**Machine Learning Project: Salary Prediction for Job Postings**

**Author**: Samuel Vierny  
**Date**: 30/11/2024

**Project 1:** [**Salary Prediction for Job Postings**](https://www.kaggle.com/competitions/salary-prediction-for-job-postings)

* **Type of ML Task:** Regression
  + Predicting numerical values (salaries) based on job posting data.
* **Dataset:**
  + Contains job posting metadata like job descriptions, titles, and possibly categorical features like location or industry.
  + Preprocessing involves handling text data (NLP methods), encoding categorical variables, and managing missing values.
* **Benefits:**
  + Straightforward regression task for beginners, focusing on common preprocessing techniques like handling text and categorical data.
  + Practical real-world application, making results interpretable and engaging.
  + Lower computational overhead compared to tasks with unstructured or large datasets (e.g., images or audio).
* **Limitations:**
  + Text data requires careful preprocessing, which might introduce complexity (e.g., tokenization, embeddings).
  + Possible skewed salary distribution could affect model performance.

**Project 2:** [**Music Hackathon**](https://www.kaggle.com/competitions/MusicHackathon)

* **Type of ML Task:** Regression or Classification
  + Could involve predicting numerical music properties or classifying genres or user preferences.
* **Dataset:**
  + Includes audio features (e.g., tempo, loudness) and metadata.
  + Requires preprocessing of audio data using techniques like spectral analysis or feature extraction (e.g., MFCCs).
* **Benefits:**
  + Engaging and creative domain, appealing for music enthusiasts.
  + Opportunity to work with advanced techniques like deep learning for audio data.
* **Limitations:**
  + High computational requirements, especially for audio file preprocessing.
  + Audio data preprocessing requires specialized knowledge of signal processing.

**Project 3:** [**Classifying the Brain on Music**](https://www.kaggle.com/competitions/classifying-the-brain-on-music)

* **Type of ML Task:** Classification
  + Predict brain activity patterns based on music-related stimuli.
* **Dataset:**
  + Likely includes EEG or fMRI data, which are high-dimensional time-series signals.
  + Preprocessing involves noise filtering, feature extraction, and dimensionality reduction.
* **Benefits:**
  + Unique interdisciplinary challenge combining neuroscience and machine learning.
  + Opportunity to apply techniques for time-series or high-dimensional data.
* **Limitations:**
  + Very complex for a first project, requiring expertise in both neuroscience and ML.
  + Computationally expensive and prone to overfitting due to the high-dimensional nature of the data.

**Why Project 1 is the Best Fit for You**

* **Alignment with Experience Level:**
  + Project 1 has manageable complexity for a first task, with data that’s easy to understand and preprocess.
* **Focus on Core ML Skills:**
  + It allows you to focus on supervised learning fundamentals, such as feature engineering, model tuning, and evaluation, without being overwhelmed by domain-specific complexities.
* **Balanced Challenge:**
  + Preprocessing job descriptions (text data) introduces a challenge without being overly complex, making it a good learning opportunity.
* **Interpretability of Results:**
  + Predicting salaries is a relatable and interpretable task, which helps in understanding model performance and potential biases.
* **Computational Requirements:**
  + The dataset is structured and less computationally demanding compared to audio or neuroscience datasets, making it feasible for faster experimentation.
* Passion subject:
  + While Music related assessments might be more related to my passions and interests, Project 1 still realistically the most relevant and fitting for my assessment. I will endeavour to take on Projects 2 and 3 another time or in the future once I have strengthened my machine learning skills.

**Competition Selection**

The **"Salary Prediction for Job Postings"** competition was ultimately chosen as it aligns with my experience level and allows for concentration on core machine learning skills. It offers a practical application with interpretable results, manageable computational requirements, and the opportunity to delve into data preprocessing and feature engineering extensively.

**Data Description**

The dataset comprises job postings enriched with various features:

* **Numerical Features**: Metrics such as company scores, director scores, and reviews.
* **Categorical Features**: Attributes including job titles, company names, locations, sectors, and skills.
* **Text Data**: Job descriptions and skills, often presented in multivalued formats.
* **Target Variable**: Mean\_Salary, representing the average salary for each job posting.

Notably, the data includes both single-valued and multivalued columns, with some features exhibiting high cardinality due to a large number of unique values.

**Model Development**

**Overview**

The project adhered to a structured methodology encompassing the following stages:

1. **Exploratory Data Analysis (EDA)**
2. **Data Cleaning and Preprocessing**
3. **Feature Engineering**
4. **Model Selection and Hyperparameter Tuning**
5. **Model Evaluation and Refinement**

**1. Exploratory Data Analysis (EDA)**

A thorough understanding of the data was imperative. The EDA phase involved:

* **Data Structure Analysis**: Identifying numerical and categorical features, with attention to multivalued columns such as Skills, Job, and Company.
* **Missing Value Assessment**: Evaluating the extent and distribution of missing data across the dataset.
* **Target Variable Examination**: Analyzing the distribution of Mean\_Salary to detect potential skewness and outliers that could influence model performance.

**2. Data Cleaning and Preprocessing**

**Handling Missing Values**:

* **Numerical Columns**: Missing values were imputed using the median to mitigate the influence of outliers.
* **Categorical Columns**: Missing entries were replaced with the placeholder "unknown" to retain records and signal missingness to the model.

**Scaling Numerical Features**:

* The StandardScaler was applied to standardize numerical features, ensuring that algorithms sensitive to feature scale, such as gradient-based methods, function optimally.

**3. Feature Engineering**

**Encoding Categorical Variables**:

* **Low-Cardinality Features**: Implemented One-Hot Encoding (OHE) for features with 139 or fewer unique values (e.g., State, Sector) to capture categorical distinctions without excessive dimensionality.
* **High-Cardinality Features**: Employed frequency encoding for features with a high number of unique values (e.g., Skills, Company) to reduce dimensionality while preserving informative variability.

**Processing Multivalued Columns**:

* **Skills**: Multivalued entries were parsed to extract individual skills, followed by binary encoding to represent the presence of each skill.
* **Other Multivalued Features**: Similar parsing and encoding techniques were applied to Job, Company, and Director to accurately capture their contributions.

**Feature Selection**:

* Leveraged feature importance metrics from ensemble models like Random Forest and CatBoost to identify and select the most impactful features, enhancing model efficiency and performance.

**4. Model Selection and Hyperparameter Tuning**

**Initial Modeling with PyCaret**:

* Utilized PyCaret for rapid prototyping and comparison of various algorithms, assessing performance based on metrics such as Root Mean Square Error (RMSE) and R-squared (R²).
* **Challenges Encountered**: The high dimensionality resulting from OHE led to memory constraints, necessitating alternative approaches.

**Transition to a Custom Pipeline**:

* Developed a custom preprocessing and modeling pipeline using scikit-learn and CatBoost to handle the data more efficiently.
* **Preprocessing Pipeline**: Modularized preprocessing steps to ensure consistency between training and test datasets, enhancing reproducibility.

**Hyperparameter Tuning**:

* Conducted hyperparameter optimization for CatBoostRegressor, tuning parameters such as depth, learning\_rate, and iterations using cross-validation techniques to improve model performance.

**5. Model Evaluation and Refinement**

**Performance Metrics**:

* **RMSE**: Served as the primary evaluation metric due to its sensitivity to large errors, which is critical in salary prediction contexts.
* **MAE and R²**: Provided additional insights into the model's predictive accuracy and explanatory power.

**Model Refinements**:

* **Dimensionality Reduction**: Addressed issues of high dimensionality by adjusting OHE thresholds and applying feature selection methods to reduce computational load without sacrificing predictive capability.
* **Feature Importance Analysis**: Focused on features contributing significantly to cumulative importance, streamlining the model and enhancing interpretability.

**Final Model**:

* Trained a CatBoostRegressor on the refined feature set with optimized hyperparameters.
* **Results**: Achieved improvements in RMSE and R² compared to initial models, indicating enhanced predictive performance.

**Reflections on the Modeling Process**

**Challenges Faced**

1. **High Dimensionality**:
   * **Issue**: The application of One-Hot Encoding to high-cardinality features resulted in an excessive number of features, leading to computational inefficiency and memory errors.
   * **Solution**: Implemented frequency encoding for high-cardinality features and adjusted the OHE threshold to balance detail retention with dimensionality concerns.
2. **Handling Multivalued Features**:
   * **Issue**: Features containing lists of values, such as Skills, complicated the encoding process.
   * **Solution**: Exploded multivalued entries to isolate individual elements, enabling proper encoding and ensuring that the model could capture the nuances of these features.
3. **Pipeline Development**:
   * **Issue**: Initial code structures lacked reusability and consistency between training and test datasets, leading to potential discrepancies in data processing.
   * **Solution**: Established a robust preprocessing pipeline to standardize data transformation across all datasets, enhancing consistency and reproducibility.
4. **Swapping Between Pipeline and Jupyter Notebook Stream**:

* **Issue**: Frequent switching between pipeline automation and Jupyter Notebook exploration caused inefficiencies and increased double work.
* **Solution**: Used the notebook environment for finding out feature exploration and trying other techniques. Stuck to pipeline components for reusability and consistency across experiments.

**Key Learnings**

* **Significance of EDA**: Comprehensive data exploration is essential for uncovering underlying patterns, anomalies, and data quality issues that can impact model effectiveness.
* **Impact of Feature Engineering**: Thoughtful handling of categorical and multivalued features is crucial for model performance, as it directly influences the quality of the input data.
* **Model Selection Nuances**: While automated tools like PyCaret facilitate rapid experimentation, they may not efficiently handle complex data structures, highlighting the need for customized approaches.
* **Importance of Reproducibility**: Developing modular, well-documented code ensures that results can be consistently reproduced and facilitates future model enhancements.

**Personal Reflection**

This project underscored the importance of a deep understanding of each step in the machine learning pipeline. Initially, reliance on external tools and assistance was significant; however, through iterative problem-solving and hands-on coding, I gained confidence in independently building and refining models. The experience enriched my skills in data preprocessing, feature engineering, and highlighted the iterative nature of effective model development.

**Conclusion**

The "Salary Prediction for Job Postings" project provided a practical application of supervised learning techniques to a real-world problem. Through systematic exploration, meticulous preprocessing, and iterative modeling, a predictive model was developed that estimates mean salaries with reasonable accuracy. The challenges encountered reinforced the importance of adaptability, thoroughness, and continuous learning in the field of machine learning.

**Kaggle Competition Details**

* **Kaggle Username**: *Samuel Vierny*
* **Competition Name**: Salary Prediction for Job Postings
* **Number of Teams**: *83*
* **Leaderboard Position**: *60 (if could submit it)*

*Please refer to the attached screenshot for verification of the leaderboard position.*

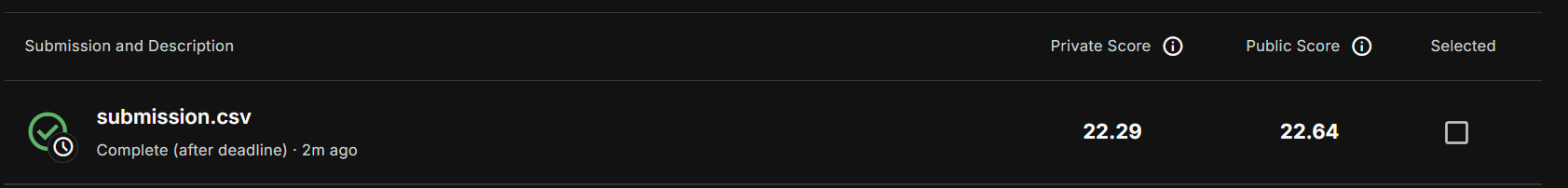
**Attachments**

* **Leaderboard Screenshot**

I could not access the leaderboard, because the competition somehow would not let me as it was disabled... See the following screenshot.

A screenshot of a computer

Description automatically generated

However below is my score. I can   


Unfortunately I would have place 60 in the leaderboard, out of then 84. This is a shame I and I would have liked to have done better.

* **Model Files and Code**: Submitted as a zip file containing the preprocessing pipeline, training script, and prediction script.