Amazon Customers Data Analysis

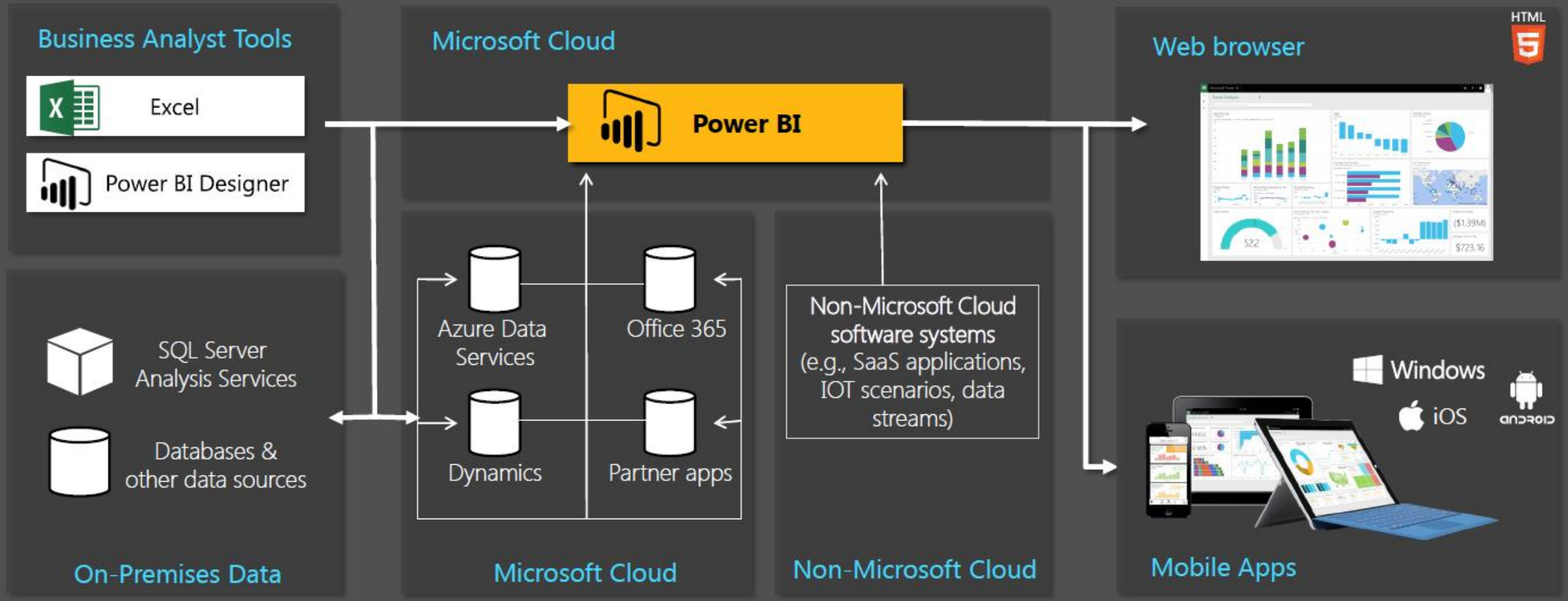
Objective:

Development of a predictive model for monitoring the Customers’ Data Analysis. "To analyze Amazon customer data to identify trends in feedback, understand customer preferences, and improve service quality based on insightful patterns and actionable insights.“

Benefits:

- Enhanced Customer Understanding.
- Improved Customer Satisfaction.
- Optimized Business Strategy.
- Proactive Issue Resolution .

Power BI Architecture Overview



Data Validation:

Implement data validation checks to ensure accuracy, completeness, and consistency of Amazon customer data.
Validate data against predefined rules, formats, and constraints to identify and address anomalies or errors.
Use techniques like data profiling, outlier detection, and statistical analysis to validate data quality.
Perform duplicate detection and removal to maintain clean and reliable datasets.
Data Transformation:

Convert and standardize data formats, units, and representations for consistency and compatibility.
Apply data cleansing techniques to correct errors, handle missing values, and remove inconsistencies.
Transform and aggregate data to generate meaningful insights and support analytical processes.
Use data enrichment methods such as merging with external datasets to enhance the richness and depth of analysis.

Data Insertion into Database:

- Establish a connection to the target database using appropriate credentials and configuration settings.
- Prepare the Amazon customer data for insertion by organizing it into structured tables or collections.
- Use SQL or database-specific commands to create tables that match the schema of the Amazon customer data.
- Execute database insertion operations (e.g., INSERT statements) to load the prepared data into the database.
- Handle errors and exceptions during insertion, implementing strategies such as transaction management and error logging.
- Validate the completeness and correctness of inserted data through querying and verification processes.
- Implement indexing and optimization techniques to ensure efficient retrieval and manipulation of the inserted data.
- These steps ensure that Amazon customer data is successfully integrated into a database system, enabling scalable storage, retrieval, and analysis of the dataset

Model Training:

Model Training with Random Forest in Python:

- Import Libraries:

Import necessary libraries including pandas for data handling and sklearn for machine learning.

- Data Preparation:

Load and preprocess the Amazon customer dataset, handling missing values and encoding categorical variables.

- Split Data:

Split the dataset into training and testing sets using train_test_split from sklearn.model_selection.

- Initialize Model:

Import RandomForestClassifier or RandomForestRegression from sklearn.ensemble.

Initialize the Random Forest model with desired parameters (e.g., number of trees, maximum depth).

- Train Model:

Fit the Random Forest model on the training data using model.fit(X_train, y_train).

- Model Evaluation:

Evaluate the model performance on the test set using metrics like accuracy, precision, recall, or RMSE for regression.

- Feature Importance:

Analyze feature importance using model.feature_importances_ to understand key factors influencing predictions.

- Hyperparameter Tuning (Optional):

Use techniques like grid search (GridSearchCV) or random search (RandomizedSearchCV) to optimize model hyperparameters.

- Prediction:

Use the trained model to make predictions on new data or the test set using model.predict(X_test).

- Model Persistence (Optional):

Save the trained model to disk using joblib or pickle for future use without retraining.

Prediction:

- Data Preprocessing:

Prepare new customer data by applying the same preprocessing steps used during model training (e.g., handling missing values, encoding categorical variables).

- Load Trained Model:

Load the trained Random Forest model using joblib or pickle to make predictions.

- Feature Engineering:

Extract relevant features from the new customer data to match the input format used during model training.

- Prediction Process:

Use the loaded model to predict customer behavior, preferences, or satisfaction levels based on input data.

- Result Interpretation:

Interpret model predictions to derive insights and make informed decisions for customer engagement and business strategies.

- Feedback Loop:

Incorporate prediction outcomes into ongoing customer analysis and refinement of business approaches.

Q & A:

Q1) What's the source of data?

The data for training is provided by the client in multiple batches and each batch contain multiple files

Q 2) What was the type of data?

The data was the combination of numerical and Categorical values.

Q 3) What's the complete flow you followed in this Project?

Refer slide 5th for better Understanding

Q 4) After the File validation what you do with incompatible file or files which didn't pass the validation?

Files like these are moved to the Achieve Folder and a list of these files has been shared with the client and we removed the bad data folder.

Q 5) What techniques were you using for data pre-processing?

- Handling the missing values
- Visualizing relation of independent variables with each other and output variables
- Removing outliers
- Cleaning data and imputing if null values are present.
- Converting categorical data into numeric values.
- Scaling the data

Q 6) How training was done or what models were used?

The training process involved using machine learning models to analyze and learn from the Amazon customer dataset. Various models were explored, with emphasis on those suitable for classification or regression tasks, such as Random Forest, Logistic Regression, and Gradient Boosting Machines. Each model was trained using a portion of the dataset to learn patterns and relationships between input features (e.g., customer behavior, product preferences) and target outcomes (e.g., customer satisfaction, purchase likelihood). Hyperparameter tuning and cross-validation techniques were employed to optimize model performance. The trained models were evaluated based on metrics like accuracy, precision, recall, or RMSE to select the best-performing model for prediction tasks.