**Final Project Report for Machine Learning Model for Auto Insurance Industry**

**About the Project**

The "Machine Learning Model for Auto Insurance Industry" project focuses on creating a predictive model that assists insurance companies in managing risk, optimizing premiums, and boosting profitability. With increasing online insurance applications and competitive pricing, this model aims to analyze customer data to identify those less likely to file claims. The project involves preprocessing a dataset of 600,000 records and 57 features, performing exploratory data analysis (EDA), and selecting the most relevant features to train a model capable of predicting insurance claims. By leveraging advanced machine learning techniques and cloud deployment, this project enables real-time analysis and decision-making, supporting effective and competitive pricing strategies for the auto insurance industry.

**System Requirements**

* **Software:**
  + Python 3.8 or above
  + Libraries: TensorFlow, Scikit-Learn, Pandas, NumPy, Matplotlib, Seaborn
  + Jupyter Notebook / Google Colab
  + Docker (for deployment)
  + Nimbus Platform (for cloud testing and deployment)
* **Hardware:**
  + CPU: Quad-core processor or higher
  + RAM: 8GB minimum (16GB recommended)
  + Storage: At least 10GB free space
  + Internet Connection: For accessing cloud resources and deploying on Nimbus

**Functional Requirements**

1. **Data Preprocessing** – Clean data by handling missing values, normalizing features, and managing class imbalance.
2. **Feature Engineering** – Select and engineer the most influential features to improve model accuracy.
3. **Model Training** – Develop and train models using algorithms such as Logistic Regression, Random Forest, and XGBoost.
4. **Model Evaluation** – Evaluate models using metrics like F1-score, precision, recall, and accuracy.
5. **Deployment** – Deploy the final model on the Nimbus cloud platform for real-time prediction.

**User Interface Requirements**

A basic interface for viewing predictions and generating reports could enhance usability for business analysts, allowing them to input customer data and view claim predictions. This may include:

* Form inputs for new customer data
* Dashboard displaying model predictions, probability scores, and key insights

**Inputs and Outputs**

* **Inputs:**
  + Customer demographic data (age, gender, location)
  + Vehicle information (age, model, driving record)
  + Historical claim data
* **Outputs:**
  + Prediction (Claim or No Claim)
  + Probability score for likelihood of claim
  + Key features impacting the prediction

**List of Subsystems**

1. **Data Processing Subsystem:** Handles data cleaning, normalization, and preparation.
2. **Feature Engineering Subsystem:** Identifies and transforms relevant features.
3. **Model Training Subsystem:** Develops and trains machine learning models.
4. **Evaluation Subsystem:** Assesses model performance using various metrics.
5. **Deployment Subsystem:** Deploys the final model on Nimbus for real-time usage.

**Other Applications Relevant to the Project**

The ML model can be applied to other insurance domains, including:

1. **Health Insurance:** To predict patient risk and optimize premium costs.
2. **Life Insurance:** To assess the likelihood of claims based on individual data points.
3. **Property Insurance:** To predict claims for properties based on location, size, and other factors.

**Designing of Test Cases**

| **Test Case ID** | **Description** | **Expected Outcome** |
| --- | --- | --- |
| TC-01 | Verify dataset upload and preprocessing | Dataset should be loaded and cleaned successfully |
| TC-02 | Check for missing values and handle appropriately | Missing values should be filled or removed |
| TC-03 | Verify feature selection and engineering | Key features should be selected based on correlation and importance |
| TC-04 | Train model using Logistic Regression | Model should be trained with accuracy displayed |
| TC-05 | Evaluate F1-score for trained model | F1-score should be calculated and displayed |
| TC-06 | Deploy model on Nimbus | Model should deploy without errors and generate predictions |

**Future Work**

1. **Improvement in Model Accuracy** – Explore ensemble techniques and more complex architectures, such as neural networks, for better accuracy.
2. **Expansion to Other Types of Insurance** – Adapt the model to different insurance types, allowing for customized solutions across various insurance products.
3. **User Interface Development** – Build a more user-friendly interface with predictive dashboards and reporting tools.

**References**

1. [Auto Insurance Claims Prediction: A Machine Learning Approach](https://link.springer.com)
2. Brownlee, J. (2020). *Machine Learning for Imbalanced Data*. Retrieved from MachineLearningMastery.com
3. Russell, S., & Norvig, P. (2016). *Artificial Intelligence: A Modern Approach.* Pearson Education.

**Reflection on Project Creation**

* **Technical Challenges:**
  + **Data Imbalance:** The dataset was heavily skewed, making accurate predictions challenging. Addressed with techniques like oversampling and undersampling.
  + **Model Selection:** Identifying the best model was complex due to varying performances across algorithms. Iterative tuning and comparative analysis were essential.
  + **Deployment Issues:** Implementing a model on Nimbus required familiarity with cloud deployment, which posed initial challenges.
* **Application of Existing Knowledge:**
  + Leveraged knowledge in data preprocessing and machine learning algorithms to handle data cleaning, feature selection, and model evaluation.
  + Experience with Python and Scikit-learn libraries helped streamline data manipulation and model training.
* **Personal Benefits:**
  + Enhanced understanding of machine learning application in real-world scenarios.
  + Gained confidence in end-to-end project execution, from data analysis to model deployment.
* **Additional Knowledge for Improvement:**
  + **Cloud Infrastructure Knowledge:** More experience with cloud platforms would streamline deployment.
  + **Advanced Model Optimization Techniques:** Familiarity with advanced hyperparameter tuning methods and ensemble techniques would benefit model performance.