show()
```

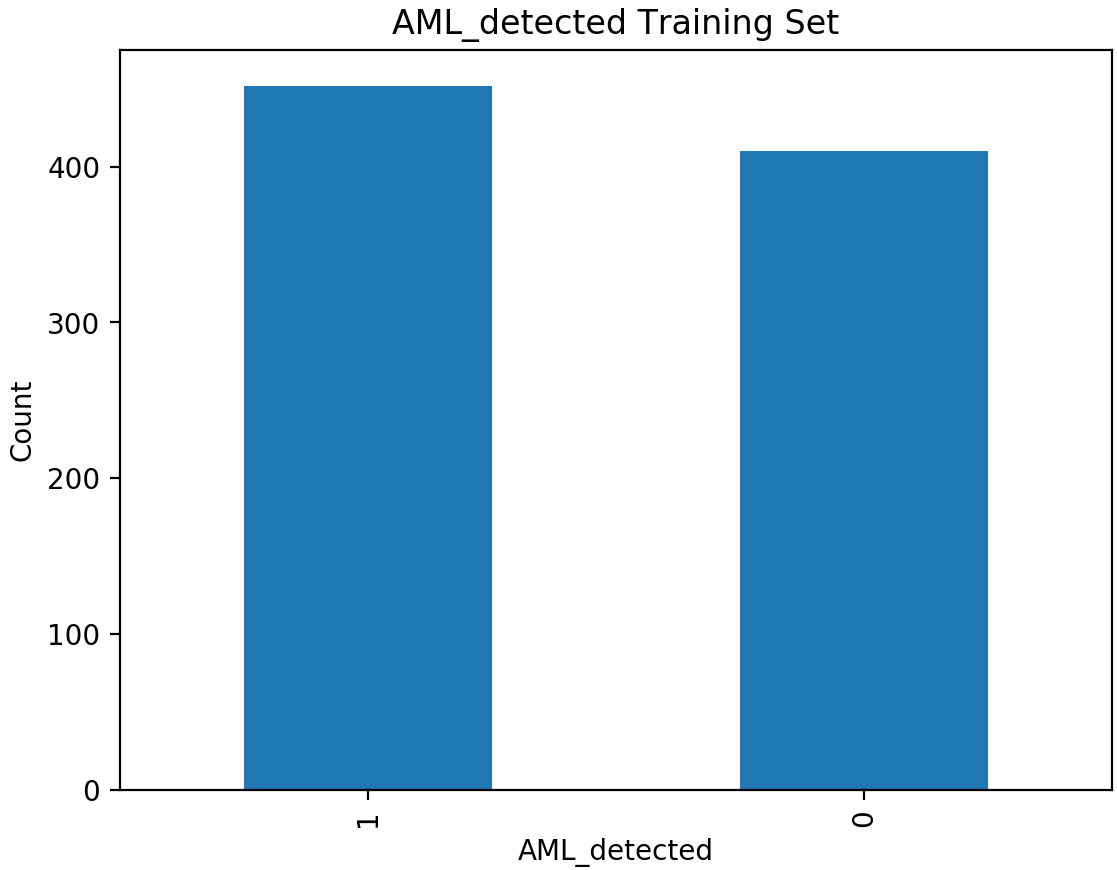


Show 70% Train Data Split

In [66]:
```
df_train.shape
```

Out[66]:
```
(862, 20)
```

In [67]:
```
df_train["AML_detected"].value_counts().plot(kind="bar", title="AML_detected Training Set")
plt.xlabel("AML_detected")
plt.ylabel("Count")

plt.show()
```

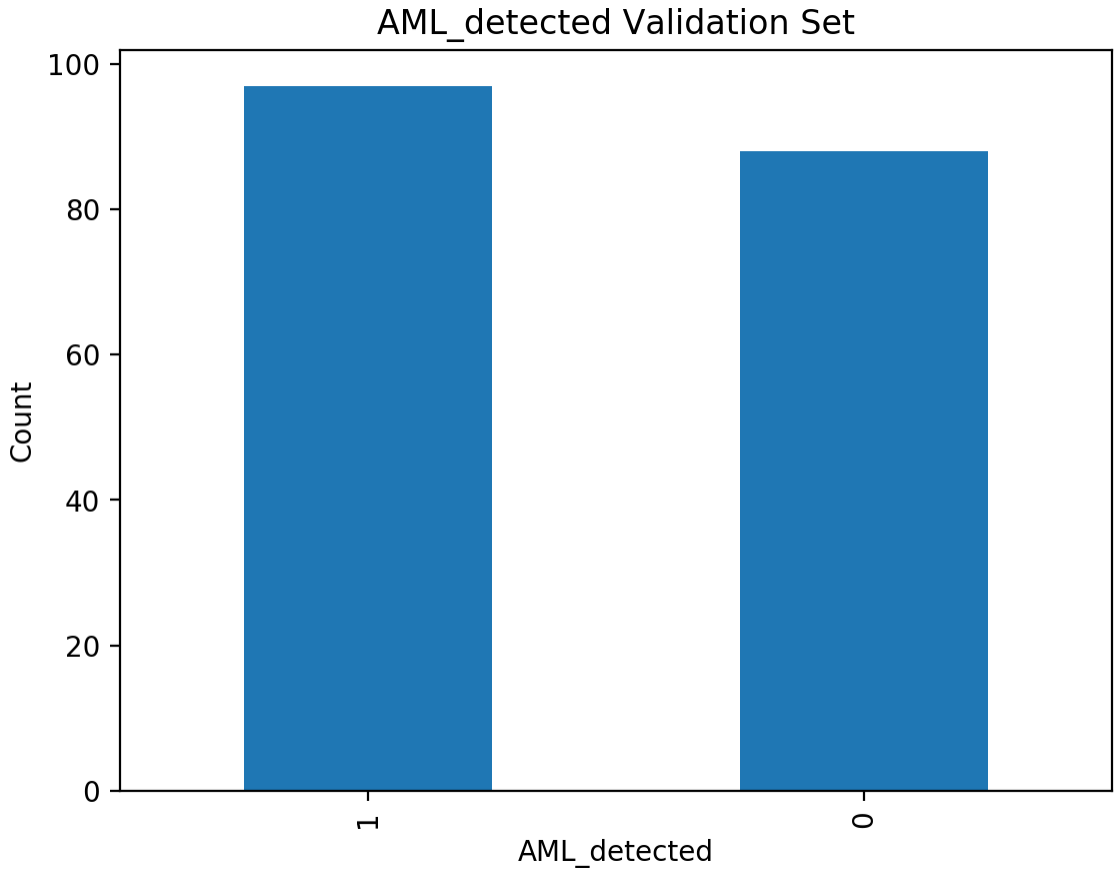


Show 15% Validation Split

In [68]:
```
df_validation.shape
```

Out[68]:
```
(185, 20)
```

In [69]:
```
df_validation["AML_detected"].value_counts().plot(kind="bar", title="AML_detected Validation Set")
plt.xlabel("AML_detected")
plt.ylabel("Count")

plt.show()
```

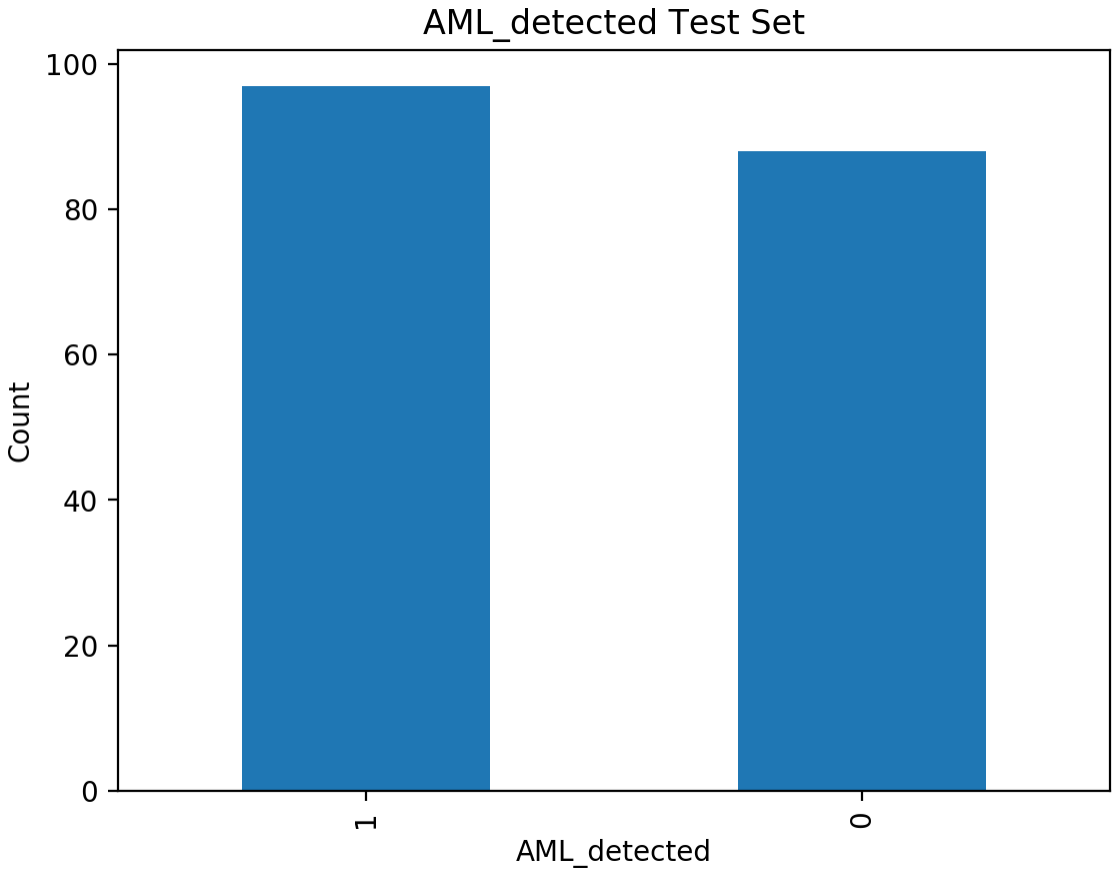


Show 15% Test Data Split

In [70]:
```
df_test.shape
```

Out[70]:
```
(185, 20)
```

In [71]:
```
df_test["AML_detected"].value_counts().plot(kind="bar", title="AML_detected Test Set")
plt.xlabel("AML_detected")
plt.ylabel("Count")

plt.show()
```



Write a Train CSV with Header for Autopilot

In [72]:
```
autopilot_train_path = "./df_autopilot.csv"
df_train.to_csv(autopilot_train_path, index=False, header=True)
```

Upload Train Data to S3 for Autopilot

In [73]:
```
train_s3_prefix = "data"
autopilot_train_s3_uri = sess.upload_data(path=autopilot_train_path, key_prefix=train_s3_prefix)
autopilot_train_s3_uri
```

Out[73]:
```
's3://sagemaker-us-east-1-614093401978/data/df_autopilot.csv'
```

In [74]:
```
!aws s3 ls $autopilot_train_s3_uri
```

```
2023-04-17 03:55:43      42710 df_autopilot.csv
```

Store Variables

In [75]:
```
%store autopilot_train_s3_uri
```

```
Stored 'autopilot_train_s3_uri' (str)
```

In [76]:
```
%store
```

```
Stored variables and their in-db values:
auto_ml_job_name                          -> 'automl-dm-16-20-10-05'
autopilot_endpoint_arn                    -> 'arn:aws:sagemaker:us-east-1:614093401978:endpoint
autopilot_endpoint_name                   -> 'automl-dm-ep-16-22-24-43'
autopilot_model_arn                       -> 'arn:aws:sagemaker:us-east-1:614093401978:model/au
autopilot_model_name                      -> 'automl-dm-model-16-22-20-58'
autopilot_train_s3_uri                    -> 's3://sagemaker-us-east-1-614093401978/data/df_aut
ingest_create_athena_db_passed            -> True
s3_private_path_csv                       -> 's3://sagemaker-us-east-1-614093401978/cell_data'
s3_public_path_clsm                       -> 's3://team4rawdatasets/CSV/Input/OHSU_BeatAML_Clin
s3_public_path_csv                        -> 's3://gdc-beataml1.0-crenolanib-phs001628-2-open/'
s3_public_path_pi                         -> 's3://team4rawdatasets/CSV/Input/OpenCell_ProteinI
setup_dependencies_passed                 -> True
setup_iam_roles_passed                    -> True
setup_instance_check_passed               -> True
setup_s3_bucket_passed                    -> True
```

Train AML Detection

Training Data

In [77]:
```
print(autopilot_train_s3_uri)
```

```
s3://sagemaker-us-east-1-614093401978/data/df_autopilot.csv
```

In [78]:
```
!aws s3 ls $autopilot_train_s3_uri
```

```
2023-04-17 03:55:43      42710 df_autopilot.csv
```

See our prepared traiing data which we use as input for Autopilot

In [79]:
```
!aws s3 cp $autopilot_train_s3_uri ./tmp/
```

```
download: s3://sagemaker-us-east-1-614093401978/data/df_autopilot.csv to tmp/df_autopilot.csv
```

In [80]:
```
import csv

df = pd.read_csv("./tmp/df_autopilot.csv")
df.head()
```

Out[80]:

	AML_detected	Feature_1	Feature_2	Feature_3	Feature_4	Feature_5	Feature_6	Feature_7	Feature_8	Feature_9	Feature_10	Feature_1
0	0	6.0	4.0	286.0	1	0	1	0	1	0	0	
1	0	15.0	63.0	299.0	1	0	1	0	1	0	0	
2	1	41.5	45.0	323.0	1	0	1	0	0	1	1	
3	0	0.0	5.0	189.0	1	0	1	0	1	0	0	
4	1	0.0	51.5	414.0	0	1	1	0	1	0	1	

Setup the S3 location for the Autopilot-Generated Assests

This includes Jupyter Notebooks (analysis), Python Scripts (Feature Engineering), and Trained Models

In [81]:
```
prefix_model_output = "models/autopilot"

model_output_s3_uri = "s3://{}/{}".format(bucket, prefix_model_output)

print(model_output_s3_uri)
```

```
s3://sagemaker-us-east-1-614093401978/models/autopilot
```

In [82]:
```
max_candidates = 3

job_config = {
    "CompletionCriteria": {
        "MaxRuntimePerTrainingJobInSeconds": 900,
        "MaxCandidates": max_candidates,
        "MaxAutoMLJobRuntimeInSeconds": 5400,
    },
}

input_data_config = [
    {
        "DataSource": {"S3DataSource": {"S3DataType": "S3Prefix", "S3Uri": "{}".format(autopilot_train_s3_uri)}},
        "TargetAttributeName": "AML_detected",
    }
]

output_data_config = {"S3OutputPath": "{}".format(model_output_s3_uri)}
```

Check for existing Autopilot jobs

In [83]:
```
existing_jobs_response = sm.list_auto_ml_jobs()
```

In [84]:
```
num_existing_jobs = 0
running_jobs = 0

if "AutoMLJobSummaries" in existing_jobs_response.keys():
    job_list = existing_jobs_response["AutoMLJobSummaries"]
    num_existing_jobs = len(job_list)
    # print('[INFO] You already created {} Autopilot job(s) in this account.'.format(num_existing_jobs))
    for j in job_list:
        if "AutoMLJobStatus" in j.keys():
            if j["AutoMLJobStatus"] == "InProgress":
                running_jobs = running_jobs + 1
    print("[INFO] You have {} Autopilot job(s) currently running << Should be 0 jobs.".format(running_jobs))
else:
    print("[OK] Please continue.")
```

```
[INFO] You have 0 Autopilot job(s) currently running << Should be 0 jobs.
```

Launch Sagemaker Autopilot Job

In [85]:
```
from time import gmtime, strftime, sleep
```

In [86]:
```
%store -r auto_ml_job_name

try:
    auto_ml_job_name
except NameError:
    timestamp_suffix = strftime("%d-%H-%M-%S", gmtime())
    auto_ml_job_name = "automl-dm-" + timestamp_suffix
    print("Created AutoMLJobName: " + auto_ml_job_name)
```

In [87]:
```
print(auto_ml_job_name)
```

```
automl-dm-16-20-10-05
```

In [88]:
```
%store auto_ml_job_name
```

```
Stored 'auto_ml_job_name' (str)
```

In [89]:
```
max_running_jobs = 1

if running_jobs < max_running_jobs:  # Limiting to max. 1 Jobs
    try:
        sm.create_auto_ml_job(
            AutoMLJobName=auto_ml_job_name,
            InputDataConfig=input_data_config,
            OutputDataConfig=output_data_config,
            AutoMLJobConfig=job_config,
            RoleArn=role,
        )
        print("[OK] Autopilot Job {} created.".format(auto_ml_job_name))
        running_jobs = running_jobs + 1
    except:
        print(
            "[INFO] You have already launched an Autopilot job. Please continue see the output of this job.".format(
                running_jobs
            )
        )
else:
    print(
        "[INFO] You have already launched {} Autopilot running job(s). Please continue see the output of the running
            running_jobs
        )
    )
```

```
[INFO] You have already launched an Autopilot job. Please continue see the output of this job.
```

Analyzing Data and Generate Notebooks

In [90]:
```
job_description_response = sm.describe_auto_ml_job(AutoMLJobName=auto_ml_job_name)

while (
    "AutoMLJobStatus" not in job_description_response.keys()
    and "AutoMLJobSecondaryStatus" not in job_description_response.keys()
):
    job_description_response = sm.describe_auto_ml_job(AutoMLJobName=auto_ml_job_name)
    print("[INFO] Autopilot Job has not yet started. Please wait. ")
    print(json.dumps(job_description_response, indent=4, sort_keys=True, default=str))
    print("[INFO] Waiting for Autopilot Job to start...")
    sleep(15)

print("[OK] AutoMLJob started.")
```

```
[OK] AutoMLJob started.
```

Review the Sagemaker Processing Jobs

In [91]:
```
from IPython.core.display import display, HTML

display(
    HTML(
        '<b>Review <a target="blank" href="https://console.aws.amazon.com/sagemaker/home?region={}#/processing-jobs/'
            region
        )
    )
)
```

Review Processing Jobs

The next cell will show InProgress for a few minutes

In [92]:
```
%%time

job_status = job_description_response["AutoMLJobStatus"]
job_sec_status = job_description_response["AutoMLJobSecondaryStatus"]

if job_status not in ("Stopped", "Failed"):
    while job_status in ("InProgress") and job_sec_status in ("Starting", "AnalyzingData"):
        job_description_response = sm.describe_auto_ml_job(AutoMLJobName=auto_ml_job_name)
        job_status = job_description_response["AutoMLJobStatus"]
        job_sec_status = job_description_response["AutoMLJobSecondaryStatus"]
        print(job_status, job_sec_status)
        sleep(15)
    print("[OK] Data analysis phase completed.\n")

print(json.dumps(job_description_response, indent=4, sort_keys=True, default=str))
```

```
[OK] Data analysis phase completed.

{
    "AutoMLJobArn": "arn:aws:sagemaker:us-east-1:614093401978:automl-job/automl-dm-16-20-10-05",
    "AutoMLJobArtifacts": {
        "CandidateDefinitionNotebookLocation": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-
20-10-05/sagemaker-automl-candidates/automl-dm-16-20-10-05-pr-1-4b26f436ffcc4edaace14844f354501a5785/notebooks/SageM
akerAutopilotCandidateDefinitionNotebook.ipynb",
        "DataExplorationNotebookLocation": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-1
0-05/sagemaker-automl-candidates/automl-dm-16-20-10-05-pr-1-4b26f436ffcc4edaace14844f354501a5785/notebooks/SageMaker
AutopilotDataExplorationNotebook.ipynb"
    },
    "AutoMLJobConfig": {
        "CompletionCriteria": {
            "MaxAutoMLJobRuntimeInSeconds": 5400,
            "MaxCandidates": 3,
            "MaxRuntimePerTrainingJobInSeconds": 900
        }
    },
    "AutoMLJobName": "automl-dm-16-20-10-05",
    "AutoMLJobSecondaryStatus": "Completed",
    "AutoMLJobStatus": "Completed",
    "BestCandidate": {
        "CandidateName": "automl-dm-16-20-10-054If5RqjsyAN-001-61c635e4",
        "CandidateProperties": {
            "CandidateArtifactLocations": {
                "Explainability": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/docu
mentation/explainability/output",
                "ModelInsights": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/docum
entation/model_monitor/output"
            },
            "CandidateMetrics": [
                {
                    "MetricName": "F1",
                    "Set": "Validation",
                    "StandardMetricName": "F1",
                    "Value": 0.995169997215271
                },
                {
                    "MetricName": "LogLoss",
                    "Set": "Validation",
                    "StandardMetricName": "LogLoss",
                    "Value": 0.11816000193357468
                },
```

```
                {
                    "MetricName": "Recall",
                    "Set": "Validation",
                    "StandardMetricName": "Recall",
                    "Value": 1.0
                },
                {
                    "MetricName": "Precision",
                    "Set": "Validation",
                    "StandardMetricName": "Precision",
                    "Value": 0.9904199838638306
                },
                {
                    "MetricName": "AUC",
                    "Set": "Validation",
                    "StandardMetricName": "AUC",
                    "Value": 0.9990599751472473
                },
                {
                    "MetricName": "Accuracy",
                    "Set": "Validation",
                    "StandardMetricName": "Accuracy",
                    "Value": 0.9953600168228149
                },
                {
                    "MetricName": "BalancedAccuracy",
                    "Set": "Validation",
                    "StandardMetricName": "BalancedAccuracy",
                    "Value": 0.9955800175666809
                }
            ]
        },
        "CandidateStatus": "Completed",
        "CandidateSteps": [
            {
                "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:processing-job/automl-dm-16-20-10-05-d
b-1-2887815310fd4416804d2079485d7d062e1a",
                "CandidateStepName": "automl-dm-16-20-10-05-db-1-2887815310fd4416804d2079485d7d062e1a",
                "CandidateStepType": "AWS::SageMaker::ProcessingJob"
            },
            {
                "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:training-job/automl-dm-16-20-10-05-dpp
1-1-8411a0fbc81748a9958acf62493120d4d7",
                "CandidateStepName": "automl-dm-16-20-10-05-dpp1-1-8411a0fbc81748a9958acf62493120d4d7",
```

```
                "CandidateStepType": "AWS::SageMaker::TrainingJob"
            },
            {
                "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:transform-job/automl-dm-16-20-10-05-dp
p1-csv-1-18357715aec34f7eb4c3b75bad6e49",
                "CandidateStepName": "automl-dm-16-20-10-05-dpp1-csv-1-18357715aec34f7eb4c3b75bad6e49",
                "CandidateStepType": "AWS::SageMaker::TransformJob"
            },
            {
                "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:training-job/automl-dm-16-20-10-054If5
RqjsyAN-001-61c635e4",
                "CandidateStepName": "automl-dm-16-20-10-054If5RqjsyAN-001-61c635e4",
                "CandidateStepType": "AWS::SageMaker::TrainingJob"
            }
        ],
        "CreationTime": "2023-04-16 20:33:00+00:00",
        "EndTime": "2023-04-16 20:38:38+00:00",
        "FinalAutoMLJobObjectiveMetric": {
            "MetricName": "validation:f1_binary",
            "StandardMetricName": "F1",
            "Value": 0.995169997215271
        },
        "InferenceContainers": [
            {
                "Environment": {
                    "AUTOML_TRANSFORM_MODE": "feature-transform",
                    "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "application/x-recordio-protobuf",
                    "SAGEMAKER_PROGRAM": "sagemaker_serve",
                    "SAGEMAKER_SUBMIT_DIRECTORY": "/opt/ml/model/code"
                },
                "Image": "683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-sklearn-automl:2.5-1-cpu-py3",
                "ModelDataUrl": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/data-p
rocessor-models/automl-dm-16-20-10-05-dpp1-1-8411a0fbc81748a9958acf62493120d4d7/output/model.tar.gz"
            },
            {
                "Environment": {
                    "MAX_CONTENT_LENGTH": "20971520",
                    "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "text/csv",
                    "SAGEMAKER_INFERENCE_OUTPUT": "predicted_label",
                    "SAGEMAKER_INFERENCE_SUPPORTED": "predicted_label,probability,probabilities"
                },
                "Image": "683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-xgboost:1.3-1-cpu-py3",
                "ModelDataUrl": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/tuning
/automl-dm--dpp1-xgb/automl-dm-16-20-10-054If5RqjsyAN-001-61c635e4/output/model.tar.gz"
```

```
            },
            {
                "Environment": {
                    "AUTOML_TRANSFORM_MODE": "inverse-label-transform",
                    "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "text/csv",
                    "SAGEMAKER_INFERENCE_INPUT": "predicted_label",
                    "SAGEMAKER_INFERENCE_OUTPUT": "predicted_label",
                    "SAGEMAKER_INFERENCE_SUPPORTED": "predicted_label,probability,labels,probabilities",
                    "SAGEMAKER_PROGRAM": "sagemaker_serve",
                    "SAGEMAKER_SUBMIT_DIRECTORY": "/opt/ml/model/code"
                },
                "Image": "683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-sklearn-automl:2.5-1-cpu-py3",
                "ModelDataUrl": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/data-p
rocessor-models/automl-dm-16-20-10-05-dpp1-1-8411a0fbc81748a9958acf62493120d4d7/output/model.tar.gz"
            }
        ],
        "LastModifiedTime": "2023-04-16 20:39:21.759000+00:00",
        "ObjectiveStatus": "Succeeded"
    },
    "CreationTime": "2023-04-16 20:14:58.074000+00:00",
    "EndTime": "2023-04-16 20:47:31.359000+00:00",
    "GenerateCandidateDefinitionsOnly": false,
    "InputDataConfig": [
        {
            "ChannelType": "training",
            "ContentType": "text/csv;header=present",
            "DataSource": {
                "S3DataSource": {
                    "S3DataType": "S3Prefix",
                    "S3Uri": "s3://sagemaker-us-east-1-614093401978/data/df_autopilot.csv"
                }
            },
            "TargetAttributeName": "AML_detected"
        }
    ],
    "LastModifiedTime": "2023-04-16 20:47:31.403000+00:00",
    "OutputDataConfig": {
        "S3OutputPath": "s3://sagemaker-us-east-1-614093401978/models/autopilot"
    },
    "ResolvedAttributes": {
        "AutoMLJobObjective": {
            "MetricName": "F1"
        },
        "CompletionCriteria": {
```

```
            "MaxAutoMLJobRuntimeInSeconds": 5400,
            "MaxCandidates": 3,
            "MaxRuntimePerTrainingJobInSeconds": 900
        },
        "ProblemType": "BinaryClassification"
    },
    "ResponseMetadata": {
        "HTTPHeaders": {
            "content-length": "5897",
            "content-type": "application/x-amz-json-1.1",
            "date": "Mon, 17 Apr 2023 03:55:47 GMT",
            "x-amzn-requestid": "b1efaf91-fb54-4faa-9fa8-131aa73acd1e"
        },
        "HTTPStatusCode": 200,
        "RequestId": "b1efaf91-fb54-4faa-9fa8-131aa73acd1e",
        "RetryAttempts": 0
    },
    "RoleArn": "arn:aws:iam::614093401978:role/LabRole"
}
CPU times: user 725 µs, sys: 81 µs, total: 806 µs
Wall time: 787 µs
```


View Generated Notebook Samples

In [93]:

```python
job_description_response = sm.describe_auto_ml_job(AutoMLJobName=auto_ml_job_name)

while "AutoMLJobArtifacts" not in job_description_response.keys():
    job_description_response = sm.describe_auto_ml_job(AutoMLJobName=auto_ml_job_name)
    print("[INFO] Autopilot Job has not yet generated the artifacts. Please wait. ")
    print(json.dumps(job_description_response, indent=4, sort_keys=True, default=str))
    print("[INFO] Waiting for AutoMLJobArtifacts...")
    sleep(15)

print("[OK] AutoMLJobArtifacts generated.")
```

```
[OK] AutoMLJobArtifacts generated.
```

In [94]:
```
job_description_response = sm.describe_auto_ml_job(AutoMLJobName=auto_ml_job_name)

while "DataExplorationNotebookLocation" not in job_description_response["AutoMLJobArtifacts"].keys():
    job_description_response = sm.describe_auto_ml_job(AutoMLJobName=auto_ml_job_name)
    print("[INFO] Autopilot Job has not yet generated the notebooks. Please wait. ")
    print(json.dumps(job_description_response, indent=4, sort_keys=True, default=str))
    print("[INFO] Waiting for DataExplorationNotebookLocation...")
    sleep(15)

print("[OK] DataExplorationNotebookLocation found.")
```

```
[OK] DataExplorationNotebookLocation found.
```

In [95]:
```
generated_resources = job_description_response["AutoMLJobArtifacts"]["DataExplorationNotebookLocation"]
download_path = generated_resources.rsplit("/notebooks/SageMakerAutopilotDataExplorationNotebook.ipynb")[0]
job_id = download_path.rsplit("/", 1)[-1]
```

In [96]:
```
from IPython.core.display import display, HTML

if not job_id:
    print("No AutoMLJobArtifacts found.")
else:
    display(
        HTML(
            '<b>Review <a target="blank" href="https://s3.console.aws.amazon.com/s3/buckets/{}/{}/{}/sagemaker-automl
                bucket, prefix_model_output, auto_ml_job_name, job_id
            )
        )
    )
```

Review S3 Generated Resources

Download Generated Notebooks and code

In [97]:
```
print(download_path)
```

```
s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/sagemaker-automl-candidates/automl-dm-16-20-10-05-pr-1-4b26f436ffcc4edaace14844f354501a5785
```

In [98]:
```
try:
    !aws s3 cp --recursive $download_path .
except:
    print('Could not download the generated resources. Make sure the path is correct.')
```

```
download: s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/sagemaker-automl-candidates/a
utoml-dm-16-20-10-05-pr-1-4b26f436ffcc4edaace14844f354501a5785/generated_module/MANIFEST.in to generated_module/MANI
FEST.in
download: s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/sagemaker-automl-candidates/a
utoml-dm-16-20-10-05-pr-1-4b26f436ffcc4edaace14844f354501a5785/generated_module/candidate_data_processors/dpp0.py to
generated_module/candidate_data_processors/dpp0.py
download: s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/sagemaker-automl-candidates/a
utoml-dm-16-20-10-05-pr-1-4b26f436ffcc4edaace14844f354501a5785/generated_module/candidate_data_processors/trainer.py
to generated_module/candidate_data_processors/trainer.py
download: s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/sagemaker-automl-candidates/a
utoml-dm-16-20-10-05-pr-1-4b26f436ffcc4edaace14844f354501a5785/generated_module/candidate_data_processors/dpp2.py to
generated_module/candidate_data_processors/dpp2.py
download: s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/sagemaker-automl-candidates/a
utoml-dm-16-20-10-05-pr-1-4b26f436ffcc4edaace14844f354501a5785/generated_module/README.md to generated_module/READM
E.md
download: s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/sagemaker-automl-candidates/a
utoml-dm-16-20-10-05-pr-1-4b26f436ffcc4edaace14844f354501a5785/generated_module/candidate_data_processors/sagemaker_
serve.py to generated_module/candidate_data_processors/sagemaker_serve.py
download: s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/sagemaker-automl-candidates/a
utoml-dm-16-20-10-05-pr-1-4b26f436ffcc4edaace14844f354501a5785/generated_module/setup.py to generated_module/setup.p
y
download: s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/sagemaker-automl-candidates/a
utoml-dm-16-20-10-05-pr-1-4b26f436ffcc4edaace14844f354501a5785/notebooks/SageMakerAutopilotCandidateDefinitionNotebo
ok.ipynb to notebooks/SageMakerAutopilotCandidateDefinitionNotebook.ipynb
download: s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/sagemaker-automl-candidates/a
utoml-dm-16-20-10-05-pr-1-4b26f436ffcc4edaace14844f354501a5785/generated_module/candidate_data_processors/dpp1.py to
generated_module/candidate_data_processors/dpp1.py
download: s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/sagemaker-automl-candidates/a
utoml-dm-16-20-10-05-pr-1-4b26f436ffcc4edaace14844f354501a5785/notebooks/sagemaker_automl/README.md to notebooks/sag
emaker_automl/README.md
download: s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/sagemaker-automl-candidates/a
utoml-dm-16-20-10-05-pr-1-4b26f436ffcc4edaace14844f354501a5785/notebooks/sagemaker_automl/common.py to notebooks/sag
emaker_automl/common.py
download: s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/sagemaker-automl-candidates/a
utoml-dm-16-20-10-05-pr-1-4b26f436ffcc4edaace14844f354501a5785/notebooks/sagemaker_automl/__init__.py to notebooks/s
agemaker_automl/__init__.py
download: s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/sagemaker-automl-candidates/a
utoml-dm-16-20-10-05-pr-1-4b26f436ffcc4edaace14844f354501a5785/notebooks/sagemaker_automl/local_candidate.py to note
books/sagemaker_automl/local_candidate.py
download: s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/sagemaker-automl-candidates/a
utoml-dm-16-20-10-05-pr-1-4b26f436ffcc4edaace14844f354501a5785/notebooks/sagemaker_automl/interactive_runner.py to n
otebooks/sagemaker_automl/interactive_runner.py
download: s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/sagemaker-automl-candidates/a
utoml-dm-16-20-10-05-pr-1-4b26f436ffcc4edaace14844f354501a5785/notebooks/sagemaker_automl/config.py to notebooks/sag
```

```
emaker_automl/config.py
download: s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/sagemaker-automl-candidates/a
utoml-dm-16-20-10-05-pr-1-4b26f436ffcc4edaace14844f354501a5785/notebooks/sagemaker_automl/steps.py to notebooks/sage
maker_automl/steps.py
download: s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/sagemaker-automl-candidates/a
utoml-dm-16-20-10-05-pr-1-4b26f436ffcc4edaace14844f354501a5785/notebooks/SageMakerAutopilotDataExplorationNotebook.i
pynb to notebooks/SageMakerAutopilotDataExplorationNotebook.ipynb
```

Review the generated Resources

In [99]:
```
!ls ./generated_module/candidate_data_processors
```

```
dpp0.py  dpp1.py  dpp2.py  sagemaker_serve.py  trainer.py
```

In [100…
```
!ls ./notebooks
```

```
SageMakerAutopilotCandidateDefinitionNotebook.ipynb  sagemaker_automl
SageMakerAutopilotDataExplorationNotebook.ipynb
```

In [101…
```
from IPython.core.display import display, HTML

display(
    HTML(
        '<b>Review <a target="blank" href="https://console.aws.amazon.com/sagemaker/home?region={}#/jobs/">Training J
            region
        )
    )
)
```

Review Training Jobs

In [102…
```
from IPython.core.display import display, HTML

display(
    HTML(
        '<b>Review <a target="blank" href="https://console.aws.amazon.com/sagemaker/home?region={}#/transform-jobs/">
            region
        )
    )
)
```

Review Batch Transform Jobs

The next cell will show InProgress for a few minutes

In [103…

```
%%time

job_description_response = sm.describe_auto_ml_job(AutoMLJobName=auto_ml_job_name)
job_status = job_description_response["AutoMLJobStatus"]
job_sec_status = job_description_response["AutoMLJobSecondaryStatus"]
print(job_status)
print(job_sec_status)
if job_status not in ("Stopped", "Failed"):
    while job_status in ("InProgress") and job_sec_status in ("FeatureEngineering"):
        job_description_response = sm.describe_auto_ml_job(AutoMLJobName=auto_ml_job_name)
        job_status = job_description_response["AutoMLJobStatus"]
        job_sec_status = job_description_response["AutoMLJobSecondaryStatus"]
        print(job_status, job_sec_status)
        sleep(15)
    print("[OK] Feature engineering phase completed.\n")

print(json.dumps(job_description_response, indent=4, sort_keys=True, default=str))
```

```
Completed
Completed
[OK] Feature engineering phase completed.

{
    "AutoMLJobArn": "arn:aws:sagemaker:us-east-1:614093401978:automl-job/automl-dm-16-20-10-05",
    "AutoMLJobArtifacts": {
        "CandidateDefinitionNotebookLocation": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-
20-10-05/sagemaker-automl-candidates/automl-dm-16-20-10-05-pr-1-4b26f436ffcc4edaace14844f354501a5785/notebooks/SageM
akerAutopilotCandidateDefinitionNotebook.ipynb",
        "DataExplorationNotebookLocation": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-1
0-05/sagemaker-automl-candidates/automl-dm-16-20-10-05-pr-1-4b26f436ffcc4edaace14844f354501a5785/notebooks/SageMaker
AutopilotDataExplorationNotebook.ipynb"
    },
    "AutoMLJobConfig": {
        "CompletionCriteria": {
            "MaxAutoMLJobRuntimeInSeconds": 5400,
            "MaxCandidates": 3,
            "MaxRuntimePerTrainingJobInSeconds": 900
        }
    },
    "AutoMLJobName": "automl-dm-16-20-10-05",
    "AutoMLJobSecondaryStatus": "Completed",
    "AutoMLJobStatus": "Completed",
    "BestCandidate": {
        "CandidateName": "automl-dm-16-20-10-054If5RqjsyAN-001-61c635e4",
        "CandidateProperties": {
            "CandidateArtifactLocations": {
                "Explainability": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/docu
mentation/explainability/output",
                "ModelInsights": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/docum
entation/model_monitor/output"
            },
            "CandidateMetrics": [
                {
                    "MetricName": "F1",
                    "Set": "Validation",
                    "StandardMetricName": "F1",
                    "Value": 0.995169997215271
                },
                {
                    "MetricName": "LogLoss",
                    "Set": "Validation",
                    "StandardMetricName": "LogLoss",
```

```
                    "Value": 0.11816000193357468
                },
                {
                    "MetricName": "Recall",
                    "Set": "Validation",
                    "StandardMetricName": "Recall",
                    "Value": 1.0
                },
                {
                    "MetricName": "Precision",
                    "Set": "Validation",
                    "StandardMetricName": "Precision",
                    "Value": 0.9904199838638306
                },
                {
                    "MetricName": "AUC",
                    "Set": "Validation",
                    "StandardMetricName": "AUC",
                    "Value": 0.9990599751472473
                },
                {
                    "MetricName": "Accuracy",
                    "Set": "Validation",
                    "StandardMetricName": "Accuracy",
                    "Value": 0.9953600168228149
                },
                {
                    "MetricName": "BalancedAccuracy",
                    "Set": "Validation",
                    "StandardMetricName": "BalancedAccuracy",
                    "Value": 0.9955800175666809
                }
            ]
        },
        "CandidateStatus": "Completed",
        "CandidateSteps": [
            {
                "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:processing-job/automl-dm-16-20-10-05-d
b-1-2887815310fd4416804d2079485d7d062e1a",
                "CandidateStepName": "automl-dm-16-20-10-05-db-1-2887815310fd4416804d2079485d7d062e1a",
                "CandidateStepType": "AWS::SageMaker::ProcessingJob"
            },
            {
                "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:training-job/automl-dm-16-20-10-05-dpp
```

```
1-1-8411a0fbc81748a9958acf62493120d4d7",
                "CandidateStepName": "automl-dm-16-20-10-05-dpp1-1-8411a0fbc81748a9958acf62493120d4d7",
                "CandidateStepType": "AWS::SageMaker::TrainingJob"
            },
            {
                "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:transform-job/automl-dm-16-20-10-05-dp
p1-csv-1-18357715aec34f7eb4c3b75bad6e49",
                "CandidateStepName": "automl-dm-16-20-10-05-dpp1-csv-1-18357715aec34f7eb4c3b75bad6e49",
                "CandidateStepType": "AWS::SageMaker::TransformJob"
            },
            {
                "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:training-job/automl-dm-16-20-10-054If5
RqjsyAN-001-61c635e4",
                "CandidateStepName": "automl-dm-16-20-10-054If5RqjsyAN-001-61c635e4",
                "CandidateStepType": "AWS::SageMaker::TrainingJob"
            }
        ],
        "CreationTime": "2023-04-16 20:33:00+00:00",
        "EndTime": "2023-04-16 20:38:38+00:00",
        "FinalAutoMLJobObjectiveMetric": {
            "MetricName": "validation:f1_binary",
            "StandardMetricName": "F1",
            "Value": 0.995169997215271
        },
        "InferenceContainers": [
            {
                "Environment": {
                    "AUTOML_TRANSFORM_MODE": "feature-transform",
                    "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "application/x-recordio-protobuf",
                    "SAGEMAKER_PROGRAM": "sagemaker_serve",
                    "SAGEMAKER_SUBMIT_DIRECTORY": "/opt/ml/model/code"
                },
                "Image": "683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-sklearn-automl:2.5-1-cpu-py3",
                "ModelDataUrl": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/data-p
rocessor-models/automl-dm-16-20-10-05-dpp1-1-8411a0fbc81748a9958acf62493120d4d7/output/model.tar.gz"
            },
            {
                "Environment": {
                    "MAX_CONTENT_LENGTH": "20971520",
                    "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "text/csv",
                    "SAGEMAKER_INFERENCE_OUTPUT": "predicted_label",
                    "SAGEMAKER_INFERENCE_SUPPORTED": "predicted_label,probability,probabilities"
                },
                "Image": "683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-xgboost:1.3-1-cpu-py3",
```

```
                "ModelDataUrl": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/tuning
/automl-dm--dpp1-xgb/automl-dm-16-20-10-054If5RqjsyAN-001-61c635e4/output/model.tar.gz"
            },
            {
                "Environment": {
                    "AUTOML_TRANSFORM_MODE": "inverse-label-transform",
                    "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "text/csv",
                    "SAGEMAKER_INFERENCE_INPUT": "predicted_label",
                    "SAGEMAKER_INFERENCE_OUTPUT": "predicted_label",
                    "SAGEMAKER_INFERENCE_SUPPORTED": "predicted_label,probability,labels,probabilities",
                    "SAGEMAKER_PROGRAM": "sagemaker_serve",
                    "SAGEMAKER_SUBMIT_DIRECTORY": "/opt/ml/model/code"
                },
                "Image": "683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-sklearn-automl:2.5-1-cpu-py3",
                "ModelDataUrl": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/data-p
rocessor-models/automl-dm-16-20-10-05-dpp1-1-8411a0fbc81748a9958acf62493120d4d7/output/model.tar.gz"
            }
        ],
        "LastModifiedTime": "2023-04-16 20:39:21.759000+00:00",
        "ObjectiveStatus": "Succeeded"
    },
    "CreationTime": "2023-04-16 20:14:58.074000+00:00",
    "EndTime": "2023-04-16 20:47:31.359000+00:00",
    "GenerateCandidateDefinitionsOnly": false,
    "InputDataConfig": [
        {
            "ChannelType": "training",
            "ContentType": "text/csv;header=present",
            "DataSource": {
                "S3DataSource": {
                    "S3DataType": "S3Prefix",
                    "S3Uri": "s3://sagemaker-us-east-1-614093401978/data/df_autopilot.csv"
                }
            },
            "TargetAttributeName": "AML_detected"
        }
    ],
    "LastModifiedTime": "2023-04-16 20:47:31.403000+00:00",
    "OutputDataConfig": {
        "S3OutputPath": "s3://sagemaker-us-east-1-614093401978/models/autopilot"
    },
    "ResolvedAttributes": {
        "AutoMLJobObjective": {
            "MetricName": "F1"
```

```
        },
        "CompletionCriteria": {
            "MaxAutoMLJobRuntimeInSeconds": 5400,
            "MaxCandidates": 3,
            "MaxRuntimePerTrainingJobInSeconds": 900
        },
        "ProblemType": "BinaryClassification"
    },
    "ResponseMetadata": {
        "HTTPHeaders": {
            "content-length": "5897",
            "content-type": "application/x-amz-json-1.1",
            "date": "Mon, 17 Apr 2023 03:55:49 GMT",
            "x-amzn-requestid": "ce5a80c9-5e76-4b3c-85ac-430065d52082"
        },
        "HTTPStatusCode": 200,
        "RequestId": "ce5a80c9-5e76-4b3c-85ac-430065d52082",
        "RetryAttempts": 0
    },
    "RoleArn": "arn:aws:iam::614093401978:role/LabRole"
}
CPU times: user 6.96 ms, sys: 0 ns, total: 6.96 ms
Wall time: 105 ms
```

Model Training and Tuning

In [104…

```
from IPython.core.display import display, HTML

display(
    HTML(
        '<b>Review <a target="blank" href="https://console.aws.amazon.com/sagemaker/home?region={}#/hyper-tuning-jobs
            region
        )
    )
)
```

Review Hyperparameter Tuning Jobs

In [105…

```
from IPython.core.display import display, HTML

display(
    HTML(
        '<b>Review <a target="blank" href="https://console.aws.amazon.com/sagemaker/home?region={}#/jobs/">Training J
            region
        )
    )
)
```

Review Training Jobs

The next cell will show InProgress for a few minutes

In [106…

```
%%time

job_description_response = sm.describe_auto_ml_job(AutoMLJobName=auto_ml_job_name)
job_status = job_description_response["AutoMLJobStatus"]
job_sec_status = job_description_response["AutoMLJobSecondaryStatus"]
print(job_status)
print(job_sec_status)
if job_status not in ("Stopped", "Failed"):
    while job_status in ("InProgress") and job_sec_status in ("ModelTuning"):
        job_description_response = sm.describe_auto_ml_job(AutoMLJobName=auto_ml_job_name)
        job_status = job_description_response["AutoMLJobStatus"]
        job_sec_status = job_description_response["AutoMLJobSecondaryStatus"]
        print(job_status, job_sec_status)
        sleep(15)
    print("[OK] Model tuning phase completed.\n")

print(json.dumps(job_description_response, indent=4, sort_keys=True, default=str))
```

```
Completed
Completed
[OK] Model tuning phase completed.

{
    "AutoMLJobArn": "arn:aws:sagemaker:us-east-1:614093401978:automl-job/automl-dm-16-20-10-05",
    "AutoMLJobArtifacts": {
        "CandidateDefinitionNotebookLocation": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-
20-10-05/sagemaker-automl-candidates/automl-dm-16-20-10-05-pr-1-4b26f436ffcc4edaace14844f354501a5785/notebooks/SageM
akerAutopilotCandidateDefinitionNotebook.ipynb",
        "DataExplorationNotebookLocation": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-1
0-05/sagemaker-automl-candidates/automl-dm-16-20-10-05-pr-1-4b26f436ffcc4edaace14844f354501a5785/notebooks/SageMaker
AutopilotDataExplorationNotebook.ipynb"
    },
    "AutoMLJobConfig": {
        "CompletionCriteria": {
            "MaxAutoMLJobRuntimeInSeconds": 5400,
            "MaxCandidates": 3,
            "MaxRuntimePerTrainingJobInSeconds": 900
        }
    },
    "AutoMLJobName": "automl-dm-16-20-10-05",
    "AutoMLJobSecondaryStatus": "Completed",
    "AutoMLJobStatus": "Completed",
    "BestCandidate": {
        "CandidateName": "automl-dm-16-20-10-054If5RqjsyAN-001-61c635e4",
        "CandidateProperties": {
            "CandidateArtifactLocations": {
                "Explainability": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/docu
mentation/explainability/output",
                "ModelInsights": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/docum
entation/model_monitor/output"
            },
            "CandidateMetrics": [
                {
                    "MetricName": "F1",
                    "Set": "Validation",
                    "StandardMetricName": "F1",
                    "Value": 0.995169997215271
                },
                {
                    "MetricName": "LogLoss",
                    "Set": "Validation",
                    "StandardMetricName": "LogLoss",
```

```
                    "Value": 0.11816000193357468
                },
                {
                    "MetricName": "Recall",
                    "Set": "Validation",
                    "StandardMetricName": "Recall",
                    "Value": 1.0
                },
                {
                    "MetricName": "Precision",
                    "Set": "Validation",
                    "StandardMetricName": "Precision",
                    "Value": 0.9904199838638306
                },
                {
                    "MetricName": "AUC",
                    "Set": "Validation",
                    "StandardMetricName": "AUC",
                    "Value": 0.9990599751472473
                },
                {
                    "MetricName": "Accuracy",
                    "Set": "Validation",
                    "StandardMetricName": "Accuracy",
                    "Value": 0.9953600168228149
                },
                {
                    "MetricName": "BalancedAccuracy",
                    "Set": "Validation",
                    "StandardMetricName": "BalancedAccuracy",
                    "Value": 0.9955800175666809
                }
            ]
        },
        "CandidateStatus": "Completed",
        "CandidateSteps": [
            {
                "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:processing-job/automl-dm-16-20-10-05-d
b-1-2887815310fd4416804d2079485d7d062e1a",
                "CandidateStepName": "automl-dm-16-20-10-05-db-1-2887815310fd4416804d2079485d7d062e1a",
                "CandidateStepType": "AWS::SageMaker::ProcessingJob"
            },
            {
                "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:training-job/automl-dm-16-20-10-05-dpp
```

```
1-1-8411a0fbc81748a9958acf62493120d4d7",
                "CandidateStepName": "automl-dm-16-20-10-05-dpp1-1-8411a0fbc81748a9958acf62493120d4d7",
                "CandidateStepType": "AWS::SageMaker::TrainingJob"
            },
            {
                "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:transform-job/automl-dm-16-20-10-05-dp
p1-csv-1-18357715aec34f7eb4c3b75bad6e49",
                "CandidateStepName": "automl-dm-16-20-10-05-dpp1-csv-1-18357715aec34f7eb4c3b75bad6e49",
                "CandidateStepType": "AWS::SageMaker::TransformJob"
            },
            {
                "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:training-job/automl-dm-16-20-10-054If5
RqjsyAN-001-61c635e4",
                "CandidateStepName": "automl-dm-16-20-10-054If5RqjsyAN-001-61c635e4",
                "CandidateStepType": "AWS::SageMaker::TrainingJob"
            }
        ],
        "CreationTime": "2023-04-16 20:33:00+00:00",
        "EndTime": "2023-04-16 20:38:38+00:00",
        "FinalAutoMLJobObjectiveMetric": {
            "MetricName": "validation:f1_binary",
            "StandardMetricName": "F1",
            "Value": 0.995169997215271
        },
        "InferenceContainers": [
            {
                "Environment": {
                    "AUTOML_TRANSFORM_MODE": "feature-transform",
                    "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "application/x-recordio-protobuf",
                    "SAGEMAKER_PROGRAM": "sagemaker_serve",
                    "SAGEMAKER_SUBMIT_DIRECTORY": "/opt/ml/model/code"
                },
                "Image": "683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-sklearn-automl:2.5-1-cpu-py3",
                "ModelDataUrl": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/data-p
rocessor-models/automl-dm-16-20-10-05-dpp1-1-8411a0fbc81748a9958acf62493120d4d7/output/model.tar.gz"
            },
            {
                "Environment": {
                    "MAX_CONTENT_LENGTH": "20971520",
                    "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "text/csv",
                    "SAGEMAKER_INFERENCE_OUTPUT": "predicted_label",
                    "SAGEMAKER_INFERENCE_SUPPORTED": "predicted_label,probability,probabilities"
                },
                "Image": "683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-xgboost:1.3-1-cpu-py3",
```

```
                "ModelDataUrl": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/tuning
/automl-dm--dpp1-xgb/automl-dm-16-20-10-054If5RqjsyAN-001-61c635e4/output/model.tar.gz"
            },
            {
                "Environment": {
                    "AUTOML_TRANSFORM_MODE": "inverse-label-transform",
                    "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "text/csv",
                    "SAGEMAKER_INFERENCE_INPUT": "predicted_label",
                    "SAGEMAKER_INFERENCE_OUTPUT": "predicted_label",
                    "SAGEMAKER_INFERENCE_SUPPORTED": "predicted_label,probability,labels,probabilities",
                    "SAGEMAKER_PROGRAM": "sagemaker_serve",
                    "SAGEMAKER_SUBMIT_DIRECTORY": "/opt/ml/model/code"
                },
                "Image": "683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-sklearn-automl:2.5-1-cpu-py3",
                "ModelDataUrl": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/data-p
rocessor-models/automl-dm-16-20-10-05-dpp1-1-8411a0fbc81748a9958acf62493120d4d7/output/model.tar.gz"
            }
        ],
        "LastModifiedTime": "2023-04-16 20:39:21.759000+00:00",
        "ObjectiveStatus": "Succeeded"
    },
    "CreationTime": "2023-04-16 20:14:58.074000+00:00",
    "EndTime": "2023-04-16 20:47:31.359000+00:00",
    "GenerateCandidateDefinitionsOnly": false,
    "InputDataConfig": [
        {
            "ChannelType": "training",
            "ContentType": "text/csv;header=present",
            "DataSource": {
                "S3DataSource": {
                    "S3DataType": "S3Prefix",
                    "S3Uri": "s3://sagemaker-us-east-1-614093401978/data/df_autopilot.csv"
                }
            },
            "TargetAttributeName": "AML_detected"
        }
    ],
    "LastModifiedTime": "2023-04-16 20:47:31.403000+00:00",
    "OutputDataConfig": {
        "S3OutputPath": "s3://sagemaker-us-east-1-614093401978/models/autopilot"
    },
    "ResolvedAttributes": {
        "AutoMLJobObjective": {
            "MetricName": "F1"
```

```
        },
        "CompletionCriteria": {
            "MaxAutoMLJobRuntimeInSeconds": 5400,
            "MaxCandidates": 3,
            "MaxRuntimePerTrainingJobInSeconds": 900
        },
        "ProblemType": "BinaryClassification"
    },
    "ResponseMetadata": {
        "HTTPHeaders": {
            "content-length": "5897",
            "content-type": "application/x-amz-json-1.1",
            "date": "Mon, 17 Apr 2023 03:55:49 GMT",
            "x-amzn-requestid": "83313c47-458e-44dc-8f62-7c4d38216386"
        },
        "HTTPStatusCode": 200,
        "RequestId": "83313c47-458e-44dc-8f62-7c4d38216386",
        "RetryAttempts": 0
    },
    "RoleArn": "arn:aws:iam::614093401978:role/LabRole"
}
CPU times: user 4.86 ms, sys: 96 µs, total: 4.96 ms
Wall time: 112 ms
```


In [107…

```
%%time

job_description_response = sm.describe_auto_ml_job(AutoMLJobName=auto_ml_job_name)
job_status = job_description_response["AutoMLJobStatus"]
print(job_status)
if job_status not in ("Stopped", "Failed"):
    while job_status not in ("Completed"):
        job_description_response = sm.describe_auto_ml_job(AutoMLJobName=auto_ml_job_name)
        job_status = job_description_response["AutoMLJobStatus"]
        print(job_status)
        sleep(10)
    print("[OK] Autopilot Job completed.\n")
else:
    print(job_status)
```

```
Completed
[OK] Autopilot Job completed.

CPU times: user 4.51 ms, sys: 516 µs, total: 5.03 ms
Wall time: 102 ms
```

Viewing all Candidates

In [108…

```
candidates_response = sm.list_candidates_for_auto_ml_job(
    AutoMLJobName=auto_ml_job_name, SortBy="FinalObjectiveMetricValue"
)
```

In [109…

```
while "Candidates" not in candidates_response.keys():
    candidates_response = sm.list_candidates_for_auto_ml_job(
        AutoMLJobName=auto_ml_job_name, SortBy="FinalObjectiveMetricValue"
    )
    print("[INFO] Autopilot Job is generating the Candidates. Please wait.")
    print(json.dumps(candidates_response, indent=4, sort_keys=True, default=str))
    sleep(10)

candidates = candidates_response["Candidates"]
print("[OK] Candidates generated.")
```

```
[OK] Candidates generated.
```

In [110…

```
print(candidates[0].keys())
```

```
dict_keys(['CandidateName', 'FinalAutoMLJobObjectiveMetric', 'ObjectiveStatus', 'CandidateSteps', 'CandidateStatus',
'InferenceContainers', 'CreationTime', 'EndTime', 'LastModifiedTime', 'CandidateProperties'])
```

In [111…

```
while "CandidateName" not in candidates[0]:
    candidates_response = sm.list_candidates_for_auto_ml_job(
        AutoMLJobName=auto_ml_job_name, SortBy="FinalObjectiveMetricValue"
    )
    candidates = candidates_response["Candidates"]
    print("[INFO] Autopilot Job is generating CandidateName. Please wait. ")
    print(json.dumps(candidates, indent=4, sort_keys=True, default=str))
    sleep(10)

print("[OK] CandidateName generated.")
```

```
[OK] CandidateName generated.
```

In [112…

```
while "FinalAutoMLJobObjectiveMetric" not in candidates[0]:
    candidates_response = sm.list_candidates_for_auto_ml_job(
        AutoMLJobName=auto_ml_job_name, SortBy="FinalObjectiveMetricValue"
    )
    candidates = candidates_response["Candidates"]
    print("[INFO] Autopilot Job is generating FinalAutoMLJobObjectiveMetric. Please wait. ")
    print(json.dumps(candidates, indent=4, sort_keys=True, default=str))
    sleep(10)

print("[OK] FinalAutoMLJobObjectiveMetric generated.")
```

```
[OK] FinalAutoMLJobObjectiveMetric generated.
```

In [113…

```
print(json.dumps(candidates, indent=4, sort_keys=True, default=str))
```

```
[
    {
        "CandidateName": "automl-dm-16-20-10-054If5RqjsyAN-001-61c635e4",
        "CandidateProperties": {
            "CandidateArtifactLocations": {
                "Explainability": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/docu
mentation/explainability/output",
                "ModelInsights": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/docum
entation/model_monitor/output"
            },
            "CandidateMetrics": [
                {
                    "MetricName": "F1",
                    "Set": "Validation",
                    "StandardMetricName": "F1",
                    "Value": 0.995169997215271
                },
                {
                    "MetricName": "LogLoss",
                    "Set": "Validation",
                    "StandardMetricName": "LogLoss",
                    "Value": 0.11816000193357468
                },
                {
                    "MetricName": "Recall",
                    "Set": "Validation",
                    "StandardMetricName": "Recall",
                    "Value": 1.0
                },
                {
                    "MetricName": "Precision",
                    "Set": "Validation",
                    "StandardMetricName": "Precision",
                    "Value": 0.9904199838638306
                },
                {
                    "MetricName": "AUC",
                    "Set": "Validation",
                    "StandardMetricName": "AUC",
                    "Value": 0.9990599751472473
                },
                {
                    "MetricName": "Accuracy",
                    "Set": "Validation",
```

```
                    "StandardMetricName": "Accuracy",
                    "Value": 0.9953600168228149
                },
                {
                    "MetricName": "BalancedAccuracy",
                    "Set": "Validation",
                    "StandardMetricName": "BalancedAccuracy",
                    "Value": 0.9955800175666809
                }
            ]
        },
        "CandidateStatus": "Completed",
        "CandidateSteps": [
            {
                "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:processing-job/automl-dm-16-20-10-05-d
b-1-2887815310fd4416804d2079485d7d062e1a",
                "CandidateStepName": "automl-dm-16-20-10-05-db-1-2887815310fd4416804d2079485d7d062e1a",
                "CandidateStepType": "AWS::SageMaker::ProcessingJob"
            },
            {
                "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:training-job/automl-dm-16-20-10-05-dpp
1-1-8411a0fbc81748a9958acf62493120d4d7",
                "CandidateStepName": "automl-dm-16-20-10-05-dpp1-1-8411a0fbc81748a9958acf62493120d4d7",
                "CandidateStepType": "AWS::SageMaker::TrainingJob"
            },
            {
                "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:transform-job/automl-dm-16-20-10-05-dp
p1-csv-1-18357715aec34f7eb4c3b75bad6e49",
                "CandidateStepName": "automl-dm-16-20-10-05-dpp1-csv-1-18357715aec34f7eb4c3b75bad6e49",
                "CandidateStepType": "AWS::SageMaker::TransformJob"
            },
            {
                "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:training-job/automl-dm-16-20-10-054If5
RqjsyAN-001-61c635e4",
                "CandidateStepName": "automl-dm-16-20-10-054If5RqjsyAN-001-61c635e4",
                "CandidateStepType": "AWS::SageMaker::TrainingJob"
            }
        ],
        "CreationTime": "2023-04-16 20:33:00+00:00",
        "EndTime": "2023-04-16 20:38:38+00:00",
        "FinalAutoMLJobObjectiveMetric": {
            "MetricName": "validation:f1_binary",
            "StandardMetricName": "F1",
            "Value": 0.995169997215271
```

```
        },
        "InferenceContainers": [
            {
                "Environment": {
                    "AUTOML_TRANSFORM_MODE": "feature-transform",
                    "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "application/x-recordio-protobuf",
                    "SAGEMAKER_PROGRAM": "sagemaker_serve",
                    "SAGEMAKER_SUBMIT_DIRECTORY": "/opt/ml/model/code"
                },
                "Image": "683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-sklearn-automl:2.5-1-cpu-py3",
                "ModelDataUrl": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/data-p
rocessor-models/automl-dm-16-20-10-05-dpp1-1-8411a0fbc81748a9958acf62493120d4d7/output/model.tar.gz"
            },
            {
                "Environment": {
                    "MAX_CONTENT_LENGTH": "20971520",
                    "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "text/csv",
                    "SAGEMAKER_INFERENCE_OUTPUT": "predicted_label",
                    "SAGEMAKER_INFERENCE_SUPPORTED": "predicted_label,probability,probabilities"
                },
                "Image": "683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-xgboost:1.3-1-cpu-py3",
                "ModelDataUrl": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/tuning
/automl-dm--dpp1-xgb/automl-dm-16-20-10-054If5RqjsyAN-001-61c635e4/output/model.tar.gz"
            },
            {
                "Environment": {
                    "AUTOML_TRANSFORM_MODE": "inverse-label-transform",
                    "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "text/csv",
                    "SAGEMAKER_INFERENCE_INPUT": "predicted_label",
                    "SAGEMAKER_INFERENCE_OUTPUT": "predicted_label",
                    "SAGEMAKER_INFERENCE_SUPPORTED": "predicted_label,probability,labels,probabilities",
                    "SAGEMAKER_PROGRAM": "sagemaker_serve",
                    "SAGEMAKER_SUBMIT_DIRECTORY": "/opt/ml/model/code"
                },
                "Image": "683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-sklearn-automl:2.5-1-cpu-py3",
                "ModelDataUrl": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/data-p
rocessor-models/automl-dm-16-20-10-05-dpp1-1-8411a0fbc81748a9958acf62493120d4d7/output/model.tar.gz"
            }
        ],
        "LastModifiedTime": "2023-04-16 20:39:21.759000+00:00",
        "ObjectiveStatus": "Succeeded"
    },
    {
        "CandidateName": "automl-dm-16-20-10-054If5RqjsyAN-003-910bc585",
```

```
"CandidateProperties": {
    "CandidateMetrics": [
        {
            "MetricName": "F1",
            "Set": "Validation",
            "StandardMetricName": "F1",
            "Value": 0.9795699715614319
        },
        {
            "MetricName": "LogLoss",
            "Set": "Validation",
            "StandardMetricName": "LogLoss",
            "Value": 0.11089000105857849
        },
        {
            "MetricName": "Recall",
            "Set": "Validation",
            "StandardMetricName": "Recall",
            "Value": 0.9756100177764893
        },
        {
            "MetricName": "Precision",
            "Set": "Validation",
            "StandardMetricName": "Precision",
            "Value": 0.9837700128555298
        },
        {
            "MetricName": "AUC",
            "Set": "Validation",
            "StandardMetricName": "AUC",
            "Value": 0.998420000076294
        },
        {
            "MetricName": "Accuracy",
            "Set": "Validation",
            "StandardMetricName": "Accuracy",
            "Value": 0.9806600213050842
        },
        {
            "MetricName": "BalancedAccuracy",
            "Set": "Validation",
            "StandardMetricName": "BalancedAccuracy",
            "Value": 0.9804199934005737
        }
```

```
            ]
        },
        "CandidateStatus": "Completed",
        "CandidateSteps": [
            {
                "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:processing-job/automl-dm-16-20-10-05-d
b-1-2887815310fd4416804d2079485d7d062e1a",
                "CandidateStepName": "automl-dm-16-20-10-05-db-1-2887815310fd4416804d2079485d7d062e1a",
                "CandidateStepType": "AWS::SageMaker::ProcessingJob"
            },
            {
                "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:training-job/automl-dm-16-20-10-05-dpp
1-1-8411a0fbc81748a9958acf62493120d4d7",
                "CandidateStepName": "automl-dm-16-20-10-05-dpp1-1-8411a0fbc81748a9958acf62493120d4d7",
                "CandidateStepType": "AWS::SageMaker::TrainingJob"
            },
            {
                "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:transform-job/automl-dm-16-20-10-05-dp
p1-csv-1-18357715aec34f7eb4c3b75bad6e49",
                "CandidateStepName": "automl-dm-16-20-10-05-dpp1-csv-1-18357715aec34f7eb4c3b75bad6e49",
                "CandidateStepType": "AWS::SageMaker::TransformJob"
            },
            {
                "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:training-job/automl-dm-16-20-10-054If5
RqjsyAN-003-910bc585",
                "CandidateStepName": "automl-dm-16-20-10-054If5RqjsyAN-003-910bc585",
                "CandidateStepType": "AWS::SageMaker::TrainingJob"
            }
        ],
        "CreationTime": "2023-04-16 20:33:21+00:00",
        "EndTime": "2023-04-16 20:36:31+00:00",
        "FinalAutoMLJobObjectiveMetric": {
            "MetricName": "validation:f1_binary",
            "StandardMetricName": "F1",
            "Value": 0.9795699715614319
        },
        "InferenceContainers": [
            {
                "Environment": {
                    "AUTOML_TRANSFORM_MODE": "feature-transform",
                    "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "application/x-recordio-protobuf",
                    "SAGEMAKER_PROGRAM": "sagemaker_serve",
                    "SAGEMAKER_SUBMIT_DIRECTORY": "/opt/ml/model/code"
                },
```

```
                "Image": "683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-sklearn-automl:2.5-1-cpu-py3",
                "ModelDataUrl": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/data-p
rocessor-models/automl-dm-16-20-10-05-dpp1-1-8411a0fbc81748a9958acf62493120d4d7/output/model.tar.gz"
            },
            {
                "Environment": {
                    "MAX_CONTENT_LENGTH": "20971520",
                    "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "text/csv",
                    "SAGEMAKER_INFERENCE_OUTPUT": "predicted_label",
                    "SAGEMAKER_INFERENCE_SUPPORTED": "predicted_label,probability,probabilities"
                },
                "Image": "683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-xgboost:1.3-1-cpu-py3",
                "ModelDataUrl": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/tuning
/automl-dm--dpp1-xgb/automl-dm-16-20-10-054If5RqjsyAN-003-910bc585/output/model.tar.gz"
            },
            {
                "Environment": {
                    "AUTOML_TRANSFORM_MODE": "inverse-label-transform",
                    "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "text/csv",
                    "SAGEMAKER_INFERENCE_INPUT": "predicted_label",
                    "SAGEMAKER_INFERENCE_OUTPUT": "predicted_label",
                    "SAGEMAKER_INFERENCE_SUPPORTED": "predicted_label,probability,labels,probabilities",
                    "SAGEMAKER_PROGRAM": "sagemaker_serve",
                    "SAGEMAKER_SUBMIT_DIRECTORY": "/opt/ml/model/code"
                },
                "Image": "683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-sklearn-automl:2.5-1-cpu-py3",
                "ModelDataUrl": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/data-p
rocessor-models/automl-dm-16-20-10-05-dpp1-1-8411a0fbc81748a9958acf62493120d4d7/output/model.tar.gz"
            }
        ],
        "LastModifiedTime": "2023-04-16 20:39:21.685000+00:00",
        "ObjectiveStatus": "Succeeded"
    },
    {
        "CandidateName": "automl-dm-16-20-10-054If5RqjsyAN-002-ebe18ef1",
        "CandidateProperties": {
            "CandidateMetrics": [
                {
                    "MetricName": "F1",
                    "Set": "Validation",
                    "StandardMetricName": "F1",
                    "Value": 0.9746400117874146
                },
                {
```

```
                    "MetricName": "LogLoss",
                    "Set": "Validation",
                    "StandardMetricName": "LogLoss",
                    "Value": 0.1131799966096878
                },
                {
                    "MetricName": "Recall",
                    "Set": "Validation",
                    "StandardMetricName": "Recall",
                    "Value": 0.9666699767112732
                },
                {
                    "MetricName": "Precision",
                    "Set": "Validation",
                    "StandardMetricName": "Precision",
                    "Value": 0.9835600256919861
                },
                {
                    "MetricName": "AUC",
                    "Set": "Validation",
                    "StandardMetricName": "AUC",
                    "Value": 0.9985899925231934
                },
                {
                    "MetricName": "Accuracy",
                    "Set": "Validation",
                    "StandardMetricName": "Accuracy",
                    "Value": 0.9764099717140198
                },
                {
                    "MetricName": "BalancedAccuracy",
                    "Set": "Validation",
                    "StandardMetricName": "BalancedAccuracy",
                    "Value": 0.9759500026702881
                }
            ]
        },
        "CandidateStatus": "Completed",
        "CandidateSteps": [
            {
                "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:processing-job/automl-dm-16-20-10-05-d
b-1-2887815310fd4416804d2079485d7d062e1a",
                "CandidateStepName": "automl-dm-16-20-10-05-db-1-2887815310fd4416804d2079485d7d062e1a",
                "CandidateStepType": "AWS::SageMaker::ProcessingJob"
```

```
            },
            {
                "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:training-job/automl-dm-16-20-10-05-dpp
0-1-cbc8bbf88b664a35a963ac2c65fe94899c",
                "CandidateStepName": "automl-dm-16-20-10-05-dpp0-1-cbc8bbf88b664a35a963ac2c65fe94899c",
                "CandidateStepType": "AWS::SageMaker::TrainingJob"
            },
            {
                "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:transform-job/automl-dm-16-20-10-05-dp
p0-csv-1-0d6756d0637d421dabef7d3c55af5a",
                "CandidateStepName": "automl-dm-16-20-10-05-dpp0-csv-1-0d6756d0637d421dabef7d3c55af5a",
                "CandidateStepType": "AWS::SageMaker::TransformJob"
            },
            {
                "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:training-job/automl-dm-16-20-10-054If5
RqjsyAN-002-ebe18ef1",
                "CandidateStepName": "automl-dm-16-20-10-054If5RqjsyAN-002-ebe18ef1",
                "CandidateStepType": "AWS::SageMaker::TrainingJob"
            }
        ],
        "CreationTime": "2023-04-16 20:33:01+00:00",
        "EndTime": "2023-04-16 20:35:19+00:00",
        "FinalAutoMLJobObjectiveMetric": {
            "MetricName": "validation:f1_binary",
            "StandardMetricName": "F1",
            "Value": 0.9746400117874146
        },
        "InferenceContainers": [
            {
                "Environment": {
                    "AUTOML_TRANSFORM_MODE": "feature-transform",
                    "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "application/x-recordio-protobuf",
                    "SAGEMAKER_PROGRAM": "sagemaker_serve",
                    "SAGEMAKER_SUBMIT_DIRECTORY": "/opt/ml/model/code"
                },
                "Image": "683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-sklearn-automl:2.5-1-cpu-py3",
                "ModelDataUrl": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/data-p
rocessor-models/automl-dm-16-20-10-05-dpp0-1-cbc8bbf88b664a35a963ac2c65fe94899c/output/model.tar.gz"
            },
            {
                "Environment": {
                    "MAX_CONTENT_LENGTH": "20971520",
                    "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "text/csv",
                    "SAGEMAKER_INFERENCE_OUTPUT": "predicted_label",
```

```
                    "SAGEMAKER_INFERENCE_SUPPORTED": "predicted_label,probability,probabilities"
                },
                "Image": "683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-xgboost:1.3-1-cpu-py3",
                "ModelDataUrl": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/tuning
/automl-dm--dpp0-xgb/automl-dm-16-20-10-054If5RqjsyAN-002-ebe18ef1/output/model.tar.gz"
            },
            {
                "Environment": {
                    "AUTOML_TRANSFORM_MODE": "inverse-label-transform",
                    "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "text/csv",
                    "SAGEMAKER_INFERENCE_INPUT": "predicted_label",
                    "SAGEMAKER_INFERENCE_OUTPUT": "predicted_label",
                    "SAGEMAKER_INFERENCE_SUPPORTED": "predicted_label,probability,labels,probabilities",
                    "SAGEMAKER_PROGRAM": "sagemaker_serve",
                    "SAGEMAKER_SUBMIT_DIRECTORY": "/opt/ml/model/code"
                },
                "Image": "683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-sklearn-automl:2.5-1-cpu-py3",
                "ModelDataUrl": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/data-p
rocessor-models/automl-dm-16-20-10-05-dpp0-1-cbc8bbf88b664a35a963ac2c65fe94899c/output/model.tar.gz"
            }
        ],
        "LastModifiedTime": "2023-04-16 20:39:21.686000+00:00",
        "ObjectiveStatus": "Succeeded"
    }
]
```


In [114…

```
for index, candidate in enumerate(candidates):
    print(
        str(index)
        + "  "
        + candidate["CandidateName"]
        + "  "
        + str(candidate["FinalAutoMLJobObjectiveMetric"]["Value"])
    )
```

```
0  automl-dm-16-20-10-054If5RqjsyAN-001-61c635e4  0.995169997215271
1  automl-dm-16-20-10-054If5RqjsyAN-003-910bc585  0.9795699715614319
2  automl-dm-16-20-10-054If5RqjsyAN-002-ebe18ef1  0.9746400117874146
```

Inspect Trials using Experiments API

In [115…

```
from sagemaker.analytics import ExperimentAnalytics, TrainingJobAnalytics

exp = ExperimentAnalytics(
    sagemaker_session=sess,
    experiment_name=auto_ml_job_name + "-aws-auto-ml-job",
)

df = exp.dataframe()
print(df)
```

```
                                TrialComponentName  \
0  automl-dm-16-20-10-054If5RqjsyAN-001-61c635e4-...
1  automl-dm-16-20-10-054If5RqjsyAN-003-910bc585-...
2  automl-dm-16-20-10-054If5RqjsyAN-002-ebe18ef1-...
3  automl-dm-16-20-10-05-dpp1-csv-1-18357715aec34...
4  automl-dm-16-20-10-05-dpp0-csv-1-0d6756d0637d4...
5  automl-dm-16-20-10-05-dpp1-1-8411a0fbc81748a99...
6  automl-dm-16-20-10-05-dpp0-1-cbc8bbf88b664a35a...
7  automl-dm-16-20-10-05-db-1-2887815310fd4416804...

                                     DisplayName  \
0  automl-dm-16-20-10-054If5RqjsyAN-001-61c635e4-...
1  automl-dm-16-20-10-054If5RqjsyAN-003-910bc585-...
2  automl-dm-16-20-10-054If5RqjsyAN-002-ebe18ef1-...
3  automl-dm-16-20-10-05-dpp1-csv-1-18357715aec34...
4  automl-dm-16-20-10-05-dpp0-csv-1-0d6756d0637d4...
5  automl-dm-16-20-10-05-dpp1-1-8411a0fbc81748a99...
6  automl-dm-16-20-10-05-dpp0-1-cbc8bbf88b664a35a...
7  automl-dm-16-20-10-05-db-1-2887815310fd4416804...

                                       SourceArn  \
0  arn:aws:sagemaker:us-east-1:614093401978:train...
1  arn:aws:sagemaker:us-east-1:614093401978:train...
2  arn:aws:sagemaker:us-east-1:614093401978:train...
3  arn:aws:sagemaker:us-east-1:614093401978:trans...
4  arn:aws:sagemaker:us-east-1:614093401978:trans...
5  arn:aws:sagemaker:us-east-1:614093401978:train...
6  arn:aws:sagemaker:us-east-1:614093401978:train...
7  arn:aws:sagemaker:us-east-1:614093401978:proce...

                                SageMaker.ImageUri  SageMaker.InstanceCount  \
0  683313688378.dkr.ecr.us-east-1.amazonaws.com/s...                      1.0
1  683313688378.dkr.ecr.us-east-1.amazonaws.com/s...                      1.0
2  683313688378.dkr.ecr.us-east-1.amazonaws.com/s...                      1.0
3                                                NaN                      1.0
4                                                NaN                      1.0
5  683313688378.dkr.ecr.us-east-1.amazonaws.com/s...                      1.0
6  683313688378.dkr.ecr.us-east-1.amazonaws.com/s...                      1.0
7                                                NaN                      1.0

  SageMaker.InstanceType  SageMaker.VolumeSizeInGB  _kfold  _num_cv_round  \
0        ml.m5.12xlarge                      50.0     5.0            3.0
1        ml.m5.12xlarge                      50.0     5.0            3.0
2        ml.m5.12xlarge                      50.0     5.0            3.0
```

```
3          ml.m5.4xlarge                       NaN     NaN            NaN
4          ml.m5.4xlarge                       NaN     NaN            NaN
5         ml.m5.12xlarge                      50.0     NaN            NaN
6         ml.m5.12xlarge                      50.0     NaN            NaN
7          ml.m5.2xlarge                     250.0     NaN            NaN

  _tuning_objective_metric  ...  enable_validation_split  input_channel_mode  \
0    validation:f1_binary  ...                      NaN                 NaN
1    validation:f1_binary  ...                      NaN                 NaN
2    validation:f1_binary  ...                      NaN                 NaN
3                     NaN  ...                      NaN                 NaN
4                     NaN  ...                      NaN                 NaN
5                     NaN  ...                      NaN                 NaN
6                     NaN  ...                      NaN                 NaN
7                     NaN  ...                     true                Pipe

               job_name     label_col  max_dataset_size  \
0                   NaN           NaN               NaN
1                   NaN           NaN               NaN
2                   NaN           NaN               NaN
3                   NaN           NaN               NaN
4                   NaN           NaN               NaN
5                   NaN           NaN               NaN
6                   NaN           NaN               NaN
7  automl-dm-16-20-10-05  AML_detected               100

   max_subsampled_dataset_size  SageMaker.ImageUri - MediaType  \
0                          NaN                             NaN
1                          NaN                             NaN
2                          NaN                             NaN
3                          NaN                             NaN
4                          NaN                             NaN
5                          NaN                             NaN
6                          NaN                             NaN
7                            5                             NaN

                        SageMaker.ImageUri - Value  ds - MediaType  \
0                                              NaN             NaN
1                                              NaN             NaN
2                                              NaN             NaN
3                                              NaN             NaN
4                                              NaN             NaN
5                                              NaN             NaN
6                                              NaN             NaN
```

```
7  120479346908.dkr.ecr.us-east-1.amazonaws.com/d...            NaN

                                          ds - Value
0                                                NaN
1                                                NaN
2                                                NaN
3                                                NaN
4                                                NaN
5                                                NaN
6                                                NaN
7  s3://sagemaker-us-east-1-614093401978/models/a...

[8 rows x 126 columns]
```

Explore the Best Candidate

In [116…
```
best_candidate_response = sm.describe_auto_ml_job(AutoMLJobName=auto_ml_job_name)
```

In [117…
```
print(best_candidate_response.keys())
```

```
dict_keys(['AutoMLJobName', 'AutoMLJobArn', 'InputDataConfig', 'OutputDataConfig', 'RoleArn', 'AutoMLJobConfig', 'Cr
eationTime', 'EndTime', 'LastModifiedTime', 'BestCandidate', 'AutoMLJobStatus', 'AutoMLJobSecondaryStatus', 'Generat
eCandidateDefinitionsOnly', 'AutoMLJobArtifacts', 'ResolvedAttributes', 'ResponseMetadata'])
```

In [118…
```
while "BestCandidate" not in best_candidate_response:
    best_candidate_response = sm.describe_auto_ml_job(AutoMLJobName=auto_ml_job_name)
    print("[INFO] Autopilot Job is generating BestCandidate. Please wait. ")
    print(json.dumps(best_candidate_response, indent=4, sort_keys=True, default=str))
    sleep(10)

best_candidate = best_candidate_response["BestCandidate"]
print("[OK] BestCandidate generated.")
```

```
[OK] BestCandidate generated.
```

In [119…
```
print(json.dumps(best_candidate_response, indent=4, sort_keys=True, default=str))
```

```
{
    "AutoMLJobArn": "arn:aws:sagemaker:us-east-1:614093401978:automl-job/automl-dm-16-20-10-05",
    "AutoMLJobArtifacts": {
        "CandidateDefinitionNotebookLocation": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-
20-10-05/sagemaker-automl-candidates/automl-dm-16-20-10-05-pr-1-4b26f436ffcc4edaace14844f354501a5785/notebooks/SageM
akerAutopilotCandidateDefinitionNotebook.ipynb",
        "DataExplorationNotebookLocation": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-1
0-05/sagemaker-automl-candidates/automl-dm-16-20-10-05-pr-1-4b26f436ffcc4edaace14844f354501a5785/notebooks/SageMaker
AutopilotDataExplorationNotebook.ipynb"
    },
    "AutoMLJobConfig": {
        "CompletionCriteria": {
            "MaxAutoMLJobRuntimeInSeconds": 5400,
            "MaxCandidates": 3,
            "MaxRuntimePerTrainingJobInSeconds": 900
        }
    },
    "AutoMLJobName": "automl-dm-16-20-10-05",
    "AutoMLJobSecondaryStatus": "Completed",
    "AutoMLJobStatus": "Completed",
    "BestCandidate": {
        "CandidateName": "automl-dm-16-20-10-054If5RqjsyAN-001-61c635e4",
        "CandidateProperties": {
            "CandidateArtifactLocations": {
                "Explainability": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/docu
mentation/explainability/output",
                "ModelInsights": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/docum
entation/model_monitor/output"
            },
            "CandidateMetrics": [
                {
                    "MetricName": "F1",
                    "Set": "Validation",
                    "StandardMetricName": "F1",
                    "Value": 0.995169997215271
                },
                {
                    "MetricName": "LogLoss",
                    "Set": "Validation",
                    "StandardMetricName": "LogLoss",
                    "Value": 0.11816000193357468
                },
                {
                    "MetricName": "Recall",
```

```
                    "Set": "Validation",
                    "StandardMetricName": "Recall",
                    "Value": 1.0
                },
                {
                    "MetricName": "Precision",
                    "Set": "Validation",
                    "StandardMetricName": "Precision",
                    "Value": 0.9904199838638306
                },
                {
                    "MetricName": "AUC",
                    "Set": "Validation",
                    "StandardMetricName": "AUC",
                    "Value": 0.9990599751472473
                },
                {
                    "MetricName": "Accuracy",
                    "Set": "Validation",
                    "StandardMetricName": "Accuracy",
                    "Value": 0.9953600168228149
                },
                {
                    "MetricName": "BalancedAccuracy",
                    "Set": "Validation",
                    "StandardMetricName": "BalancedAccuracy",
                    "Value": 0.9955800175666809
                }
            ]
        },
        "CandidateStatus": "Completed",
        "CandidateSteps": [
            {
                "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:processing-job/automl-dm-16-20-10-05-d
b-1-2887815310fd4416804d2079485d7d062e1a",
                "CandidateStepName": "automl-dm-16-20-10-05-db-1-2887815310fd4416804d2079485d7d062e1a",
                "CandidateStepType": "AWS::SageMaker::ProcessingJob"
            },
            {
                "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:training-job/automl-dm-16-20-10-05-dpp
1-1-8411a0fbc81748a9958acf62493120d4d7",
                "CandidateStepName": "automl-dm-16-20-10-05-dpp1-1-8411a0fbc81748a9958acf62493120d4d7",
                "CandidateStepType": "AWS::SageMaker::TrainingJob"
            },
```

```
            {
                "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:transform-job/automl-dm-16-20-10-05-dp
p1-csv-1-18357715aec34f7eb4c3b75bad6e49",
                "CandidateStepName": "automl-dm-16-20-10-05-dpp1-csv-1-18357715aec34f7eb4c3b75bad6e49",
                "CandidateStepType": "AWS::SageMaker::TransformJob"
            },
            {
                "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:training-job/automl-dm-16-20-10-054If5
RqjsyAN-001-61c635e4",
                "CandidateStepName": "automl-dm-16-20-10-054If5RqjsyAN-001-61c635e4",
                "CandidateStepType": "AWS::SageMaker::TrainingJob"
            }
        ],
        "CreationTime": "2023-04-16 20:33:00+00:00",
        "EndTime": "2023-04-16 20:38:38+00:00",
        "FinalAutoMLJobObjectiveMetric": {
            "MetricName": "validation:f1_binary",
            "StandardMetricName": "F1",
            "Value": 0.995169997215271
        },
        "InferenceContainers": [
            {
                "Environment": {
                    "AUTOML_TRANSFORM_MODE": "feature-transform",
                    "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "application/x-recordio-protobuf",
                    "SAGEMAKER_PROGRAM": "sagemaker_serve",
                    "SAGEMAKER_SUBMIT_DIRECTORY": "/opt/ml/model/code"
                },
                "Image": "683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-sklearn-automl:2.5-1-cpu-py3",
                "ModelDataUrl": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/data-p
rocessor-models/automl-dm-16-20-10-05-dpp1-1-8411a0fbc81748a9958acf62493120d4d7/output/model.tar.gz"
            },
            {
                "Environment": {
                    "MAX_CONTENT_LENGTH": "20971520",
                    "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "text/csv",
                    "SAGEMAKER_INFERENCE_OUTPUT": "predicted_label",
                    "SAGEMAKER_INFERENCE_SUPPORTED": "predicted_label,probability,probabilities"
                },
                "Image": "683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-xgboost:1.3-1-cpu-py3",
                "ModelDataUrl": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/tuning
/automl-dm--dpp1-xgb/automl-dm-16-20-10-054If5RqjsyAN-001-61c635e4/output/model.tar.gz"
            },
            {
```

```
                "Environment": {
                    "AUTOML_TRANSFORM_MODE": "inverse-label-transform",
                    "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "text/csv",
                    "SAGEMAKER_INFERENCE_INPUT": "predicted_label",
                    "SAGEMAKER_INFERENCE_OUTPUT": "predicted_label",
                    "SAGEMAKER_INFERENCE_SUPPORTED": "predicted_label,probability,labels,probabilities",
                    "SAGEMAKER_PROGRAM": "sagemaker_serve",
                    "SAGEMAKER_SUBMIT_DIRECTORY": "/opt/ml/model/code"
                },
                "Image": "683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-sklearn-automl:2.5-1-cpu-py3",
                "ModelDataUrl": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/data-p
rocessor-models/automl-dm-16-20-10-05-dpp1-1-8411a0fbc81748a9958acf62493120d4d7/output/model.tar.gz"
            }
        ],
        "LastModifiedTime": "2023-04-16 20:39:21.759000+00:00",
        "ObjectiveStatus": "Succeeded"
    },
    "CreationTime": "2023-04-16 20:14:58.074000+00:00",
    "EndTime": "2023-04-16 20:47:31.359000+00:00",
    "GenerateCandidateDefinitionsOnly": false,
    "InputDataConfig": [
        {
            "ChannelType": "training",
            "ContentType": "text/csv;header=present",
            "DataSource": {
                "S3DataSource": {
                    "S3DataType": "S3Prefix",
                    "S3Uri": "s3://sagemaker-us-east-1-614093401978/data/df_autopilot.csv"
                }
            },
            "TargetAttributeName": "AML_detected"
        }
    ],
    "LastModifiedTime": "2023-04-16 20:47:31.403000+00:00",
    "OutputDataConfig": {
        "S3OutputPath": "s3://sagemaker-us-east-1-614093401978/models/autopilot"
    },
    "ResolvedAttributes": {
        "AutoMLJobObjective": {
            "MetricName": "F1"
        },
        "CompletionCriteria": {
            "MaxAutoMLJobRuntimeInSeconds": 5400,
            "MaxCandidates": 3,
```

```
            "MaxRuntimePerTrainingJobInSeconds": 900
        },
        "ProblemType": "BinaryClassification"
    },
    "ResponseMetadata": {
        "HTTPHeaders": {
            "content-length": "5897",
            "content-type": "application/x-amz-json-1.1",
            "date": "Mon, 17 Apr 2023 03:55:50 GMT",
            "x-amzn-requestid": "d08f68fe-7de6-4222-9709-b84f6db538d3"
        },
        "HTTPStatusCode": 200,
        "RequestId": "d08f68fe-7de6-4222-9709-b84f6db538d3",
        "RetryAttempts": 0
    },
    "RoleArn": "arn:aws:iam::614093401978:role/LabRole"
}
```


In [120…

```
print(best_candidate.keys())
```

```
dict_keys(['CandidateName', 'FinalAutoMLJobObjectiveMetric', 'ObjectiveStatus', 'CandidateSteps', 'CandidateStatus',
'InferenceContainers', 'CreationTime', 'EndTime', 'LastModifiedTime', 'CandidateProperties'])
```

In [121…

```
while "CandidateName" not in best_candidate:
    best_candidate_response = sm.describe_auto_ml_job(AutoMLJobName=auto_ml_job_name)
    best_candidate = best_candidate_response["BestCandidate"]
    print("[INFO] Autopilot Job is generating BestCandidate CandidateName. Please wait. ")
    print(json.dumps(best_candidate, indent=4, sort_keys=True, default=str))
    sleep(10)

print("[OK] BestCandidate CandidateName generated.")
```

```
[OK] BestCandidate CandidateName generated.
```

In [122…

```
while "FinalAutoMLJobObjectiveMetric" not in best_candidate:
    best_candidate_response = sm.describe_auto_ml_job(AutoMLJobName=auto_ml_job_name)
    best_candidate = best_candidate_response["BestCandidate"]
    print("[INFO] Autopilot Job is generating BestCandidate FinalAutoMLJobObjectiveMetric. Please wait. ")
    print(json.dumps(best_candidate, indent=4, sort_keys=True, default=str))
    sleep(10)

print("[OK] BestCandidate FinalAutoMLJobObjectiveMetric generated.")
```

```
[OK] BestCandidate FinalAutoMLJobObjectiveMetric generated.
```

In [123…

```
best_candidate_identifier = best_candidate["CandidateName"]
print("Candidate name: " + best_candidate_identifier)
print("Metric name: " + best_candidate["FinalAutoMLJobObjectiveMetric"]["MetricName"])
print("Metric value: " + str(best_candidate["FinalAutoMLJobObjectiveMetric"]["Value"]))
```

```
Candidate name: automl-dm-16-20-10-054If5RqjsyAN-001-61c635e4
Metric name: validation:f1_binary
Metric value: 0.995169997215271
```

In [124…

```
print(json.dumps(best_candidate, indent=4, sort_keys=True, default=str))
```

```
{
    "CandidateName": "automl-dm-16-20-10-054If5RqjsyAN-001-61c635e4",
    "CandidateProperties": {
        "CandidateArtifactLocations": {
            "Explainability": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/document
ation/explainability/output",
            "ModelInsights": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/documenta
tion/model_monitor/output"
        },
        "CandidateMetrics": [
            {
                "MetricName": "F1",
                "Set": "Validation",
                "StandardMetricName": "F1",
                "Value": 0.995169997215271
            },
            {
                "MetricName": "LogLoss",
                "Set": "Validation",
                "StandardMetricName": "LogLoss",
                "Value": 0.11816000193357468
            },
            {
                "MetricName": "Recall",
                "Set": "Validation",
                "StandardMetricName": "Recall",
                "Value": 1.0
            },
            {
                "MetricName": "Precision",
                "Set": "Validation",
                "StandardMetricName": "Precision",
                "Value": 0.9904199838638306
            },
            {
                "MetricName": "AUC",
                "Set": "Validation",
                "StandardMetricName": "AUC",
                "Value": 0.9990599751472473
            },
            {
                "MetricName": "Accuracy",
                "Set": "Validation",
                "StandardMetricName": "Accuracy",
```

```
                "Value": 0.9953600168228149
            },
            {
                "MetricName": "BalancedAccuracy",
                "Set": "Validation",
                "StandardMetricName": "BalancedAccuracy",
                "Value": 0.9955800175666809
            }
        ]
    },
    "CandidateStatus": "Completed",
    "CandidateSteps": [
        {
            "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:processing-job/automl-dm-16-20-10-05-db-1-
2887815310fd4416804d2079485d7d062e1a",
            "CandidateStepName": "automl-dm-16-20-10-05-db-1-2887815310fd4416804d2079485d7d062e1a",
            "CandidateStepType": "AWS::SageMaker::ProcessingJob"
        },
        {
            "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:training-job/automl-dm-16-20-10-05-dpp1-1-
8411a0fbc81748a9958acf62493120d4d7",
            "CandidateStepName": "automl-dm-16-20-10-05-dpp1-1-8411a0fbc81748a9958acf62493120d4d7",
            "CandidateStepType": "AWS::SageMaker::TrainingJob"
        },
        {
            "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:transform-job/automl-dm-16-20-10-05-dpp1-c
sv-1-18357715aec34f7eb4c3b75bad6e49",
            "CandidateStepName": "automl-dm-16-20-10-05-dpp1-csv-1-18357715aec34f7eb4c3b75bad6e49",
            "CandidateStepType": "AWS::SageMaker::TransformJob"
        },
        {
            "CandidateStepArn": "arn:aws:sagemaker:us-east-1:614093401978:training-job/automl-dm-16-20-10-054If5Rqjs
yAN-001-61c635e4",
            "CandidateStepName": "automl-dm-16-20-10-054If5RqjsyAN-001-61c635e4",
            "CandidateStepType": "AWS::SageMaker::TrainingJob"
        }
    ],
    "CreationTime": "2023-04-16 20:33:00+00:00",
    "EndTime": "2023-04-16 20:38:38+00:00",
    "FinalAutoMLJobObjectiveMetric": {
        "MetricName": "validation:f1_binary",
        "StandardMetricName": "F1",
        "Value": 0.995169997215271
    },
```

```
    "InferenceContainers": [
        {
            "Environment": {
                "AUTOML_TRANSFORM_MODE": "feature-transform",
                "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "application/x-recordio-protobuf",
                "SAGEMAKER_PROGRAM": "sagemaker_serve",
                "SAGEMAKER_SUBMIT_DIRECTORY": "/opt/ml/model/code"
            },
            "Image": "683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-sklearn-automl:2.5-1-cpu-py3",
            "ModelDataUrl": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/data-proce
ssor-models/automl-dm-16-20-10-05-dpp1-1-8411a0fbc81748a9958acf62493120d4d7/output/model.tar.gz"
        },
        {
            "Environment": {
                "MAX_CONTENT_LENGTH": "20971520",
                "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "text/csv",
                "SAGEMAKER_INFERENCE_OUTPUT": "predicted_label",
                "SAGEMAKER_INFERENCE_SUPPORTED": "predicted_label,probability,probabilities"
            },
            "Image": "683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-xgboost:1.3-1-cpu-py3",
            "ModelDataUrl": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/tuning/aut
oml-dm--dpp1-xgb/automl-dm-16-20-10-054If5RqjsyAN-001-61c635e4/output/model.tar.gz"
        },
        {
            "Environment": {
                "AUTOML_TRANSFORM_MODE": "inverse-label-transform",
                "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "text/csv",
                "SAGEMAKER_INFERENCE_INPUT": "predicted_label",
                "SAGEMAKER_INFERENCE_OUTPUT": "predicted_label",
                "SAGEMAKER_INFERENCE_SUPPORTED": "predicted_label,probability,labels,probabilities",
                "SAGEMAKER_PROGRAM": "sagemaker_serve",
                "SAGEMAKER_SUBMIT_DIRECTORY": "/opt/ml/model/code"
            },
            "Image": "683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-sklearn-automl:2.5-1-cpu-py3",
            "ModelDataUrl": "s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/data-proce
ssor-models/automl-dm-16-20-10-05-dpp1-1-8411a0fbc81748a9958acf62493120d4d7/output/model.tar.gz"
        }
    ],
    "LastModifiedTime": "2023-04-16 20:39:21.759000+00:00",
    "ObjectiveStatus": "Succeeded"
}
```

View individual Autopilot jobs

In [125…

```
while "CandidateSteps" not in best_candidate:
    best_candidate_response = sm.describe_auto_ml_job(AutoMLJobName=auto_ml_job_name)
    best_candidate = best_candidate_response["BestCandidate"]
    print("[INFO] Autopilot Job is generating BestCandidate CandidateSteps. Please wait. ")
    print(json.dumps(best_candidate, indent=4, sort_keys=True, default=str))
    sleep(10)

best_candidate = best_candidate_response["BestCandidate"]
print("[OK] BestCandidate CandidateSteps generated.")
```

```
[OK] BestCandidate CandidateSteps generated.
```

In [126…

```
while "CandidateStepType" not in best_candidate["CandidateSteps"][0]:
    best_candidate_response = sm.describe_auto_ml_job(AutoMLJobName=auto_ml_job_name)
    best_candidate = best_candidate_response["BestCandidate"]
    print("[INFO] Autopilot Job is generating BestCandidate CandidateSteps CandidateStepType. Please wait. ")
    print(json.dumps(best_candidate, indent=4, sort_keys=True, default=str))
    sleep(10)

best_candidate = best_candidate_response["BestCandidate"]
print("[OK] BestCandidate CandidateSteps CandidateStepType generated.")
```

```
[OK] BestCandidate CandidateSteps CandidateStepType generated.
```

In [127…

```
while "CandidateStepName" not in best_candidate["CandidateSteps"][0]:
    best_candidate_response = sm.describe_auto_ml_job(AutoMLJobName=auto_ml_job_name)
    best_candidate = best_candidate_response["BestCandidate"]
    print("[INFO] Autopilot Job is generating BestCandidate CandidateSteps CandidateStepName. Please wait. ")
    print(json.dumps(best_candidate, indent=4, sort_keys=True, default=str))
    sleep(10)

best_candidate = best_candidate_response["BestCandidate"]
print("[OK] BestCandidate CandidateSteps CandidateStepName generated.")
```

```
[OK] BestCandidate CandidateSteps CandidateStepName generated.
```

In [128…

```
best_candidate
```

Out[128]:

```
{'CandidateName': 'automl-dm-16-20-10-054If5RqjsyAN-001-61c635e4',
 'FinalAutoMLJobObjectiveMetric': {'MetricName': 'validation:f1_binary',
  'Value': 0.995169997215271,
  'StandardMetricName': 'F1'},
 'ObjectiveStatus': 'Succeeded',
 'CandidateSteps': [{'CandidateStepType': 'AWS::SageMaker::ProcessingJob',
   'CandidateStepArn': 'arn:aws:sagemaker:us-east-1:614093401978:processing-job/automl-dm-16-20-10-05-db-1-2887815310fd4416804d2079485d7d062e1a',
   'CandidateStepName': 'automl-dm-16-20-10-05-db-1-2887815310fd4416804d2079485d7d062e1a'},
  {'CandidateStepType': 'AWS::SageMaker::TrainingJob',
   'CandidateStepArn': 'arn:aws:sagemaker:us-east-1:614093401978:training-job/automl-dm-16-20-10-05-dpp1-1-8411a0fbc81748a9958acf62493120d4d7',
   'CandidateStepName': 'automl-dm-16-20-10-05-dpp1-1-8411a0fbc81748a9958acf62493120d4d7'},
  {'CandidateStepType': 'AWS::SageMaker::TransformJob',
   'CandidateStepArn': 'arn:aws:sagemaker:us-east-1:614093401978:transform-job/automl-dm-16-20-10-05-dpp1-csv-1-18357715aec34f7eb4c3b75bad6e49',
   'CandidateStepName': 'automl-dm-16-20-10-05-dpp1-csv-1-18357715aec34f7eb4c3b75bad6e49'},
  {'CandidateStepType': 'AWS::SageMaker::TrainingJob',
   'CandidateStepArn': 'arn:aws:sagemaker:us-east-1:614093401978:training-job/automl-dm-16-20-10-054If5RqjsyAN-001-61c635e4',
   'CandidateStepName': 'automl-dm-16-20-10-054If5RqjsyAN-001-61c635e4'}],
 'CandidateStatus': 'Completed',
 'InferenceContainers': [{'Image': '683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-sklearn-automl:2.5-1-cpu-py3',
   'ModelDataUrl': 's3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/data-processor-models/automl-dm-16-20-10-05-dpp1-1-8411a0fbc81748a9958acf62493120d4d7/output/model.tar.gz',
   'Environment': {'AUTOML_TRANSFORM_MODE': 'feature-transform',
    'SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT': 'application/x-recordio-protobuf',
    'SAGEMAKER_PROGRAM': 'sagemaker_serve',
    'SAGEMAKER_SUBMIT_DIRECTORY': '/opt/ml/model/code'}},
  {'Image': '683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-xgboost:1.3-1-cpu-py3',
   'ModelDataUrl': 's3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/tuning/automl-dm--dpp1-xgb/automl-dm-16-20-10-054If5RqjsyAN-001-61c635e4/output/model.tar.gz',
   'Environment': {'MAX_CONTENT_LENGTH': '20971520',
    'SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT': 'text/csv',
    'SAGEMAKER_INFERENCE_OUTPUT': 'predicted_label',
    'SAGEMAKER_INFERENCE_SUPPORTED': 'predicted_label,probability,probabilities'}},
  {'Image': '683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-sklearn-automl:2.5-1-cpu-py3',
   'ModelDataUrl': 's3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/data-processor-models/automl-dm-16-20-10-05-dpp1-1-8411a0fbc81748a9958acf62493120d4d7/output/model.tar.gz',
   'Environment': {'AUTOML_TRANSFORM_MODE': 'inverse-label-transform',
    'SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT': 'text/csv',
    'SAGEMAKER_INFERENCE_INPUT': 'predicted_label',
    'SAGEMAKER_INFERENCE_OUTPUT': 'predicted_label',
```

```
    'SAGEMAKER_INFERENCE_SUPPORTED': 'predicted_label,probability,labels,probabilities',
    'SAGEMAKER_PROGRAM': 'sagemaker_serve',
    'SAGEMAKER_SUBMIT_DIRECTORY': '/opt/ml/model/code'}}],
 'CreationTime': datetime.datetime(2023, 4, 16, 20, 33, tzinfo=tzlocal()),
 'EndTime': datetime.datetime(2023, 4, 16, 20, 38, 38, tzinfo=tzlocal()),
 'LastModifiedTime': datetime.datetime(2023, 4, 16, 20, 39, 21, 759000, tzinfo=tzlocal()),
 'CandidateProperties': {'CandidateArtifactLocations': {'Explainability': 's3://sagemaker-us-east-1-614093401978/mod
els/autopilot/automl-dm-16-20-10-05/documentation/explainability/output',
   'ModelInsights': 's3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/documentation/mode
l_monitor/output'},
  'CandidateMetrics': [{'MetricName': 'F1',
    'Value': 0.995169997215271,
    'Set': 'Validation',
    'StandardMetricName': 'F1'},
   {'MetricName': 'LogLoss',
    'Value': 0.11816000193357468,
    'Set': 'Validation',
    'StandardMetricName': 'LogLoss'},
   {'MetricName': 'Recall',
    'Value': 1.0,
    'Set': 'Validation',
    'StandardMetricName': 'Recall'},
   {'MetricName': 'Precision',
    'Value': 0.9904199838638306,
    'Set': 'Validation',
    'StandardMetricName': 'Precision'},
   {'MetricName': 'AUC',
    'Value': 0.9990599751472473,
    'Set': 'Validation',
    'StandardMetricName': 'AUC'},
   {'MetricName': 'Accuracy',
    'Value': 0.9953600168228149,
    'Set': 'Validation',
    'StandardMetricName': 'Accuracy'},
   {'MetricName': 'BalancedAccuracy',
    'Value': 0.9955800175666809,
    'Set': 'Validation',
    'StandardMetricName': 'BalancedAccuracy'}]}}
```

In [129…

```
steps = []
for step in best_candidate["CandidateSteps"]:
    print("Candidate Step Type: {}".format(step["CandidateStepType"]))
    print("Candidate Step Name: {}".format(step["CandidateStepName"]))
    steps.append(step["CandidateStepName"])
```

```
Candidate Step Type: AWS::SageMaker::ProcessingJob
Candidate Step Name: automl-dm-16-20-10-05-db-1-2887815310fd4416804d2079485d7d062e1a
Candidate Step Type: AWS::SageMaker::TrainingJob
Candidate Step Name: automl-dm-16-20-10-05-dpp1-1-8411a0fbc81748a9958acf62493120d4d7
Candidate Step Type: AWS::SageMaker::TransformJob
Candidate Step Name: automl-dm-16-20-10-05-dpp1-csv-1-18357715aec34f7eb4c3b75bad6e49
Candidate Step Type: AWS::SageMaker::TrainingJob
Candidate Step Name: automl-dm-16-20-10-054If5RqjsyAN-001-61c635e4
```

In [130…

```
from IPython.core.display import display, HTML

display(
    HTML(
        '<b>Review Best Candidate <a target="blank" href="https://console.aws.amazon.com/sagemaker/home?region={}#/pr
            region, steps[0]
        )
    )
)
```

Review Best Candidate Processing Job

In [131…

```
from IPython.core.display import display, HTML

display(
    HTML(
        '<b>Review Best Candidate <a target="blank" href="https://console.aws.amazon.com/sagemaker/home?region={}#/jo
            region, steps[1]
        )
    )
)
```

Review Best Candidate Training Job

In [132…

```
from IPython.core.display import display, HTML

display(
    HTML(
        '<b>Review Best Candidate <a target="blank" href="https://console.aws.amazon.com/sagemaker/home?region={}#/tr
            region, steps[2]
        )
    )
)
```

Review Best Candidate Transform Job

In [133…

```
from IPython.core.display import display, HTML

display(
    HTML(
        '<b>Review Best Candidate <a target="blank" href="https://console.aws.amazon.com/sagemaker/home?region={}#/jo
            region, steps[3]
        )
    )
)
```

Review Best Candidate Training Job (Tuning)

In [134…

```
from IPython.core.display import display, HTML

display(
    HTML(
        '<b>Review All <a target="blank" href="https://console.aws.amazon.com/sagemaker/home?region={}#/processing-jo
            region
        )
    )
)
```

Review All Processing Jobs

Review all output in S3

In [135…

```
from IPython.core.display import display, HTML

display(
    HTML(
        '<b>Review All <a target="blank" href="https://s3.console.aws.amazon.com/s3/buckets/{}?region={}&prefix=model
            bucket, region, auto_ml_job_name
        )
    )
)
```

Review All Output in S3

See the containers and models within the inference pipeline

In [136…

```
while "InferenceContainers" not in best_candidate:
    best_candidate_response = sm.describe_auto_ml_job(AutoMLJobName=auto_ml_job_name)
    best_candidate = best_candidate_response["BestCandidate"]
    print("[INFO] Autopilot Job is generating BestCandidate InferenceContainers. Please wait. ")
    print(json.dumps(best_candidate, indent=4, sort_keys=True, default=str))
    sleep(10)

print("[OK] BestCandidate InferenceContainers generated.")
```

```
[OK] BestCandidate InferenceContainers generated.
```

In [137…

```
best_candidate_containers = best_candidate["InferenceContainers"]
```

In [138…

```
for container in best_candidate_containers:
    print(container["Image"])
    print(container["ModelDataUrl"])
    print("======================")
```

```
683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-sklearn-automl:2.5-1-cpu-py3
s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/data-processor-models/automl-dm-16-20-10-05-dpp1-1-8411a0fbc81748a9958acf62493120d4d7/output/model.tar.gz
======================
683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-xgboost:1.3-1-cpu-py3
s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/tuning/automl-dm--dpp1-xgb/automl-dm-16-20-10-054If5RqjsyAN-001-61c635e4/output/model.tar.gz
======================
683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-sklearn-automl:2.5-1-cpu-py3
s3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/data-processor-models/automl-dm-16-20-10-05-dpp1-1-8411a0fbc81748a9958acf62493120d4d7/output/model.tar.gz
======================
```

Update Containers to show predicted label and confidence score

In [139…

```python
for container in best_candidate_containers:
    print(container["Environment"])
    print("======================")
```

```
{'AUTOML_TRANSFORM_MODE': 'feature-transform', 'SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT': 'application/x-recordio-protobuf', 'SAGEMAKER_PROGRAM': 'sagemaker_serve', 'SAGEMAKER_SUBMIT_DIRECTORY': '/opt/ml/model/code'}
======================
{'MAX_CONTENT_LENGTH': '20971520', 'SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT': 'text/csv', 'SAGEMAKER_INFERENCE_OUTPUT': 'predicted_label', 'SAGEMAKER_INFERENCE_SUPPORTED': 'predicted_label,probability,probabilities'}
======================
{'AUTOML_TRANSFORM_MODE': 'inverse-label-transform', 'SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT': 'text/csv', 'SAGEMAKER_INFERENCE_INPUT': 'predicted_label', 'SAGEMAKER_INFERENCE_OUTPUT': 'predicted_label', 'SAGEMAKER_INFERENCE_SUPPORTED': 'predicted_label,probability,labels,probabilities', 'SAGEMAKER_PROGRAM': 'sagemaker_serve', 'SAGEMAKER_SUBMIT_DIRECTORY': '/opt/ml/model/code'}
======================
```

In [140…

```python
best_candidate_containers[1]["Environment"].update({"SAGEMAKER_INFERENCE_OUTPUT": "predicted_label, probability"})
best_candidate_containers[2]["Environment"].update({"SAGEMAKER_INFERENCE_INPUT": "predicted_label, probability"})
best_candidate_containers[2]["Environment"].update({"SAGEMAKER_INFERENCE_OUTPUT": "predicted_label, probability"})
```

In [141…

```python
for container in best_candidate_containers:
    print(container["Environment"])
    print("======================")
```

```
{'AUTOML_TRANSFORM_MODE': 'feature-transform', 'SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT': 'application/x-recordio-proto
buf', 'SAGEMAKER_PROGRAM': 'sagemaker_serve', 'SAGEMAKER_SUBMIT_DIRECTORY': '/opt/ml/model/code'}
======================
{'MAX_CONTENT_LENGTH': '20971520', 'SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT': 'text/csv', 'SAGEMAKER_INFERENCE_OUTPUT':
'predicted_label, probability', 'SAGEMAKER_INFERENCE_SUPPORTED': 'predicted_label,probability,probabilities'}
======================
{'AUTOML_TRANSFORM_MODE': 'inverse-label-transform', 'SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT': 'text/csv', 'SAGEMAKER_
INFERENCE_INPUT': 'predicted_label, probability', 'SAGEMAKER_INFERENCE_OUTPUT': 'predicted_label, probability', 'SAG
EMAKER_INFERENCE_SUPPORTED': 'predicted_label,probability,labels,probabilities', 'SAGEMAKER_PROGRAM': 'sagemaker_ser
ve', 'SAGEMAKER_SUBMIT_DIRECTORY': '/opt/ml/model/code'}
======================
```

Autopilot chose XGBoost as best candidate

In [142…

```
print(best_candidate["InferenceContainers"])
```

```
[{'Image': '683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-sklearn-automl:2.5-1-cpu-py3', 'ModelDataUrl': 's
3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/data-processor-models/automl-dm-16-20-10
-05-dpp1-1-8411a0fbc81748a9958acf62493120d4d7/output/model.tar.gz', 'Environment': {'AUTOML_TRANSFORM_MODE': 'featur
e-transform', 'SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT': 'application/x-recordio-protobuf', 'SAGEMAKER_PROGRAM': 'sagem
aker_serve', 'SAGEMAKER_SUBMIT_DIRECTORY': '/opt/ml/model/code'}}, {'Image': '683313688378.dkr.ecr.us-east-1.amazona
ws.com/sagemaker-xgboost:1.3-1-cpu-py3', 'ModelDataUrl': 's3://sagemaker-us-east-1-614093401978/models/autopilot/aut
oml-dm-16-20-10-05/tuning/automl-dm--dpp1-xgb/automl-dm-16-20-10-054If5RqjsyAN-001-61c635e4/output/model.tar.gz', 'E
nvironment': {'MAX_CONTENT_LENGTH': '20971520', 'SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT': 'text/csv', 'SAGEMAKER_INFER
ENCE_OUTPUT': 'predicted_label, probability', 'SAGEMAKER_INFERENCE_SUPPORTED': 'predicted_label,probability,probabil
ities'}}, {'Image': '683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-sklearn-automl:2.5-1-cpu-py3', 'ModelDat
aUrl': 's3://sagemaker-us-east-1-614093401978/models/autopilot/automl-dm-16-20-10-05/data-processor-models/automl-dm
-16-20-10-05-dpp1-1-8411a0fbc81748a9958acf62493120d4d7/output/model.tar.gz', 'Environment': {'AUTOML_TRANSFORM_MODE
': 'inverse-label-transform', 'SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT': 'text/csv', 'SAGEMAKER_INFERENCE_INPUT': 'pred
icted_label, probability', 'SAGEMAKER_INFERENCE_OUTPUT': 'predicted_label, probability', 'SAGEMAKER_INFERENCE_SUPPOR
TED': 'predicted_label,probability,labels,probabilities', 'SAGEMAKER_PROGRAM': 'sagemaker_serve', 'SAGEMAKER_SUBMIT_
DIRECTORY': '/opt/ml/model/code'}}]
```

Deploy Model as a REST Endpoint

In [143…

```
%store -r autopilot_model_name
```

In [144…
```
try:
    autopilot_model_name
except NameError:
    timestamp_suffix = strftime("%d-%H-%M-%S", gmtime())
    autopilot_model_name = "automl-dm-model-" + timestamp_suffix
    print("[OK] Created Autopilot Model Name: " + autopilot_model_name)
```

In [145…
```
%store autopilot_model_name
```

```
Stored 'autopilot_model_name' (str)
```

In [146…
```
%store -r autopilot_model_arn
```

In [147…
```
try:
    autopilot_model_arn
except NameError:
    create_model_response = sm.create_model(
        Containers=best_candidate["InferenceContainers"], ModelName=autopilot_model_name, ExecutionRoleArn=role
    )
    autopilot_model_arn = create_model_response["ModelArn"]
    print("[OK] Created Autopilot Model: {}".format(autopilot_model_arn))
```

In [148…
```
%store autopilot_model_arn
```

```
Stored 'autopilot_model_arn' (str)
```

Define EndpointConfig Name

In [149…
```
timestamp_suffix = strftime("%d-%H-%M-%S", gmtime())
epc_name = "automl-dm-epc-" + timestamp_suffix

print(epc_name)
```

```
automl-dm-epc-17-03-55-53
```

Define REST Endpoint Name fo rAutopilot Model

In [150…
```
%store -r autopilot_endpoint_name
```

In [151…]
```
timestamp_suffix = strftime("%d-%H-%M-%S", gmtime())

try:
    autopilot_endpoint_name
except NameError:
    autopilot_endpoint_name = "automl-dm-ep-" + timestamp_suffix
    print("[OK] Created Autopilot Endpoint Name {}: ".format(autopilot_endpoint_name))
```

In [152…]
```
variant_name = "automl-dm-variant-" + timestamp_suffix
print("[OK] Created Endpoint Variant Name {}: ".format(variant_name))
```

```
[OK] Created Endpoint Variant Name automl-dm-variant-17-03-55-53:
```

In [153…]
```
%store autopilot_endpoint_name
```

```
Stored 'autopilot_endpoint_name' (str)
```

In [154…]
```
ep_config = sm.create_endpoint_config(
    EndpointConfigName=epc_name,
    ProductionVariants=[
        {
            "InstanceType": "ml.m5.large",
            "InitialInstanceCount": 1,
            "ModelName": autopilot_model_name,
            "VariantName": variant_name,
        }
    ],
)
```

In [155…]
```
%store -r autopilot_endpoint_arn
```

In [156…]
```
try:
    autopilot_endpoint_arn
except NameError:
    create_endpoint_response = sm.create_endpoint(EndpointName=autopilot_endpoint_name, EndpointConfigName=epc_name)
    autopilot_endpoint_arn = create_endpoint_response["EndpointArn"]
    print(autopilot_endpoint_arn)
```

In [157…]
```
%store autopilot_endpoint_arn
```

```
Stored 'autopilot_endpoint_arn' (str)
```

In [158…

```
from IPython.core.display import display, HTML

display(
    HTML(
        '<b>Review <a target="blank" href="https://console.aws.amazon.com/sagemaker/home?region={}#/endpoints/{}">Sag
            region, autopilot_endpoint_name
        )
    )
)
```

Review SageMaker REST Endpoint

Store Variables

In [159…

```
%store
```

```
Stored variables and their in-db values:
auto_ml_job_name                          -> 'automl-dm-16-20-10-05'
autopilot_endpoint_arn                    -> 'arn:aws:sagemaker:us-east-1:614093401978:endpoint
autopilot_endpoint_name                   -> 'automl-dm-ep-16-22-24-43'
autopilot_model_arn                       -> 'arn:aws:sagemaker:us-east-1:614093401978:model/au
autopilot_model_name                      -> 'automl-dm-model-16-22-20-58'
autopilot_train_s3_uri                    -> 's3://sagemaker-us-east-1-614093401978/data/df_aut
ingest_create_athena_db_passed            -> True
s3_private_path_csv                       -> 's3://sagemaker-us-east-1-614093401978/cell_data'
s3_public_path_clsm                       -> 's3://team4rawdatasets/CSV/Input/OHSU_BeatAML_Clin
s3_public_path_csv                        -> 's3://gdc-beataml1.0-crenolanib-phs001628-2-open/'
s3_public_path_pi                         -> 's3://team4rawdatasets/CSV/Input/OpenCell_ProteinI
setup_dependencies_passed                 -> True
setup_iam_roles_passed                    -> True
setup_instance_check_passed               -> True
setup_s3_bucket_passed                    -> True
```

Release Resources

In [160…

```
%%html

<p><b>Shutting down your kernel for this notebook to release resources.</b></p>
<button class="sm-command-button" data-commandlinker-command="kernelmenu:shutdown" style="display:none;">Shutdown Ker

<script>
try {
    els = document.getElementsByClassName("sm-command-button");
    els[0].click();
}
catch(err) {
    // NoOp
}
</script>
```

Shutting down your kernel for this notebook to release resources.

In [161…

```
%%javascript

try {
    Jupyter.notebook.save_checkpoint();
    Jupyter.notebook.session.delete();
}
catch(err) {
    // NoOp
}
```