# **Phase3 – Data Analytics with Cognos**

The Technologies and the Libraries used for developing the customer churn

prediction are:

**Programming Languages:**

* Python: Python is a popular choice for data science and machine learning due to its extensive libraries and community support.

**Machine Learning Libraries:**

* Scikit-Learn: This library provides a wide range of tools for predictive data analysis, including classification and regression models.
* XGBoost: A popular gradient boosting library that is often used for classification tasks like churn prediction.
* LightGBM: Another gradient boosting framework for achieving high accuracy and fast training.
* CatBoost: A gradient boosting library that is specifically designed for categorical features.
* TensorFlow and Keras: These libraries are excellent for developing deep learning models, such as neural networks.
* PyTorch: Another popular deep learning library with a flexible and dynamic approach to building models.

**Data Manipulation and Analysis**:

* **Pandas**: A widely-used Python library for data manipulation and analysis.
* **NumPy**: Essential for numerical operations on large datasets

**Data Visualization:**

* **Matplotlib:** For creating static, interactive, and animated plots.
* **Seaborn:** Built on top of Matplotlib, it provides a high-level interface for creating informative and attractive statistical graphics.
* **Plotly:** Useful for creating interactive and web-based visualizations.

**Feature Engineering**:

* Feature engineering is essential for creating informative input features for your model. Techniques and libraries used will depend on your data and problem.

**Data Preprocessing and Transformation:**

* Scikit-Learn's preprocessing tools, such as StandardScaler and OneHotEncoder, for data preparation.

**Model Evaluation:**

* Cross-validation techniques (e.g., K-fold cross-validation).
* Metrics like accuracy, precision, recall, F1-score, ROC AUC, and more, depending on your problem.

**Hyperparameter Tuning:**

* GridSearchCV or RandomizedSearchCV from Scikit-Learn for optimizing hyperparameters.

**Ensemble Methods:**

* Techniques like bagging (e.g., Random Forest) and boosting (e.g., AdaBoost, XGBoost) can be helpful in improving model performance.

**Model Deployment:**

* Flask, Django, or FastAPI for creating RESTful APIs to serve the model.
* Docker for containerization.
* Cloud platforms like AWS, Google Cloud, Azure, or Heroku for deployment.

**Database and Data Storage:**

* Depending on your data sources, you may need to use databases like MySQL, PostgreSQL, or NoSQL databases like MongoDB to store and retrieve customer data.

**Version Control:**

* Git and platforms like GitHub or GitLab for version control and collaboration.

**Monitoring and Logging:**

* Tools like Prometheus and Grafana for monitoring model performance.

**Notebooks and IDEs:**

* Jupyter Notebook, JupyterLab, or integrated development environments

like PyCharm, Visual Studio Code, or Spyder.

**Documentation and Collaboration:**

* Tools like Jira, Confluence, or Notion for project management and documentation.

**Automated Machine Learning (AutoML):**

* AutoML libraries like Auto-Sklearn, H2O.ai, or Google Cloud AutoML can automate the process to some extent.

**Security and Compliance:**

* Ensure data privacy and compliance with regulations like GDPR, HIPAA, etc